ektron

# ektron CMS400

SDK

## API Reference Documentation

Version 7.6

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

# Table of Contents


## Welcome 1

## How Do I... 2

## List of Developer Resources 4

## Ektron Namespace 7

**Server Controls API** **7**

Ektron.Cms Namespace 7

**Plug-in Extension API** **524**

Ektron.Cms.Extensibility Namespace 524

Classes 525

ActionRecipientInfo Class 526

ActionRecipientInfo.ActionRecipientInfo Constructor 526

ActionRecipientInfo Properties 526

ActionRecipientInfo.DisplayName Property 526

ActionRecipientInfo.Email Property 527

ActionRecipientInfo.FirstName Property 527

ActionRecipientInfo.IsUser Property 527

ActionRecipientInfo.LastName Property 527

ActionRecipientInfo.UserId Property 527

ActionRequestInfo Class 527

ActionRequestInfo.ActionRequestInfo Constructor 528

ActionRequestInfo Enumerations 528

Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestType Enumeration 528

ActionRequestInfo Properties 528

ActionRequestInfo.ErrorMessage Property 529

ActionRequestInfo.Id Property 529

ActionRequestInfo.Recipients Property 529

ActionRequestInfo.SenderId Property 529

ActionRequestInfo.Type Property 529

ContentInfo Class 530

ContentInfo Properties 530

ContentInfo.AssetID Property 530

ContentInfo.Comments Property 531
ContentInfo.ErrorMessage Property 531
ContentInfo.FileAndPath Property 531
ContentInfo.FileExtension Property 531
ContentInfo.FileName Property 531
ContentInfo.FolderID Property 531
ContentInfo.Html Property 532
ContentInfo.ID Property 532
ContentInfo.Language Property 532
ContentInfo.MimeType Property 532
ContentInfo.Status Property 532
ContentInfo.Title Property 532
ContentInfo.Type Property 533
ContentInfo.UserID Property 533
ExtensionAuthor Class 533
ExtensionAuthor.ExtensionAuthor Constructor 533
ExtensionAuthor Methods 533
ExtensionAuthor.ToString Method 534
ExtensionAuthor Properties 534
ExtensionAuthor.Author Property 534
ExtensionDescription Class 534
ExtensionDescription.ExtensionDescription Constructor 535
ExtensionDescription Methods 535
ExtensionDescription.ToString Method 535
ExtensionDescription Properties 535
ExtensionDescription.Desc Property 535
ExtensionEvent Class 535
ExtensionEvent Methods 536
ExtensionEvent.GetActiveDirectoryUserProxy Method 538
ExtensionEvent.GetAssetProxy Method 538
ExtensionEvent.GetBlogProxy Method 538
ExtensionEvent.GetBusinessRulesProxy Method 539
ExtensionEvent.GetCalendarEventTypeProxy Method 539
ExtensionEvent.GetCalendarProxy Method 539
ExtensionEvent.GetCommunityGroupProxy Method 539
ExtensionEvent.GetContentProxy Method 539
ExtensionEvent.GetContentRatingProxy Method 540
ExtensionEvent.GetCustomFieldsProxy Method 540
ExtensionEvent.GetFavoritesProxy Method 540

ExtensionEvent.GetFlagProxy Method 540
ExtensionEvent.GetFolderProxy Method 541
ExtensionEvent.GetFontProxy Method 541
ExtensionEvent.GetFormProxy Method 541
ExtensionEvent.GetFriendsProxy Method 541
ExtensionEvent.GetLibraryProxy Method 542
ExtensionEvent.GetMessageBoardProxy Method 542
ExtensionEvent.GetMetadataProxy Method 542
ExtensionEvent.GetObjectStateAsXml Method 542
ExtensionEvent.GetPermissionsProxy Method 542
ExtensionEvent.GetSearchManagerProxy Method 543
ExtensionEvent.GetSiteMapProxy Method 543
ExtensionEvent.GetSiteProxy Method 543
ExtensionEvent.GetTagsProxy Method 543
ExtensionEvent.GetTaskCategoryProxy Method 544
ExtensionEvent.GetTaskCategoryTypeProxy Method 544
ExtensionEvent.GetTaskProxy Method 544
ExtensionEvent.GetTaxonomyProxy Method 544
ExtensionEvent.GetThreadedDiscussionProxy Method 545
ExtensionEvent.GetUserProxy Method 545
ExtensionEvent.OnAfterActionRequestSent Method 545
ExtensionEvent.OnAfterContentAdd Method 545
ExtensionEvent.OnAfterContentDelete Method 546
ExtensionEvent.OnAfterContentEdit Method 546
ExtensionEvent.OnAfterContentFlagged Method 546
ExtensionEvent.OnAfterFolderAdd Method 546
ExtensionEvent.OnAfterFolderDelete Method 546
ExtensionEvent.OnAfterFolderEdit Method 547
ExtensionEvent.OnAfterFormSubmit Method 547
ExtensionEvent.OnAfterPublish Method 547
ExtensionEvent.OnAfterReviewRejected Method 547
ExtensionEvent.OnAfterUserAdd Method 548
ExtensionEvent.OnAfterUserDelete Method 548
ExtensionEvent.OnAfterUserEdit Method 548
ExtensionEvent.OnAfterUserLogin Method 548
ExtensionEvent.OnBeforeActionRequestSent Method 549
ExtensionEvent.OnBeforeContentAdd Method 549
ExtensionEvent.OnBeforeContentDelete Method 549
ExtensionEvent.OnBeforeContentEdit Method 549

ExtensionEvent.OnBeforeContentFlagged Method 550
ExtensionEvent.OnBeforeFolderAdd Method 550
ExtensionEvent.OnBeforeFolderDelete Method 550
ExtensionEvent.OnBeforeFolderEdit Method 550
ExtensionEvent.OnBeforeFormSubmit Method 550
ExtensionEvent.OnBeforePublish Method 551
ExtensionEvent.OnBeforeReviewRejected Method 551
ExtensionEvent.OnBeforeUserAdd Method 551
ExtensionEvent.OnBeforeUserDelete Method 551
ExtensionEvent.OnBeforeUserEdit Method 552
ExtensionEvent.OnBeforeUserLogin Method 552
ExtensionEvent Properties 552
ExtensionEvent.Content Property 552
ExtensionEvent.Folder Property 553
ExtensionEvent.Form Property 553
ExtensionEvent.Site Property 553
ExtensionEvent.User Property 553
FlagInfo Class 553
FlagInfo Properties 554
FlagInfo.ContentHtml Property 554
FlagInfo.ContentID Property 554
FlagInfo.ContentLanguage Property 555
FlagInfo.ContentTitle Property 555
FlagInfo.EntryId Property 555
FlagInfo.ErrorMessage Property 555
FlagInfo.FlagComment Property 555
FlagInfo.FlagDate Property 555
FlagInfo.FlagId Property 556
FlagInfo.FlagName Property 556
FlagInfo.UserAvatar Property 556
FlagInfo.UserDisplayName Property 556
FlagInfo.UserFirstName Property 556
FlagInfo.UserID Property 556
FlagInfo.UserLastName Property 557
FlagInfo.UserName Property 557
FlagInfo.VisitorID Property 557
FolderInfo Class 557
FolderInfo Properties 557
FolderInfo.Description Property 558

FolderInfo.ErrorMessage Property 558
FolderInfo.ID Property 558
FolderInfo.Name Property 558
FolderInfo.ParentID Property 558
FolderInfo.ReplicationState Property 558
FolderInfo.StyleSheet Property 559
FolderInfo.Type Property 559
FormField Class 559
FormField.FormField Constructor 559
FormField Properties 560
FormField.FieldName Property 560
FormField.FieldValue Property 560
FormInfo Class 560
FormInfo Methods 560
FormInfo.this Indexer 560
FormInfo Properties 561
FormInfo.ErrorMessage Property 561
FormInfo.FormDescription Property 561
FormInfo.FormFields Property 561
FormInfo.FormTitle Property 561
MessageBoardInfo Class 562
MessageBoardInfo.MessageBoardInfo Constructor 562
MessageBoardInfo Properties 562
MessageBoardInfo.ErrorMessage Property 562
RequestInfo Class 563
ReviewInfo Class 563
ReviewInfo Properties 563
ReviewInfo.AverageRating Property 564
ReviewInfo.ContentHtml Property 564
ReviewInfo.ContentID Property 564
ReviewInfo.ContentLanguage Property 564
ReviewInfo.ContentTitle Property 564
ReviewInfo.ErrorMessage Property 565
ReviewInfo.ID Property 565
ReviewInfo.NumberofRatings Property 565
ReviewInfo.Rating Property 565
ReviewInfo.RatingDate Property 565
ReviewInfo.State Property 565
ReviewInfo.UserAvatar Property 566

Copyright (c) Ektron Inc. 2008

ReviewInfo.UserComments Property 566
ReviewInfo.UserDisplayName Property 566
ReviewInfo.UserFirstName Property 566
ReviewInfo.UserID Property 566
ReviewInfo.UserLastName Property 566
ReviewInfo.UserName Property 567
ReviewInfo.VisitorID Property 567
SiteInfo Class 567
SiteInfo Properties 567
SiteInfo.AppPath Property 567
SiteInfo.IsSSL Property 568
SiteInfo.Name Property 568
SiteInfo.PhysicalSitePath Property 568
SiteInfo.Port Property 568
SiteInfo.Url Property 568
UserInfo Class 568
UserInfo Properties 569
UserInfo.Addres Property 569
UserInfo.DisplayName Property 569
UserInfo.Editor Property 570
UserInfo.EmailAddress Property 570
UserInfo.ErrorMessage Property 570
UserInfo.FirstName Property 570
UserInfo.ID Property 570
UserInfo.Language Property 570
UserInfo.LastName Property 571
UserInfo.Latitude Property 571
UserInfo.Longitude Property 571
UserInfo.Notifications Property 571
UserInfo.Type Property 571
UserInfo.UserName Property 571

**.NET Assemblies** **572**
eCommerce 573
Classes 573
AddressApi Class 574
AddressApi Methods 574
Add Method 575
AddressApi.Add Method (AddressData) 575
AddressApi.Add Method (AddressData, long) 575

AddressApi.Delete Method 576
AddressApi.GetItem Method 576
AddressApi.GetList Method 576
AddressApi.Update Method 576
AddressApi.UpdateOrderAddress Method 577
BasketApi Class 577
BasketApi Methods 577
CalculateBasket Method 578
BasketApi.CalculateBasket Method (Basket) 578
BasketApi.CalculateBasket Method (Basket, AddressData) 578
BasketApi.CalculateBasket Method (long) 579
BasketApi.CalculateBasket Method (long, AddressData) 579
BasketApi.CreateNewBasket Method 579
BasketApi.Delete Method 580
BasketApi.Empty Method 580
GetDefaultBasket Method 580
BasketApi.GetDefaultBasket Method () 581
BasketApi.GetDefaultBasket Method (long) 581
BasketApi.GetDefaultBasket Method (string) 581
BasketApi.GetItem Method 581
BasketApi.GetList Method 582
SetDefaultBasket Method 582
BasketApi.SetDefaultBasket Method (long) 582
BasketApi.SetDefaultBasket Method (long, long) 583
CatalogEntryApi Class 583
CatalogEntryApi Methods 583
CatalogEntryApi.Add Method 584
CatalogEntryApi.AddAndCheckIn Method 584
CatalogEntryApi.AddAndPublish Method 584
CatalogEntryApi.AddAndSave Method 584
CatalogEntryApi.Checkout Method 585
CatalogEntryApi.Delete Method 585
CatalogEntryApi.GetAttributeEntryList Method 585
GetItem Method 585
CatalogEntryApi.GetItem Method (long) 586
CatalogEntryApi.GetItem Method (long, long) 586
CatalogEntryApi.GetItemEdit Method 586
CatalogEntryApi.GetList Method 587
CatalogEntryApi.GetTaxonomyEntryList Method 587

CatalogEntryApi.Restore Method 587
CatalogEntryApi.UndoCheckOut Method 587
CatalogEntryApi.Update Method 588
CatalogEntryApi.UpdateAndCheckIn Method 588
CatalogEntryApi.UpdateAndPublish Method 588
ContractApi Class 588
ContractApi Methods 589
ContractApi.Add Method 589
ContractApi.Delete Method 589
ContractApi.GetItem Method 589
ContractApi.GetList Method 590
ContractApi.Update Method 590
CountryApi Class 590
CountryApi Methods 590
CountryApi.Add Method 591
CountryApi.CanDelete Method 591
CountryApi.Delete Method 591
CountryApi.GetItem Method 592
CountryApi.GetList Method 592
CountryApi.Update Method 592
CouponApi Class 592
CouponApi Methods 593
CouponApi.Add Method 593
CouponApi.AddCouponToCatalog Method 594
CouponApi.AddCouponToProduct Method 594
CouponApi.AddCouponToTaxonomy Method 594
CouponApi.CalculateBasketCoupons Method 595
CouponApi.Deactivate Method 595
CouponApi.Delete Method 595
CouponApi.DeleteCouponProducts Method 595
CouponApi.GetCatalogList Method 596
CouponApi.GetCouponsAppliedToBasket Method 596
CouponApi.GetItem Method 596
CouponApi.GetList Method 596
GetProductList Method 597
CouponApi.GetProductList Method (Criteria<CouponEntryProperty>) 597
CouponApi.GetProductList Method (long) 597
CouponApi.GetTaxonomyList Method 597
CouponApi.IsCouponAppliedToProduct Method 598

CouponApi.IsCouponUsedForBasket Method 598
CouponApi.IsCouponUsedForOrder Method 598
CouponApi.SaveCouponApplications Method 598
CouponApi.Update Method 599
ValidateCoupon Method 599
CouponApi.ValidateCoupon Method (long) 599
CouponApi.ValidateCoupon Method (long, long) 599
CurrencyApi Class 600
CurrencyApi Methods 600
CurrencyApi.Add Method 600
CurrencyApi.CanDelete Method 601
CurrencyApi.Delete Method 601
CurrencyApi.GetItem Method 601
CurrencyApi.GetList Method 601
CurrencyApi.Update Method 602
CustomerApi Class 602
CustomerApi Methods 602
Add Method 603
CustomerApi.Add Method (CustomerData, AddressData) 603
CustomerApi.Add Method (CustomerData, AddressData, AddressData) 603
CustomerApi.Delete Method 603
CustomerApi.GetItem Method 604
CustomerApi.GetList Method 604
CustomerApi.GetOrderList Method 604
CustomerApi.GetOrderTotal Method 605
CustomerApi.Update Method 605
ExchangeRateApi Class 605
ExchangeRateApi Methods 605
ExchangeRateApi.Add Method 606
ExchangeRateApi.ConvertPrice Method 606
ExchangeRateApi.Delete Method 606
ExchangeRateApi.GetCurrentExchangeRate Method 606
ExchangeRateApi.GetCurrentList Method 607
ExchangeRateApi.GetList Method 607
ExchangeRateApi.Update Method 607
InventoryApi Class 608
InventoryApi Methods 608
InventoryApi.SaveInventory Method 608
OrderApi Class 608

OrderApi Methods 609
Capture Method 609
OrderApi.Capture Method (long) 609
OrderApi.Capture Method (long, long) 610
OrderApi.GetCustomerOrderList Method 610
OrderApi.GetItem Method 610
GetList Method 611
OrderApi.GetList Method (Criteria<OrderProperty>) 611
OrderApi.GetList Method (DateTime, DateTime, PagingInfo) 611
GetOnHoldOrderList Method 611
OrderApi.GetOnHoldOrderList Method (PagingInfo) 612
OrderApi.GetOnHoldOrderList Method (PagingInfo, OrderProperty, EkEnumeration.OrderByDirection) 612
GetOrderMessage Method 612
OrderApi.GetOrderMessage Method (EkEnumeration.EmailMessageTypes, long) 613
OrderApi.GetOrderMessage Method (long, long) 613
OrderApi.GetOrderPayment Method 613
OrderApi.GetOrderStatus Method 614
GetProductOrderList Method 614
OrderApi.GetProductOrderList Method (long, Criteria<OrderProperty>) 614
OrderApi.GetProductOrderList Method (long, PagingInfo) 614
OrderApi.GetReport Method 615
OrderApi.ItemsInStock Method 615
OrderApi.MarkAsFraud Method 615
OrderApi.ProcessOrderInventory Method 615
OrderApi.RollbackOrderInventory Method 616
OrderApi.SetOrderStatus Method 616
OrderApi.SetOrderToHold Method 616
OrderApi.SetTrackingNumber Method 616
PackageApi Class 617
PackageApi Methods 617
PackageApi.Add Method 617
PackageApi.Delete Method 618
PackageApi.GetItem Method 618
PackageApi.GetList Method 618
PackageApi.Update Method 618
ProductTypeApi Class 619
ProductTypeApi Methods 619
ProductTypeApi.Add Method 619

ProductTypeApi.AddProductAttribute Method 620
ProductTypeApi.Delete Method 620
DeleteProductAttribute Method 620
ProductTypeApi.DeleteProductAttribute Method (long, System.Data.IDbTransaction) 620
ProductTypeApi.DeleteProductAttribute Method (long, bool, System.Data.IDbTransaction) 621
ProductTypeApi.GetFolderProductTypeList Method 621
GetItem Method 621
ProductTypeApi.GetItem Method (long) 621
ProductTypeApi.GetItem Method (long, bool) 621
ProductTypeApi.GetProductAttribute Method 622
ProductTypeApi.GetProductAttributeList Method 622
ProductTypeApi.Update Method 622
ProductTypeApi.UpdateProductAttribute Method 622
RecommendationApi Class 622
RecommendationApi Methods 623
RecommendationApi.Add Method 623
RecommendationApi.Delete Method 623
RecommendationApi.DeleteRecommendationByEntry Method 624
RecommendationApi.GetItem Method 624
RecommendationApi.Update Method 624
RecommendationApi.UpdateCrossSell Method 624
RecommendationApi.UpdateUpSell Method 625
RegionApi Class 625
RegionApi Methods 625
RegionApi.Add Method 625
RegionApi.CanDelete Method 626
RegionApi.Delete Method 626
RegionApi.GetItem Method 626
RegionApi.GetList Method 626
RegionApi.Update Method 627
ShippingMethodApi Class 627
ShippingMethodApi Methods 627
ShippingMethodApi.Add Method 627
ShippingMethodApi.Update Method 628
TaxApi Class 628
TaxApi Methods 628
TaxApi.Add Method 629
TaxApi.Delete Method 629
TaxApi.GetApplicableTaxRate Method 629

TaxApi.GetApplicableTaxRateList Method 629
GetItem Method 630
TaxApi.GetItem Method (long) 630
TaxApi.GetItem Method (long, long) 630
TaxApi.GetItemByCountryId Method 630
GetItemByPostalCode Method 631
TaxApi.GetItemByPostalCode Method (long, string) 631
TaxApi.GetItemByPostalCode Method (long, string, int) 631
TaxApi.GetItemByPostalId Method 632
TaxApi.GetItemByRegionId Method 632
TaxApi.GetList Method 632
TaxApi.Update Method 632
TaxClassApi Class 633
TaxClassApi Methods 633
TaxClassApi.Add Method 633
TaxClassApi.Delete Method 633
TaxClassApi.GetItem Method 634
TaxClassApi.GetList Method 634
TaxClassApi.Update Method 634
TaxTypeApi Class 635
TaxTypeApi Methods 635
TaxTypeApi.Add Method 635
TaxTypeApi.Delete Method 635
TaxTypeApi.GetItem Method 636
TaxTypeApi.GetList Method 636
TaxTypeApi.Update Method 636
WarehouseApi Class 636
WarehouseApi Methods 637
WarehouseApi.Add Method 637
WarehouseApi.Delete Method 637
WarehouseApi.GetDefaultWarehouse Method 637
WarehouseApi.GetItem Method 638
WarehouseApi.GetList Method 638
WarehouseApi.MarkAsDefault Method 638
WarehouseApi.Update Method 638
Ektron.Cms.API Namespace 639
Classes 639
BusinessRule Class 640
BusinessRule Methods 641

GetBusinessRules Method 641
BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, Long) 641
BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, String) 642
CustomFields Class 642
CustomFields Methods 643
CustomFields.AddNewEditableCustomFieldAssignments Method 644
CustomFields.BoolToYesNo Method 644
CustomFields.GetEditableCustomFieldAssignments Method 644
CustomFields.GetEnhancedMetadataArea Method 645
CustomFields.GetEnhancedMetadataScript Method 645
CustomFields.GetFieldsByFolder Method 645
GetSearchProperties Method 646
CustomFields.GetSearchProperties Method (Ektron.Cms.Common.SearchAssetDispayRequest) 646
CustomFields.GetSearchProperties Method (String, String, String, String, Collection, String, String, String) 647
CustomFields.GetSearchProperties Method (System.Collections.Specialized.NameValueCollection, Ektron.Cms.Common.SearchItemCustom, Ektron.Cms.Common.SearchItemCustom) 647
CustomFields.Populate_AssetRequestObjectFromForm Method 647
CustomFields.ProcessCustomFields Method 648
CustomFields.SearchAllAssets Method 648
CustomFields.WriteFilteredMetadataForEdit Method 649
CustomFields.WriteFilteredMetadataForView Method 649
CustomFields.WriteLibrarySearchExtended Method 649
CustomFields.WriteMetadataDefaultForEdit Method 650
CustomFields.WriteMetadataTypeForView Method 650
CustomFields.WritePublicSearch Method 650
CustomFields.WritePublicSearchEx Method 651
Folder Class 652
Folder Methods 652
Folder.AddCommunityFolder Method 653
Folder.AddFolder Method 653
Folder.CreateFolder Method 654
Folder.DeleteFolderById Method 655
GetChildFolders Method 656
Folder.GetChildFolders Method (Long, Boolean, EkEnumeration.FolderOrderBy) 656
Folder.GetChildFolders Method (Long, Boolean, String) 658
GetFolder Method 659

Folder.GetFolder Method (Long) 659
Folder.GetFolder Method (Long, Boolean) 660
Folder.GetFolder Method (Long, Boolean, Boolean) 660
Folder.GetFolderDataWithPermission Method 661
GetFolderId Method 661
Folder.GetFolderId Method (Long) 661
Folder.GetFolderId Method (String, Long) 661
Folder.GetFolderInfo Method 662
Folder.GetPath Method 662
Folder.IsFolderHidden Method 662
Folder.RenameFolder Method 663
UpdateFolder Method 663
Folder.UpdateFolder Method (FolderData) 663
Folder.UpdateFolder Method (FolderRequest) 664
Folder.UpdateFolderApprovalMethod Method 664
Library Class 664
Library Methods 664
Library.AddLibraryItem Method 665
Library.AddLoadBalanceItem Method 665
Library.DeleteLibraryItem Method 666
Library.DeleteLibraryItemById Method 666
Library.DeleteLoadBalanceItem Method 666
Library.GetAllChildLibItems Method 666
GetAllLoadBalancePaths Method 668
Library.GetAllLoadBalancePaths Method (Long, String) 669
Library.GetAllLoadBalancePaths Method (String) 669
GetLibraryItem Method 669
Library.GetLibraryItem Method (Long) 670
Library.GetLibraryItem Method (Long, Long) 670
Library.GetLibrarySettings Method 670
Library.GetLibraryTypes Method 671
Library.GetLoadBalancePath Method 671
Library.InternalLibrarySearch Method 671
Library.UpdateLibrary Method 672
Library.UpdateLibrarySettings Method 672
Library.UpdateLoadBalanceSettings Method 672
Library.UpdateQlinkTemplates Method 672
Localization Class 673
Localization Methods 673

Localization.GetAddViewLanguage Method 674
Localization.GetAddViewLanguageForAllContent Method 674
Localization.GetAddViewLanguageForMenus Method 674
Localization.GetContentItems Method 675
Localization.GetFile Method 675
Localization.GetFiles Method 675
Localization.GetFilesInZipFile Method 676
GetFlagUrl Method 676
Localization.GetFlagUrl Method (Integer) 676
Localization.GetFlagUrl Method (LanguageData) 677
Localization.GetFlagUrl Method (LanguageDataSet.LanguageRow) 677
Localization.GetInterchangeFile Method 678
Localization.GetInterchangeFiles Method 678
Localization.GetInterchangeFilesInZipFile Method 678
GetJobs Method 679
Localization.GetJobs Method () 679
Localization.GetJobs Method (Long) 679
Localization.GetLocalizationUrl Method 680
Localization.GetTranslationUploadDirectory Method 680
Localization.StartExportContentForTranslation Method 680
Localization.StartExportFolderForTranslation Method 681
Localization.StartExportMenusForTranslation Method 681
Localization.StartImportTranslation Method 682
Messaging Class 682
Messaging Methods 682
Messaging.GetUnreadMessageCountForUser Method 682
Metadata Class 683
Metadata Methods 683
AddMetaDataType Method 683
Metadata.AddMetaDataType Method (Collection) 684
Metadata.AddMetaDataType Method (ContentMetaData) 684
Metadata.DeleteMetadataType Method 686
GetContentMetadataList Method 686
Metadata.GetContentMetadataList Method (Long) 686
Metadata.GetContentMetadataList Method (Long, Boolean) 688
Metadata.GetIntegratedSearchFolder Method 688
Metadata.GetIntegratedSearchFolders Method 688
Metadata.GetMetadataByFolder Method 689
Metadata.GetMetadataType Method 689

Metadata.GetMetaDataTypes Method 689
Metadata.UpdateMetadataType Method 690
Permissions Class 690
Permissions Methods 690
Permissions.AddItemPermission Method 691
Permissions.DeleteItemPermission Method 691
Permissions.DisableContentInheritance Method 691
Permissions.DisableFolderInheritance Method 692
Permissions.EnableContentInheritance Method 692
Permissions.EnableFolderInheritance Method 692
GetUserPermissions Method 692
Permissions.GetUserPermissions Method (Long, String, Long, String) 693
Permissions.GetUserPermissions Method (Long, String, Long, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) 693
Permissions.LoadPermissions Method 694
Permissions.UpdateItemPermission Method 694
Site Class 694
Site Methods 695
Site.BreakTaskPermissionInherit Method 695
Site.DeleteCookie Method 696
Site.GetAllActiveLanguages Method 696
GetAllLanguages Method 696
Site.GetAllLanguages Method () 696
Site.GetAllLanguages Method (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) 697
Site.GetClientDayNames Method 697
Site.GetClientMonthNames Method 697
Site.GetCookieValue Method 697
Site.GetLanguage Method 698
Site.GetLanguages Method 698
Site.GetSiteVariables Method 698
Site.GetTaskPermission Method 699
Site.IsTaskPermissionInherited Method 699
Site.SetCookieValue Method 699
Site.SetTaskPermission Method 700
Site.UpdateSiteEnabledLanguages Method 700
Site.UpdateSiteVariables Method 700
SiteMap Class 701
SiteMap Methods 701
SiteMap.GetFolderBreadcrumbPath Method 701

SiteMap.GetWebsiteMap Method 703
SiteMap.RenderFolderBreadcrumbHtml Method 704
SiteMap.RenderWebsiteMap Method 705
Subscription Class 705
Subscription Methods 706
Subscription.AddEmailFrom Method 706
Subscription.AddSubscription Method 707
Subscription.AddSubscriptionMessage Method 707
Subscription.DeleteEmailFrom Method 707
Subscription.DeleteSubscription Method 707
Subscription.DeleteSubscriptionMessage Method 708
Subscription.GetAllActiveSubscriptions Method 708
Subscription.GetAllEmailFrom Method 708
Subscription.GetAllSubscriptionMessages Method 708
Subscription.GetAllSubscriptions Method 709
Subscription.GetFolderInheritedFrom Method 709
Subscription.GetSubscription Method 709
Subscription.GetSubscriptionMessage Method 709
Subscription.GetSubscriptionMessagesForType Method 710
Subscription.GetSubscriptionPropertiesForContent Method 710
Subscription.GetSubscriptionPropertiesForFolder Method 710
Subscription.GetSubscriptionsForContent Method 711
Subscription.GetSubscriptionsForFolder Method 711
Subscription.SendWebAlert Method 711
Subscription.UpdateEmailFrom Method 712
Subscription.UpdateSubscription Method 712
Subscription.UpdateSubscriptionMessage Method 712
Subscription.UpdateSubscriptionPropertiesForContent Method 712
Subscription.UpdateSubscriptionPropertiesForFolder Method 713
Utilities Class 713
Utilities Methods 713
Utilities.LoadPermissions Method 714
Ektron.Cms.API.Analytics Namespace 714
Classes 714
Analytics Class 714
Analytics Methods 715
Analytics.GetSessionRank Method 715
Analytics.HardDumpAnalyticsBuffer Method 715
Analytics.QueryAnalytics Method 715

Analytics.SoftDumpAnalyticsBuffer Method 716
TrackVisit Method 716
Analytics Properties 717
Analytics.AnalyticsEnabled Property 717
Analytics.AppendToIISLog Property 717
Analytics.InMemoryContentHitsData Property 718
Analytics.RecordsBeforeWrite Property 718
Analytics.TimeBeforeWrite Property 718
Ektron.Cms.API.Community Namespace 718
Classes 719
CommunityGroup Class 719
CommunityGroup Methods 720
CommunityGroup.AcceptGroupInvite Method 721
CommunityGroup.AddCommunityGroup Method 721
AddUserToCommunityGroup Method 721
CommunityGroup.ApprovePendingGroupUser Method 722
CommunityGroup.CancelJoinRequestForCommunityGroup Method 722
CommunityGroup.DeleteCommunityGroupByID Method 723
CommunityGroup.DeletePendingGroupUser Method 723
CommunityGroup.GetAllCommunityGroups Method 723
CommunityGroup.GetAllCommunityGroupsInDirectory Method 724
CommunityGroup.GetAllUnassignedCommunityGroupUsers Method 724
CommunityGroup.GetCommunityGroupBlog Method 724
CommunityGroup.GetCommunityGroupBlogIds Method 724
CommunityGroup.GetCommunityGroupByID Method 725
CommunityGroup.GetCommunityGroupDiscussionBoardIds Method 725
CommunityGroup.GetCommunityGroups Method 725
CommunityGroup.GetCommunityGroupUserIDs Method 726
GetCommunityGroupUsers Method 726
CommunityGroup.GetFolderIdForCommunityGroup Method 727
CommunityGroup.GetGroupMemberStatus Method 727
CommunityGroup.GetInvitedCommunityGroups Method 728
GetMostPopularCommunityGroups Method 728
CommunityGroup.GetPendingCommunityGroups Method 729
CommunityGroup.GetPendingCommunityGroupUsers Method 729
CommunityGroup.InviteUsers Method 730
CommunityGroup.IsCommunityGroupAdmin Method 730
CommunityGroup.IsUserInCommunityGroup Method 730
CommunityGroup.RemoveUserFromCommunityGroup Method 731

Copyright (c) Ektron Inc. 2008

CommunityGroup.UpdateCommunityGroup Method 731
Favorites Class 731
Favorites Methods 732
AddContentFavorite Method 732
Favorites.AddFavoriteLink Method 733
Favorites.DeleteContentFavorite Method 733
Favorites.DeleteFavoriteLink Method 734
GetAllFavoriteLink Method 734
GetContentFavorites Method 735
Favorites.GetFavoriteId Method 737
Favorites.GetFavoriteLink Method 738
Favorites.GetFavoritesDirectory Method 738
Favorites.IsContentFavorite Method 738
Favorites.MoveContentFavorite Method 739
Favorites.MoveFavoriteLink Method 739
Favorites.UpdateFavoriteLink Method 739
Flagging Class 739
Flagging Methods 740
Flagging.AddFlagObject Method 740
Flagging.AssignFlagToContent Method 741
GetAllFlagEntries Method 741
GetContentFlagData Method 742
Flagging.GetDefaultFolderFlagDef Method 743
Flagging.GetFlagDataById Method 744
Flagging.GetFlaggedRank Method 744
Flagging.PurgeFlagEntries Method 744
Flagging.RemoveObjectFlag Method 744
Flagging.UpdateFlagObject Method 745
Friends Class 745
Friends Methods 745
Friends.AcceptFriendInvite Method 746
Friends.AcceptPendingFriend Method 746
AddFriend Method 746
Friends.AddPendingFriend Method 747
Friends.AddSelectedFriend Method 748
Friends.DeleteFriend Method 748
DeletePendingFriendRequest Method 748
Friends.DeleteSelectedFriend Method 749
GetFriends Method 749

Friends.GetFriendsDirectory Method 751
Friends.GetFriendStatus Method 752
Friends.GetInvitedUsers Method 752
Friends.GetPendingFriends Method 752
Friends.GetSelectedFriends Method 753
Friends.GetSentFriendInvitations Method 753
Friends.InviteUsers Method 753
Friends.IsFriend Method 754
Friends.IsSelectedFriend Method 754
Friends.MoveFriend Method 754
MessageBoard Class 755
MessageBoard Methods 755
MessageBoard.AddMessageBoardEntry Method 755
ApproveMessage Method 756
DeleteMessageBoardEntries Method 758
DeleteMessageBoardEntriesByUser Method 759
DeleteMessageBoardEntry Method 761
GetAllMessageBoardEntries Method 763
GetAllMessageBoardEntriesByUser Method 765
GetMessageBoardEntry Method 767
MessageBoard.IsUserMessageBoardAdmin Method 769
UpdateMessageText Method 769
Tags Class 771
Tags Methods 771
Tags.AddTag Method 772
AddTagToCommunityGroup Method 772
AddTagToContent Method 774
AddTagToObject Method 775
AddTagToUser Method 777
Tags.DeleteTagById Method 779
Tags.DeleteTagOnObject Method 779
Tags.DeleteTagsByUser Method 779
Tags.DeleteTagsOnCommunityGroup Method 780
Tags.DeleteTagsOnContent Method 780
Tags.DeleteTagsOnObject Method 780
Tags.DeleteTagsOnUser Method 781
Tags.EditTag Method 781
Tags.GetAllTagRelatedContent Method 781
Tags.GetAllTagRelatedUsers Method 781

GetAllTags Method 782
GetDefaultTags Method 783
Tags.GetTagByID Method 784
Tags.GetTagByName Method 784
Tags.GetTagCloud Method 785
Tags.GetTagsForContent Method 785
GetTagsForObject Method 785
GetTagsForUser Method 786
Tags.RemoveTagAsDefault Method 788
Tags.SaveTagAsDefault Method 788
Ektron.Cms.API.Content Namespace 788
Classes 789
Asset Class 789
Asset Methods 789
Asset.GetAssetMgtConfigInfo Method 790
Asset.GetAssetMgtInfo Method 790
Asset.GetAssetSupertypes Method 790
Asset.GetEditSnippet Method 790
Asset.GetMyAssetMgtID Method 791
Asset.GetViewUrl Method 791
Asset.MakeCopyDMSAsset Method 791
Asset.SetAssetMgtConfigInfo Method 792
Blog Class 792
Blog Methods 792
Blog.AddBlog Method 793
Blog.AddCommentForPost Method 794
Blog.AddPost Method 794
Blog.DeleteBlog Method 796
Blog.DeleteBlogPostComment Method 796
Blog.DeletePost Method 796
Blog.GetBlankBlogPostData Method 796
GetBlog Method 797
GetBlogData Method 800
Blog.GetBlogDataForPost Method 803
Blog.GetBlogPostData Method 803
GetBlogRoll Method 804
Blog.GetPost Method 805
Blog.GetPostbyID Method 805
Blog.GetUserBlog Method 806

Blog.UpdateBlogProperties Method 806
Content Class 806
Content Methods 806
Content.AddAsset Method 808
AddContent Method 808
Content.AddXmlConfiguration Method 813
Content.CheckOutContent Method 813
Content.CopyContentToFolder Method 814
Content.DeleteContentItem Method 814
Content.DeleteXmlConfiguration Method 815
Content.EditAsset Method 816
Content.GetAddViewLanguage Method 816
Content.GetAllComments Method 816
Content.GetAllTemplates Method 817
Content.GetAllUnassignedItemApprovals Method 817
Content.GetAllXmlConfigurations Method 817
GetChildContent Method 818
Content.GetCommunityTemplate Method 820
GetContent Method 820
Content.GetContentByHistory Method 823
Content.GetContentDisplayLink Method 823
Content.GetContentForEditing Method 823
GetContentReport Method 824
Content.GetContentState Method 825
Content.GetContentStatus Method 825
Content.GetDomain Method 825
Content.GetExpireContent Method 826
GetList Method 826
Content.MoveContentToFolder Method 827
Content.PublishContent Method 827
Content.RelicateXmlConfiguration Method 828
Content.RenameContent Method 828
Content.ReplicateXmlConfiguration Method 828
Content.RestoreHistoryContent Method 829
Content.SaveContent Method 829
Content.SubmitForDelete Method 830
Content.UndoCheckout Method 830
Content.UpdateContentMetaData Method 830
Content.UpdatexmlCollectionPackage Method 831

Content.UpdateXmlConfiguration Method 831
ContentRating Class 831
ContentRating Methods 831
ContentRating.GetContentRating Method 832
ContentRating.GetContentRatingAddRating Method 834
ContentRating.GetContentRatingResults Method 834
GetContentRatingStatistics Method 834
ContentRating.PurgeContentRatings Method 836
Form Class 836
Form Methods 836
Form.AssignFormTask Method 837
Form.EmailFormFieldData Method 837
GetAllForms Method 838
Form.GetForm Method 839
GetFormFieldData Method 839
Form.GetFormFieldList Method 840
Form.GetFormFieldQuestions Method 840
Form.GetFormTitle Method 841
Form.SubmitFormFieldData Method 841
Form.UpdateFormFieldData Method 841
Taxonomy Class 842
Taxonomy Methods 842
AddTaxonomyItem Method 843
Taxonomy.AddTaxonomySyncFolder Method 843
Taxonomy.CloneTaxonomy Method 844
Taxonomy.CreateTaxonomy Method 844
Taxonomy.DeleteTaxonomy Method 844
Taxonomy.GetTaxonomyIdByPath Method 845
ImportTaxonomy Method 845
Taxonomy.LoadTaxonomy Method 846
Taxonomy.ReadAllAssignedCategory Method 846
Taxonomy.ReadTaxonomy Method 847
Taxonomy.RemoveTaxonomyFolder Method 847
Taxonomy.RemoveTaxonomyItem Method 847
Taxonomy.ReOrderTaxonomyItems Method 847
Taxonomy.UpdateTaxonomy Method 848
ThreadedDiscussion Class 848
ThreadedDiscussion Methods 848
ThreadedDiscussion.AddBoard Method 849

ThreadedDiscussion.AddForum Method 850
AddReply Method 852
AddTopic Method 853
ThreadedDiscussion.DeleteBoard Method 854
ThreadedDiscussion.DeleteForum Method 855
ThreadedDiscussion.DeleteTopic Method 855
GetActiveTopics Method 855
ThreadedDiscussion.GetActiveTopicsForForum Method 856
ThreadedDiscussion.GetActiveUsers Method 856
ThreadedDiscussion.GetBoardCategories Method 857
ThreadedDiscussion.GetBoardForums Method 857
GetDiscussionBoard Method 857
ThreadedDiscussion.GetDiscussionBoardString Method 858
ThreadedDiscussion.GetForum Method 859
ThreadedDiscussion.GetRepliesForTopics Method 860
ThreadedDiscussion.GetTopic Method 861
ThreadedDiscussion.SearchReplies Method 861
ThreadedDiscussion.SubscribeToBoard Method 862
ThreadedDiscussion.SubscribeToForum Method 862
ThreadedDiscussion.SubscribeToTopic Method 862
ThreadedDiscussion.UnSubscribeToBoard Method 863
ThreadedDiscussion.UnSubscribeToForum Method 863
ThreadedDiscussion.UnSubscribeToTopic Method 864
Ektron.Cms.API.Search Namespace 864
Classes 865
SearchManager Class 865
SearchManager Methods 865
Execute Method 866
Search Method 867
SearchManager.SearchForCategories Method 870
SearchManager.SearchTable Method 870
Ektron.Cms.API.Task Namespace 871
Classes 871
Task Class 871
Task Methods 871
Task.AddTask Method 872
AssignTaskToUser Method 872
AssignTaskToUsergroup Method 873
DeleteTask Method 874

DeleteTasks Method 875
Task.GetTask Method 876
GetTasks Method 876
GetTasksAssignedByUser Method 877
GetTasksAssignedToUser Method 878
GetTasksAssignedToUsergroup Method 879
Task.GetTaskState Method 881
Task.SetStateForTask Method 881
Task.UpdateTask Method 881
TaskCategory Class 882
TaskCategory Methods 882
TaskCategory.AddTaskCategory Method 882
TaskCategory.DeleteTaskCategory Method 882
TaskCategory.GetAllCategories Method 883
TaskCategory.GetTaskCategory Method 883
TaskCategory.UpdateTaskCategory Method 883
TaskCategoryType Class 884
TaskCategoryType Methods 884
TaskCategoryType.AddTaskType Method 884
TaskCategoryType.DeleteTaskType Method 885
TaskCategoryType.GetTaskType Method 885
TaskCategoryType.UpdateTaskType Method 885
Ektron.Cms.API.UrlAliasing Namespace 886
Classes 886
UrlAliasAuto Class 886
UrlAliasAuto Methods 887
UrlAliasAuto.Add Method 887
UrlAliasAuto.ClearCache Method 887
UrlAliasAuto.Delete Method 888
UrlAliasAuto.GetAutoAliasNameTypes Method 888
UrlAliasAuto.GetAutoAliasTypes Method 888
UrlAliasAuto.GetFolderItem Method 888
UrlAliasAuto.GetFolderList Method 889
UrlAliasAuto.GetFolderSource Method 889
UrlAliasAuto.GetItem Method 889
UrlAliasAuto.GetItemForContent Method 890
UrlAliasAuto.GetList Method 890
UrlAliasAuto.GetListForContent Method 891
GetTarget Method 891

UrlAliasAuto.GetTaxonomyItem Method 893
UrlAliasAuto.GetTaxonomyList Method 893
UrlAliasAuto.GetTaxonomySource Method 893
UrlAliasAuto.Update Method 893
UrlAliasCommon Class 894
UrlAliasCommon Methods 894
UrlAliasCommon.GetAliasForContent Method 894
GetTarget Method 894
UrlAliasManual Class 895
UrlAliasManual Methods 896
UrlAliasManual.Add Method 896
UrlAliasManual.ClearCache Method 896
UrlAliasManual.Delete Method 896
UrlAliasManual.GetDefaultAlias Method 897
UrlAliasManual.GetFileExtensions Method 897
UrlAliasManual.GetItem Method 897
GetList Method 898
UrlAliasManual.GetSiteList Method 902
UrlAliasManual.GetTemplateList Method 902
UrlAliasManual.Update Method 902
UrlAliasManual.ValidateName Method 903
UrlAliasRegEx Class 903
UrlAliasRegEx Methods 903
UrlAliasRegEx.Add Method 904
UrlAliasRegEx.ClearCache Method 904
UrlAliasRegEx.Delete Method 904
UrlAliasRegEx.EvaluateExpression Method 904
UrlAliasRegEx.GetItem Method 905
UrlAliasRegEx.GetList Method 905
UrlAliasRegEx.GetPriorityList Method 906
UrlAliasRegEx.GetTarget Method 906
UrlAliasRegEx.Update Method 906
UrlAliasSettings Class 907
UrlAliasSettings Methods 907
UrlAliasSettings.RefreshSettingsCache Method 907
UrlAliasSettings Properties 907
UrlAliasSettings.AutoCacheDuration Property 909
UrlAliasSettings.AutoCacheSize Property 909
UrlAliasSettings.DefaultPage Property 909

UrlAliasSettings.DisableLanguageAwareness Property 909
UrlAliasSettings.EnableAutoAliasing Property 909
UrlAliasSettings.EnableAutoCaching Property 910
UrlAliasSettings.EnabledTracing Property 910
UrlAliasSettings.EnableManualAliasing Property 910
UrlAliasSettings.EnableManualCaching Property 910
UrlAliasSettings.EnableRegExAliasing Property 910
UrlAliasSettings.EnableRegExCaching Property 911
UrlAliasSettings.IsAutoAliasEnabled Property 911
UrlAliasSettings.IsManualAliasEnabled Property 911
UrlAliasSettings.IsRegExAliasEnabled Property 911
UrlAliasSettings.ManualCacheDuration Property 912
UrlAliasSettings.ManualCacheSize Property 912
UrlAliasSettings.OverrideTemplate Property 912
UrlAliasSettings.PageExtensions Property 912
UrlAliasSettings.QueryStringAction Property 912
UrlAliasSettings.RegExCacheDuration Property 912
UrlAliasSettings.RegExCacheSize Property 913
Ektron.Cms.API.User Namespace 913
Classes 913
ADUser Class 913
ADUser Methods 913
ADUser.AddADGroupToCMS Method 914
ADUser.AddADUsersToCMSByUsername Method 914
ADUser.DeSynchUserGroups Method 915
ADUser.DeSynchUsers Method 915
ADUser.GetADMapping Method 915
ADUser.GetADStatus Method 916
ADUser.GetAvailableADGroups Method 916
ADUser.GetAvailableADUsers Method 916
ADUser.GetCMSGroupsToSync Method 917
ADUser.GetCMSRelationshipsToSync Method 917
ADUser.GetCMSUsersToSync Method 917
ADUser.GetDomains Method 918
ADUser.MapCMSUserGroupToAD Method 918
ADUser.MapCMSUserToAD Method 918
ADUser.SynchCMSGroupsToAD Method 919
ADUser.SynchCMSRelationshipsToAD Method 919
ADUser.SynchCMSUsersToAD Method 919

ADUser.UpdateADMapping Method 920
ADUser.UpdateADUsersGroups Method 920
ADUser.UpdateUsersInfoFromAD Method 920
User Class 920
User Methods 921
ActivateUserAccount Method 922
User.AddMembershipUser Method 923
User.AddNewUser Method 925
User.AddRolePermission Method 926
AddUser Method 926
User.AddUserGroup Method 927
User.AddUserToGroup Method 927
User.AutoLogInUser Method 928
CreateBlankUserObject Method 928
User.DeleteUserByID Method 929
User.DeleteUserByIds Method 929
User.DeleteUserFromGroup Method 929
User.DeleteUserGroup Method 930
User.EditUserCustomProperties Method 930
User.GetActiveUser Method 930
User.GetADMapping Method 931
User.GetAllCmsUsers Method 931
User.GetAllCustomProperties Method 931
User.GetAllUserByCustomProperty Method 932
GetAllUserGroups Method 932
GetAllUsers Method 933
User.GetGroupsUserIsIn Method 934
User.GetMembershipUser Method 934
User.GetUser Method 935
User.GetUserByUsername Method 935
User.GetUserCustomProperties Method 937
User.GetUserEmailInfo Method 937
User.GetUserForFolderAdmin Method 937
User.GetUserGroup Method 937
User.GetUserGroupByName Method 938
User.GetUserGroupForFolderAdmin Method 938
User.GetUserPermissions Method 938
User.GetUserPreference Method 939
GetUsers Method 939

User.GetUsersForTask Method 940
User.GetUsersNotInGroup Method 941
User.IsAdmin Method 941
User.IsAGroupMember Method 942
IsARoleMember Method 942
User.IsARoleMemberForContent Method 945
User.IsARoleMemberForContent_FolderUserAdmin Method 945
User.IsARoleMemberForFolder Method 946
User.IsARoleMemberForFolder_FolderUserAdmin Method 946
User.IsARoleMemberForFolder_MetadataAdmin Method 946
LogInUser Method 947
User.ReadCustomProperties Method 950
ResetMembershipUserPassword Method 950
User.SetAuthenticationCookie Method 951
User.UnsubscribeUser Method 951
User.UpdateCustomPropertiesItemOrder Method 952
UpdateUser Method 952
UpDateUserGroup Method 953
User.UpdateUserPreferences Method 954
UserObject Method 954
User Properties 955
User.UserIdAuthenticated Property 955

**Web Services API 955**
Ektron.Services Namespace 955
Ektron.Services.Task Namespace 956
Classes 956
Task Class 957
Task Methods 957
Task.AddTask Method 957
AssignTaskToUser Method 958
AssignTaskToUsergroup Method 959
DeleteTask Method 960
DeleteTasks Method 961
Task.GetTask Method 962
Task.GetTasks Method 963
GetTasksAssignedByUser Method 963
GetTasksAssignedToUser Method 964
GetTasksAssignedToUsergroup Method 965
Task.GetTaskState Method 967

Task.SetStateForTask Method 967
Task.UpdateTask Method 967
TaskCategory Class 968
TaskCategory Methods 968
TaskCategory.AddTaskCategory Method 969
TaskCategory.DeleteTaskCategory Method 969
TaskCategory.GetAllCategories Method 969
TaskCategory.GetTaskCategory Method 970
TaskCategory.UpdateTaskCategory Method 970
TaskCategoryType Class 971
TaskCategoryType Methods 971
TaskCategoryType.AddTaskType Method 971
TaskCategoryType.DeleteTaskType Method 972
TaskCategoryType.GetTaskType Method 972
TaskCategoryType.UpdateTaskType Method 972
Classes 973
ActiveDirectoryUser Class 974
ActiveDirectoryUser Methods 974
ActiveDirectoryUser.AddADGroupToCMS Method 975
ActiveDirectoryUser.DeSynchUserGroups Method 975
ActiveDirectoryUser.DeSynchUsers Method 975
ActiveDirectoryUser.GetADMapping Method 976
ActiveDirectoryUser.GetADStatus Method 976
ActiveDirectoryUser.GetAvailableADGroups Method 977
ActiveDirectoryUser.GetAvailableADUsers Method 977
ActiveDirectoryUser.GetCMSGroupsToSync Method 978
ActiveDirectoryUser.GetCMSRelationshipsToSync Method 978
ActiveDirectoryUser.GetCMSUsersToSync Method 979
ActiveDirectoryUser.GetDomains Method 979
ActiveDirectoryUser.MapCMSUserGroupToAD Method 980
ActiveDirectoryUser.MapCMSUserToAD Method 980
ActiveDirectoryUser.UpdateADMapping Method 981
ActiveDirectoryUser.UpdateADUsersGroups Method 981
ActiveDirectoryUser.UpdateUsersInfoFromAD Method 981
Asset Class 982
Asset Methods 982
Asset.GetAssetMgtConfigInfo Method 983
Asset.GetAssetMgtInfo Method 983
Asset.GetAssetSupertypes Method 983

Asset.GetEditSnippet Method 984
Asset.GetMyAssetMgtID Method 984
Asset.MakeCopyDMSAsset Method 984
Asset.SetAssetMgtConfigInfo Method 985
Blog Class 985
Blog Methods 986
Blog.AddBlog Method 986
Blog.AddCommentForPost Method 987
Blog.AddPost Method 987
Blog.DeleteBlog Method 988
Blog.DeleteBlogPostComment Method 988
Blog.DeletePost Method 988
Blog.GetBlankBlogPostData Method 989
GetBlog Method 989
Blog.GetBlog Method (long, string, int, int) 990
Blog.GetBlog Method (string, string, int, int) 991
GetBlogData Method 991
Blog.GetBlogData Method (FolderData) 992
Blog.GetBlogData Method (long) 993
Blog.GetBlogData Method (long, DateTime) 993
Blog.GetBlogData Method (long, DateTime, int, int) 994
Blog.GetBlogDataForPost Method 995
Blog.GetBlogPostData Method 995
GetBlogRoll Method 995
Blog.GetBlogRoll Method (long) 996
Blog.GetBlogRoll Method (string) 996
Blog.GetPost Method 997
Blog.GetPostbyID Method 997
Blog.GetUserBlog Method 998
Blog.UpdateBlogProperties Method 998
BusinessRules Class 999
BusinessRules Methods 999
GetBusinessRules Method 999
BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, int) 1000
BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, string) 1000
CommunityGroup Class 1001
CommunityGroup Methods 1001
CommunityGroup.AddCommunityGroup Method 1002

AddUserToCommunityGroup Method 1002
CommunityGroup.AddUserToCommunityGroup Method (long, long) 1003
CommunityGroup.AddUserToCommunityGroup Method (long, long, bool) 1003
CommunityGroup.ApprovePendingGroupUser Method 1004
CommunityGroup.CancelJoinRequestForCommunityGroup Method 1004
CommunityGroup.DeleteCommunityGroupByID Method 1004
CommunityGroup.DeletePendingGroupUser Method 1005
CommunityGroup.GetAllCommunityGroups Method 1005
CommunityGroup.GetAllCommunityGroupsInDirectory Method 1006
CommunityGroup.GetAllUnassignedCommunityGroupUsers Method 1006
CommunityGroup.GetCommunityGroupBlogIds Method 1006
CommunityGroup.GetCommunityGroupByID Method 1007
CommunityGroup.GetCommunityGroupDiscussionBoardIds Method 1007
CommunityGroup.GetCommunityGroups Method 1008
CommunityGroup.GetCommunityGroupUserIDs Method 1008
GetCommunityGroupUsers Method 1009
CommunityGroup.GetCommunityGroupUsers Method (long) 1009
CommunityGroup.GetCommunityGroupUsers Method (long, PagingInfo) 1009
CommunityGroup.GetFolderIdForCommunityGroup Method 1010
CommunityGroup.GetGroupMemberStatus Method 1010
CommunityGroup.GetInvitedCommunityGroups Method 1011
GetMostPopularCommunityGroups Method 1011
CommunityGroup.GetMostPopularCommunityGroups Method (PagingInfo) 1012
CommunityGroup.GetMostPopularCommunityGroups Method (int) 1012
CommunityGroup.GetPendingCommunityGroups Method 1013
CommunityGroup.GetPendingCommunityGroupUsers Method 1013
CommunityGroup.InviteUsers Method 1014
CommunityGroup.IsCommunityGroupAdmin Method 1014
CommunityGroup.IsUserInCommunityGroup Method 1014
CommunityGroup.RemoveUserFromCommunityGroup Method 1015
CommunityGroup.UpdateCommunityGroup Method 1015
Content Class 1016
Content Methods 1016
AddContent Method 1017
Content.AddContent Method (string, string, string, string, string, string, long, object, object, string) 1017
Content.AddContent Method (string, string, string, string, string, string, long, object, object, string, int) 1018
Content.AddContent Method (string, string, string, string, string, string, long, object, object, string, int, int) 1019

Content.CheckOutContent Method 1019
Content.CopyContentToFolder Method 1020
Content.DeleteContentItem Method 1020
Content.GetAddViewLanguage Method 1021
Content.GetAllComments Method 1021
Content.GetAllTemplates Method 1021
Content.GetAllUnassignedItemApprovals Method 1022
GetChildContent Method 1022
Content.GetChildContent Method (long, bool, string) 1022
Content.GetChildContent Method (long, bool, string, PagingInfo) 1023
Content.GetContent Method 1024
Content.GetContentByHistory Method 1024
Content.GetContentForEditing Method 1025
Content.GetContentState Method 1025
Content.GetContentStatus Method 1025
Content.GetDomain Method 1026
Content.MoveContentToFolder Method 1026
Content.PublishContent Method 1026
Content.RenameContent Method 1027
Content.SaveContent Method 1027
Content.SubmitForDelete Method 1028
Content.UndoCheckout Method 1028
Content.UpdateContentMetaData Method 1029
ContentRating Class 1029
ContentRating Methods 1029
ContentRating.AddContentRating Method 1030
ContentRating.GetContentRatingResults Method 1030
ContentRating.PurgeContentRatings Method 1030
CustomFields Class 1031
CustomFields Methods 1031
CustomFields.AddNewEditableCustomFieldAssignments Method 1032
CustomFields.BoolToYesNo Method 1033
CustomFields.GetEditableCustomFieldAssignments Method 1033
CustomFields.GetEnhancedMetadataArea Method 1034
CustomFields.GetEnhancedMetadataScript Method 1034
CustomFields.GetFieldsByFolder Method 1035
CustomFields.GetSearchProperties Method 1035
CustomFields.ProcessCustomFields Method 1035
CustomFields.SearchAllAssets Method 1036

CustomFields.WriteFilteredMetadataForEdit Method 1036
CustomFields.WriteFilteredMetadataForView Method 1037
CustomFields.WriteLibrarySearchExtended Method 1037
CustomFields.WriteMetadataDefaultForEdit Method 1037
CustomFields.WriteMetadataTypeForView Method 1038
CustomFields.WritePublicSearch Method 1038
Favorites Class 1039
Favorites Methods 1039
AddContentFavorite Method 1039
Favorites.AddContentFavorite Method (long, long) 1040
Favorites.AddContentFavorite Method (long, long, long) 1040
Favorites.DeleteContentFavorite Method 1041
GetContentFavorites Method 1041
Favorites.GetContentFavorites Method (long, PagingInfo) 1041
Favorites.GetContentFavorites Method (long, long) 1042
Favorites.GetContentFavorites Method (long, long, PagingInfo) 1042
Favorites.GetFavoritesDirectory Method 1043
Favorites.IsContentFavorite Method 1043
Favorites.MoveContentFavorite Method 1044
Flag Class 1044
Flag Methods 1045
Flag.AddFlagObject Method 1045
Flag.AssignFlagToContent Method 1045
Flag.GetAllFlagEntries Method 1046
Flag.GetContentFlagData Method 1046
Flag.GetFlagDataById Method 1047
Flag.GetFlaggedRank Method 1047
Flag.PurgeFlagEntries Method 1047
Flag.RemoveObjectObject Method 1048
Flag.UpdateFlagObject Method 1048
Folder Class 1048
Folder Methods 1049
Folder.AddFolder Method 1049
Folder.CreateFolder Method 1050
Folder.DeleteFolder Method 1050
GetChildFolders Method 1050
Folder.GetChildFolders Method (long, bool, Ektron.Cms.Common.EkEnumeration.FolderOrderBy) 1051
Folder.GetChildFolders Method (long, bool, string) 1051

GetFolder Method 1051
Folder.GetFolder Method (long) 1052
Folder.GetFolder Method (long, bool, bool) 1052
GetFolderID Method 1053
Folder.GetFolderID Method (long) 1053
Folder.GetFolderID Method (string, long) 1053
Folder.GetPath Method 1054
Folder.RenameFolder Method 1054
UpdateFolder Method 1054
Folder.UpdateFolder Method (FolderData) 1055
Folder.UpdateFolder Method (FolderRequest) 1055
Font Class 1055
Font Methods 1056
Font.AddFont Method 1056
Font.DeleteFont Method 1056
Font.GetAllFonts Method 1057
Font.GetFont Method 1057
Font.GetFontConfigList Method 1057
Font.UpdateFont Method 1058
Form Class 1058
Form Methods 1058
Form.AssignFormTask Method 1059
Form.EmailFormFieldData Method 1059
GetAllForms Method 1060
Form.GetAllForms Method () 1060
Form.GetAllForms Method (long, string, int, ContentAPI.FormResultType) 1061
Form.GetForm Method 1061
Form.GetFormDataHistogram Method 1062
GetFormFieldData Method 1062
Form.GetFormFieldData Method (long, long) 1062
Form.GetFormFieldData Method (long, string, string) 1063
Form.GetFormFieldList Method 1063
Form.GetFormFieldQuestions Method 1064
Form.GetFormTitle Method 1064
Form.SubmitFormFieldData Method 1065
Form.UpdateFormFieldData Method 1065
Friends Class 1066
Friends Methods 1066
Friends.AcceptPendingFriend Method 1067

AddFriend Method 1067
Friends.AddFriend Method (long, long) 1067
Friends.AddFriend Method (long, long, long) 1068
Friends.AddPendingFriend Method 1068
Friends.AddSelectedFriend Method 1069
Friends.DeleteFriend Method 1069
DeletePendingFriendRequest Method 1069
Friends.DeletePendingFriendRequest Method (long, long) 1070
Friends.DeletePendingFriendRequest Method (long, string) 1070
Friends.DeleteSelectedFriend Method 1071
GetFriends Method 1071
Friends.GetFriends Method (long) 1072
Friends.GetFriends Method (long, PagingInfo) 1072
Friends.GetFriends Method (long, PagingInfo, long) 1072
Friends.GetFriends Method (long, long) 1073
Friends.GetFriendsDirectory Method 1073
Friends.GetFriendStatus Method 1074
Friends.GetInvitedUsers Method 1074
Friends.GetPendingFriends Method 1075
Friends.GetSelectedFriends Method 1075
Friends.GetSentFriendInvitations Method 1076
Friends.InviteUsers Method 1076
Friends.IsFriend Method 1077
Friends.IsSelectedFriend Method 1077
Friends.MoveFriend Method 1077
Library Class 1078
Library Methods 1078
Library.AddLoadBalanceItem Method 1079
Library.DeleteLibraryItem Method 1079
Library.DeleteLibraryItemById Method 1079
Library.DeleteLoadBalanceItem Method 1080
Library.GetAllChildLibItems Method 1080
GetAllLoadBalancePaths Method 1081
Library.GetAllLoadBalancePaths Method (long, string) 1081
Library.GetAllLoadBalancePaths Method (string) 1082
GetLibraryItem Method 1082
Library.GetLibraryItem Method (long) 1083
Library.GetLibraryItem Method (long, long) 1083
Library.GetLibrarySettings Method 1083

Library.GetLibraryTypes Method 1084
Library.GetLoadBalancePath Method 1084
Library.InternalLibrarySearch Method 1085
Library.UpdateLibrary Method 1085
Library.UpdateLibrarySettings Method 1085
Library.UpdateLoadBalanceSettings Method 1086
Library.UpdateQlinkTemplates Method 1086
MessageBoard Class 1086
MessageBoard Methods 1087
MessageBoard.AddMessageBoardEntry Method 1087
ApproveMessage Method 1088
MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType) 1088
MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType, long) 1089
MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool) 1089
DeleteMessageBoardEntries Method 1090
MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType) 1090
MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, long) 1091
MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, long, bool) 1091
DeleteMessageBoardEntriesByUser Method 1092
MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long) 1092
MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, long) 1093
MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, long, bool) 1093
DeleteMessageBoardEntry Method 1094
MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType) 1094
MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long) 1095
MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool) 1095
GetAllMessageBoardEntries Method 1096
MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo) 1097
MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long) 1097

MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long, bool) 1098
GetAllMessageBoardEntriesByUser Method 1098
MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo) 1099
MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long) 1099
MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long, bool) 1100
GetMessageBoardEntry Method 1101
MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType) 1101
MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long) 1102
MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool) 1102
MessageBoard.IsUserMessageBoardAdmin Method 1103
UpdateMessageText Method 1103
MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string) 1104
MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string, long) 1104
MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string, long, bool) 1105
Metadata Class 1106
Metadata Methods 1106
Metadata.AddMetaDataType Method 1106
Metadata.DeleteMetadataType Method 1107
GetContentMetadataList Method 1107
Metadata.GetContentMetadataList Method (long) 1107
Metadata.GetContentMetadataList Method (long, bool) 1108
Metadata.GetMetadataType Method 1108
Metadata.GetMetaDataTypes Method 1109
Metadata.UpdateMetadataType Method 1109
Permissions Class 1110
Permissions Methods 1110
Permissions.AddItemPermission Method 1110
Permissions.DeleteItemPermission Method 1111
Permissions.DisableContentInheritance Method 1111
Permissions.DisableFolderInheritance Method 1112
Permissions.EnableContentInheritance Method 1112
Permissions.EnableFolderInheritance Method 1113

GetUserPermissions Method 1113
Permissions.GetUserPermissions Method (long, string, long, string) 1113
Permissions.GetUserPermissions Method (long, string, long, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) 1114
Permissions.LoadPermissions Method 1114
Permissions.UpdateItemPermission Method 1115
SearchManager Class 1115
SearchManager Methods 1116
ExecuteSearch Method 1116
SearchManager.ExecuteSearch Method (ContentSearchCondition) 1116
SearchManager.ExecuteSearch Method (UserSearchCondition) 1117
Search Method 1117
SearchManager.Search Method (MapRequestData, int, int, int) 1118
SearchManager.Search Method (SearchRequestData, int) 1118
SearchManager.Search Method (string, int) 1119
SearchManager.SearchForCategories Method 1119
SearchManager.SearchTable Method 1119
Site Class 1120
Site Methods 1120
Site.BreakTaskPermissionInherit Method 1121
Site.GetAllActiveLanguages Method 1121
GetAllLanguages Method 1122
Site.GetAllLanguages Method () 1122
Site.GetAllLanguages Method (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) 1122
Site.GetClientDayNames Method 1123
Site.GetClientMonthNames Method 1123
Site.GetCookieValue Method 1123
Site.GetLanguage Method 1124
Site.GetLanguages Method 1124
Site.GetSiteVariables Method 1125
Site.GetTaskPermission Method 1125
Site.IsTaskPermissionInherited Method 1126
Site.SetCookieValue Method 1126
Site.SetTaskPermission Method 1126
Site.UpdateSiteEnabledLanguages Method 1127
SiteMap Class 1127
SiteMap Methods 1128
SiteMap.GetFolderBreadcrumbPath Method 1128
SiteMap.GetWebsiteMap Method 1128

SiteMap.RenderFolderBreadcrumbHtml Method 1129
SiteMap.RenderWebsiteMap Method 1129
Tags Class 1130
Tags Methods 1130
Tags.AddTag Method 1131
AddTagToCommunityGroup Method 1132
Tags.AddTagToCommunityGroup Method (string, long) 1132
Tags.AddTagToCommunityGroup Method (string, long, int) 1133
Tags.AddTagToCommunityGroup Method (string, long, long, int) 1133
AddTagToContent Method 1134
Tags.AddTagToContent Method (string, long) 1134
Tags.AddTagToContent Method (string, long, long, int) 1135
AddTagToObject Method 1135
Tags.AddTagToObject Method (string, long, EkEnumeration.CMSObjectTypes) 1136
Tags.AddTagToObject Method (string, long, EkEnumeration.CMSObjectTypes, long, int) 1136
AddTagToUser Method 1137
Tags.AddTagToUser Method (long, long) 1137
Tags.AddTagToUser Method (string, long) 1138
Tags.AddTagToUser Method (string, long, int) 1138
Tags.DeleteTagById Method 1139
Tags.DeleteTagOnObject Method 1139
Tags.DeleteTagsByUser Method 1140
Tags.DeleteTagsOnCommunityGroup Method 1140
Tags.DeleteTagsOnContent Method 1141
Tags.DeleteTagsOnObject Method 1141
Tags.DeleteTagsOnUser Method 1142
Tags.EditTag Method 1142
Tags.GetAllTagRelatedContent Method 1142
Tags.GetAllTagRelatedUsers Method 1143
GetAllTags Method 1143
Tags.GetAllTags Method (PagingInfo) 1144
Tags.GetAllTags Method (TagRequestData, int) 1144
GetDefaultTags Method 1145
Tags.GetDefaultTags Method (EkEnumeration.CMSObjectTypes) 1145
Tags.GetDefaultTags Method (EkEnumeration.CMSObjectTypes, int) 1146
Tags.GetTagByID Method 1146
Tags.GetTagByName Method 1146
Tags.GetTagCloud Method 1147
Tags.GetTagsForContent Method 1147

GetTagsForObject Method 1148
Tags.GetTagsForObject Method (long, EkEnumeration.CMSObjectTypes) 1148
Tags.GetTagsForObject Method (long, EkEnumeration.CMSObjectTypes, int) 1149
GetTagsForUser Method 1149
Tags.GetTagsForUser Method (long) 1150
Tags.GetTagsForUser Method (long, int) 1150
Tags.RemoveTagAsDefault Method 1150
Tags.SaveTagAsDefault Method 1151
Taxonomy Class 1151
Taxonomy Methods 1152
AddTaxonomyItem Method 1152
Taxonomy.AddTaxonomyItem Method (TaxonomyContentRequest) 1153
Taxonomy.AddTaxonomyItem Method (TaxonomyRequest) 1153
Taxonomy.AddTaxonomySyncFolder Method 1153
Taxonomy.CreateTaxonomy Method 1154
Taxonomy.DeleteTaxonomy Method 1154
Taxonomy.GetTaxonomyIdByPath Method 1154
Taxonomy.LoadTaxonomy Method 1155
Taxonomy.ReadAllAssignedCategory Method 1155
Taxonomy.ReadTaxonomy Method 1156
Taxonomy.RemoveTaxonomyFolder Method 1156
Taxonomy.RemoveTaxonomyItem Method 1156
Taxonomy.ReOrderTaxonomyItems Method 1157
Taxonomy.UpdateTaxonomy Method 1157
ThreadedDiscussion Class 1157
ThreadedDiscussion Methods 1158
ThreadedDiscussion.AddBoard Method 1158
ThreadedDiscussion.AddForum Method 1159
AddTopic Method 1160
ThreadedDiscussion.AddTopic Method (long, string, string) 1160
ThreadedDiscussion.AddTopic Method (long, string, string, int) 1160
ThreadedDiscussion.DeleteBoard Method 1161
ThreadedDiscussion.DeleteForum Method 1161
ThreadedDiscussion.DeleteTopic Method 1162
ThreadedDiscussion.GetBoardCategories Method 1162
ThreadedDiscussion.GetBoardForums Method 1162
GetDiscussionBoard Method 1163
ThreadedDiscussion.GetDiscussionBoard Method (long) 1163
ThreadedDiscussion.GetDiscussionBoard Method (string) 1164

ThreadedDiscussion.GetDiscussionBoardString Method 1164
ThreadedDiscussion.GetForum Method 1165
ThreadedDiscussion.GetRepliesForTopics Method 1165
ThreadedDiscussion.GetTopic Method 1166
ThreadedDiscussion.SearchReplies Method 1166
User Class 1167
User Methods 1167
ActivateUserAccount Method 1169
User.ActivateUserAccount Method (long) 1169
User.ActivateUserAccount Method (string, string) 1169
User.AddMembershipUser Method 1170
User.AddNewUser Method 1170
User.AddRolePermission Method 1171
User.AddUser Method 1171
User.AddUserGroup Method 1172
User.AddUserToGroup Method 1172
User.AutoLogInUser Method 1172
CreateBlankUserObject Method 1173
User.CreateBlankUserObject Method () 1173
User.CreateBlankUserObject Method (UserData) 1174
User.DeleteUserByID Method 1174
User.DeleteUserByIds Method 1174
User.DeleteUserFromGroup Method 1175
User.DeleteUserGroup Method 1175
User.EditUserCustomProperties Method 1176
User.GetActiveUser Method 1176
User.GetADMapping Method 1176
User.GetAllCmsUsers Method 1177
User.GetAllCustomProperties Method 1178
User.GetAllUserByCustomProperty Method 1178
GetAllUserGroups Method 1179
User.GetAllUserGroups Method (EkEnumeration.UserTypes, string) 1179
User.GetAllUserGroups Method (string) 1180
GetAllUsers Method 1180
User.GetAllUsers Method (UserRequestData) 1181
User.GetAllUsers Method (long, long) 1181
User.GetGroupsUserIsIn Method 1181
User.GetMembershipUser Method 1182
User.GetUser Method 1182

User.GetUserbyUsername Method 1183
User.GetUserCustomProperties Method 1183
User.GetUserEmailInfo Method 1184
User.GetUserForFolderAdmin Method 1184
User.GetUserGroup Method 1185
User.GetUserGroupForFolderAdmin Method 1185
User.GetUserPermissions Method 1185
User.GetUserPreference Method 1186
GetUsers Method 1187
User.GetUsers Method (long, string) 1187
User.GetUsers Method (string, string) 1188
User.GetUsersForTask Method 1188
User.GetUsersNotInGroup Method 1189
User.IsAdmin Method 1189
IsARoleMember Method 1190
User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) 1191
User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, long, bool) 1191
User.IsARoleMember Method (long) 1192
User.IsARoleMember Method (long, long, bool) 1192
User.IsARoleMemberForContent Method 1193
User.IsARoleMemberForFolder Method 1193
User.LogInUser Method 1194
ResetMembershipUserPassword Method 1194
User.ResetMembershipUserPassword Method (string) 1195
User.ResetMembershipUserPassword Method (string, string) 1195
User.UnsubscribeUser Method 1195
User.UpdateCustomPropertiesItemOrder Method 1196
User.UpdateMapCoordinate Method 1196
UpdateUser Method 1197
User.UpdateUser Method (Collection) 1197
User.UpdateUser Method (UserData) 1198
User.UpDateUserGroup Method 1198
User.UpdateUserPreferences Method 1198
UserObject Method 1199
User.UserObject Method (long) 1199
User.UserObject Method (string) 1200

**Data Classes 1201**

**Ektron.Cms.Commerce Namespace** **1201**
Classes 1204
AddressData Class 1206
AddressData Properties 1206
AddressData.AddressLine1 Property 1207
AddressData.AddressLine2 Property 1207
AddressData.City Property 1207
AddressData.Company Property 1207
AddressData.Country Property 1207
AddressData.Id Property 1208
AddressData.Name Property 1208
AddressData.Phone Property 1208
AddressData.PostalCode Property 1208
AddressData.Region Property 1208
AdjustedBasketItem Class 1208
AdjustedBasketItem Constructor 1209
AdjustedBasketItem.AdjustedBasketItem Constructor () 1209
AdjustedBasketItem.AdjustedBasketItem Constructor (BasketItem) 1209
AdjustedBasketItem Properties 1210
AdjustedBasketItem.AdjustedTotal Property 1210
AdjustedBasketItem.ApplicableTaxRate Property 1210
AdjustedBasketItem.AppliedCoupons Property 1210
AdjustedBasketItem.HasFreeShipping Property 1210
AdjustedBasketItem.TaxesApplied Property 1211
AdjustedBasketItem.TotalCouponDiscount Property 1211
AppliedCouponData Class 1211
AppliedCouponData Constructor 1211
AppliedCouponData.AppliedCouponData Constructor () 1212
AppliedCouponData.AppliedCouponData Constructor (CouponData, decimal) 1212
AppliedCouponData Properties 1212
AppliedCouponData.Coupon Property 1212
AppliedCouponData.CouponDiscount Property 1212
AppliedCouponDataCollection Class 1213
AppliedCouponDataCollection Fields 1213
AppliedCouponDataCollection.ItemAdded Field 1213
AppliedCouponDataCollection Methods 1213
AppliedCouponDataCollection.FindByCouponId Method 1213
AppliedCouponEventArgs Class 1214
AppliedCouponEventArgs Constructor 1214

AppliedCouponEventArgs.AppliedCouponEventArgs Constructor () 1214
AppliedCouponEventArgs.AppliedCouponEventArgs Constructor (AppliedCouponData) 1214
AppliedCouponEventArgs Properties 1215
AppliedCouponEventArgs.AppliedCoupon Property 1215
Basket Class 1215
Basket Methods 1215
AddProduct Method 1216
Basket.AddProduct Method (long) 1216
Basket.AddProduct Method (long, KitConfigData) 1216
Basket.AddProduct Method (long, int) 1217
Basket.AddProductVariant Method 1217
Basket.Decrement Method 1217
Basket.GetItemIndex Method 1217
Basket.GetKitConfiguration Method 1218
Basket.GetKitConfigurationAsAdmin Method 1218
Basket.IncrementItemQuantity Method 1218
Basket.MakeDefaultBasket Method 1218
Basket.RemoveItem Method 1218
Basket.SaveBasket Method 1219
Basket.UpdateItemQuantity Method 1219
Basket.UpdateKitConfiguration Method 1219
Basket Properties 1219
Basket.RequestInformation Property 1220
BasketCalculatorData Class 1220
BasketCalculatorData Constructor 1220
BasketCalculatorData.BasketCalculatorData Constructor () 1220
BasketCalculatorData.BasketCalculatorData Constructor (BasketData) 1221
BasketCalculatorData Properties 1221
BasketCalculatorData.AdjustedItems Property 1221
BasketCalculatorData.Basket Property 1221
BasketCalculatorData.BasketCoupons Property 1222
BasketCalculatorData.BasketTotal Property 1222
BasketCalculatorData.HasFreeShipping Property 1222
BasketCalculatorData.ShipFrom Property 1222
BasketCalculatorData.ShippingRates Property 1222
BasketCalculatorData.ShipTo Property 1222
BasketCalculatorData.TaxClassSubtotals Property 1223
BasketCalculatorData.TotalCouponDiscount Property 1223
BasketCalculatorData.TotalShippingCost Property 1223

BasketCalculatorData.TotalTaxes Property 1223
BasketCalculatorData.ValidationResults Property 1223
BasketData Class 1224
BasketData Properties 1224
BasketData.DateCreated Property 1224
BasketData.DateModified Property 1224
BasketData.Id Property 1225
BasketData.IsDefault Property 1225
BasketData.Items Property 1225
BasketData.Name Property 1225
BasketData.ShippingAddressId Property 1225
BasketData.ShippingMethodId Property 1226
BasketData.ShopperId Property 1226
BasketData.Subtotal Property 1226
BasketData.UserId Property 1226
BasketItem Class 1226
BasketItem Constructor 1227
BasketItem.BasketItem Constructor () 1227
BasketItem.BasketItem Constructor (BasketItem) 1227
BasketItem Properties 1227
BasketItem.AdjustedTotal Property 1228
BasketItem.Configuration Property 1228
BasketItem.Dimensions Property 1228
BasketItem.Id Property 1228
BasketItem.IsTangible Property 1228
BasketItem.ListPrice Property 1229
BasketItem.Pricing Property 1229
BasketItem.ProductId Property 1229
BasketItem.ProductLanguage Property 1229
BasketItem.ProductSku Property 1229
BasketItem.ProductTitle Property 1230
BasketItem.ProductType Property 1230
BasketItem.Quantity Property 1230
BasketItem.Quicklink Property 1230
BasketItem.SalePrice Property 1230
BasketItem.TaxClassId Property 1231
BasketItem.Variant Property 1231
BasketItem.Weight Property 1231
BasketItemCollection Class 1231

BasketItemCollection.BasketItemCollection Constructor 1231
BasketItemCollection Methods 1232
BasketItemCollection.AddRange Method 1232
BasketItemCollection.Find Method 1232
BasketItemCollection.FindAll Method 1232
BasketItemCollection.FindIndex Method 1232
BasketVariantData Class 1233
BasketVariantData.BasketVariantData Constructor 1233
BasketVariantData Properties 1233
BasketVariantData.Dimensions Property 1233
BasketVariantData.Id Property 1234
BasketVariantData.Language Property 1234
BasketVariantData.ListPrice Property 1234
BasketVariantData.Pricing Property 1234
BasketVariantData.SalePrice Property 1235
BasketVariantData.Sku Property 1235
BasketVariantData.TaxClassId Property 1235
BasketVariantData.Title Property 1235
BasketVariantData.Weight Property 1235
BundleData Class 1236
BundleData.BundleData Constructor 1236
BundleData Properties 1236
BundleData.BundledItems Property 1236
CatalogData Class 1236
CatalogData Properties 1237
CatalogData.Id Property 1237
CatalogData.Name Property 1237
ContractData Class 1237
ContractData Constructor 1238
ContractData.ContractData Constructor () 1238
ContractData.ContractData Constructor (string) 1238
ContractData Properties 1238
ContractData.Id Property 1238
ContractData.Name Property 1239
CountryData Class 1239
CountryData Constructor 1239
CountryData.CountryData Constructor () 1239
CountryData.CountryData Constructor (int, string, string, string, bool) 1240
CountryData Methods 1240

CountryData.ToString Method 1240
CountryData.Validate Method 1240
CountryData Properties 1241
CountryData.Enabled Property 1241
CountryData.Id Property 1241
CountryData.LongIsoCode Property 1241
CountryData.Name Property 1241
CountryData.ShortIsoCode Property 1241
CountryTaxRateData Class 1242
CountryTaxRateData Constructor 1242
CountryTaxRateData.CountryTaxRateData Constructor () 1242
CountryTaxRateData.CountryTaxRateData Constructor (long, long, decimal) 1243
CountryTaxRateData.CountryTaxRateData Constructor (long, long, decimal, long) 1243
CountryTaxRateData Properties 1243
CountryTaxRateData.CountryData Property 1243
CouponData Class 1243
CouponData.CouponData Constructor 1244
CouponData Properties 1244
CouponData.Code Property 1244
CouponData.CouponType Property 1245
CouponData.CurrencyId Property 1245
CouponData.Description Property 1245
CouponData.DiscountType Property 1245
CouponData.DiscountValue Property 1245
CouponData.ExpirationDate Property 1245
CouponData.Id Property 1246
CouponData.IsActive Property 1246
CouponData.IsCombinable Property 1246
CouponData.MaximumAmount Property 1246
CouponData.MaximumUses Property 1246
CouponData.MinimumAmount Property 1247
CouponData.OnePerCustomer Property 1247
CouponData.StartDate Property 1247
CouponEntryData Class 1247
CouponEntryData Constructor 1247
CouponEntryData.CouponEntryData Constructor () 1248
CouponEntryData.CouponEntryData Constructor (long, EkEnumeration.CMSObjectTypes, long, string, bool) 1248
CouponEntryData Properties 1248

CouponEntryData.CouponId Property 1248
CouponEntryData.DataState Property 1249
CouponEntryData.IsIncluded Property 1249
CouponEntryData.ObjectId Property 1249
CouponEntryData.ObjectType Property 1249
CouponEntryData.Path Property 1249
CouponEntryData.Title Property 1250
CurrencyData Class 1250
CurrencyData Properties 1250
CurrencyData.AlphaIsoCode Property 1250
CurrencyData.CultureCode Property 1251
CurrencyData.CurrencySymbol Property 1251
CurrencyData.Enabled Property 1251
CurrencyData.Id Property 1251
CurrencyData.ISOCurrencySymbol Property 1251
CurrencyData.Name Property 1252
CurrencyPricingData Class 1252
CurrencyPricingData Constructor 1252
CurrencyPricingData.CurrencyPricingData Constructor () 1252
CurrencyPricingData.CurrencyPricingData Constructor (long, decimal, EkEnumeration.PricingType) 1253
CurrencyPricingData.CurrencyPricingData Constructor (long, decimal, decimal) 1253
CurrencyPricingData Methods 1253
CurrencyPricingData.Clone Method 1253
CurrencyPricingData.GetSalePrice Method 1254
CurrencyPricingData.Validate Method 1254
CurrencyPricingData Properties 1254
CurrencyPricingData.AlphaIsoCode Property 1254
CurrencyPricingData.CurrencyId Property 1254
CurrencyPricingData.ListPrice Property 1255
CurrencyPricingData.NumericIsoCode Property 1255
CurrencyPricingData.PricingType Property 1255
CurrencyPricingData.SalePrice Property 1255
CurrencyPricingData.TierPrices Property 1255
CustomerData Class 1255
CustomerData Properties 1256
CustomerData.AverageOrderValue Property 1256
CustomerData.BillingAddressId Property 1256
CustomerData.DisplayName Property 1257

CustomerData.EmailAddress Property 1257
CustomerData.FirstName Property 1257
CustomerData.Id Property 1257
CustomerData.IsDeleted Property 1257
CustomerData.IsMembershipUser Property 1257
CustomerData.LastName Property 1258
CustomerData.Password Property 1258
CustomerData.ShippingAddressId Property 1258
CustomerData.TotalOrders Property 1258
CustomerData.TotalOrderValue Property 1258
CustomerData.UserId Property 1258
CustomerData.UserName Property 1259
Dimensions Class 1259
Dimensions Constructor 1259
Dimensions.Dimensions Constructor () 1259
Dimensions.Dimensions Constructor (LinearUnit, float, float, float) 1260
Dimensions Methods 1260
Dimensions.Clone Method 1260
Dimensions.CompareTo Method 1260
Dimensions.ConvertUnits Method 1261
Dimensions Properties 1261
Dimensions.CubicUnits Property 1261
Dimensions.Height Property 1261
Dimensions.Length Property 1261
Dimensions.Units Property 1262
Dimensions.Width Property 1262
EntryAttributeData Class 1262
EntryAttributeData Constructor 1262
EntryAttributeData.EntryAttributeData Constructor () 1263
EntryAttributeData.EntryAttributeData Constructor (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus) 1263
EntryAttributeData.EntryAttributeData Constructor (string, long, object, int, EkEnumeration.ProductTypeAttributeDataType, string, EkEnumeration.ProductTypeAttributeStatus) 1263
EntryAttributeData Properties 1264
EntryAttributeData.AttributeTypeId Property 1264
EntryAttributeData.CurrentValue Property 1264
EntryAttributeData.VersionId Property 1264
EntryData Class 1264

EntryData Constructor 1265
EntryData.EntryData Constructor () 1265
EntryData.EntryData Constructor (EkEnumeration.CatalogEntryType) 1265
EntryData.EntryData Constructor (int, EkEnumeration.CatalogEntryType) 1265
EntryData Properties 1266
EntryData.Attributes Property 1267
EntryData.Comment Property 1267
EntryData.ContentStatus Property 1267
EntryData.CurrencyId Property 1267
EntryData.DateModified Property 1268
EntryData.Dimensions Property 1268
EntryData.DisableInventoryManagement Property 1268
EntryData.EndDate Property 1268
EntryData.EndDateAction Property 1268
EntryData.EntryType Property 1269
EntryData.FolderId Property 1269
EntryData.GoLive Property 1269
EntryData.Html Property 1269
EntryData.Id Property 1269
EntryData.Image Property 1270
EntryData.ImageThumbnail Property 1270
EntryData.IsArchived Property 1270
EntryData.IsBuyable Property 1270
EntryData.IsMarkedForDeletion Property 1271
EntryData.IsPublished Property 1271
EntryData.IsTangible Property 1271
EntryData.LanguageId Property 1271
EntryData.LastEditorFirstName Property 1271
EntryData.LastEditorId Property 1272
EntryData.LastEditorLastName Property 1272
EntryData.ListPrice Property 1272
EntryData.Media Property 1272
EntryData.Metadata Property 1273
EntryData.NumberSold Property 1273
EntryData.Pricing Property 1273
EntryData.ProductType Property 1273
EntryData.QuantityMultiple Property 1273
EntryData.Quicklink Property 1274
EntryData.SalePrice Property 1274

EntryData.Sku Property 1274
EntryData.Status Property 1274
EntryData.StatusLanguage Property 1275
EntryData.Summary Property 1275
EntryData.TaxClassId Property 1275
EntryData.TemplateId Property 1275
EntryData.Title Property 1275
EntryData.Url Property 1276
EntryData.VersionId Property 1276
EntryData.Weight Property 1276
EntryGroupData Class 1276
EntryGroupData Constructor 1277
EntryGroupData.EntryGroupData Constructor () 1277
EntryGroupData.EntryGroupData Constructor (long, string, string, long, EkEnumeration.EntryGroupType) 1277
EntryGroupData Properties 1277
EntryGroupData.Description Property 1278
EntryGroupData.DisplayOrder Property 1278
EntryGroupData.EntryId Property 1278
EntryGroupData.GroupType Property 1278
EntryGroupData.Id Property 1278
EntryGroupData.Name Property 1278
EntryGroupItemData Class 1279
EntryGroupItemData Constructor 1279
EntryGroupItemData.EntryGroupItemData Constructor () 1279
EntryGroupItemData.EntryGroupItemData Constructor (long, long, long) 1279
EntryGroupItemData Properties 1280
EntryGroupItemData.DisplayOrder Property 1280
EntryGroupItemData.EntryId Property 1280
EntryGroupItemData.GroupId Property 1280
EntryGroupItemData.Id Property 1281
ExchangeRateData Class 1281
ExchangeRateData.ExchangeRateData Constructor 1281
ExchangeRateData Methods 1281
ExchangeRateData.ConvertPrice Method 1281
ExchangeRateData.Validate Method 1282
ExchangeRateData Properties 1282
ExchangeRateData.BaseCurrencyId Property 1282
ExchangeRateData.EffectiveDate Property 1282

ExchangeRateData.ExchangeCurrencyId Property 1283
ExchangeRateData.Rate Property 1283
ImageMediaData Class 1283
ImageMediaData Properties 1283
ImageMediaData.Thumbnails Property 1284
InventoryData Class 1284
InventoryData Properties 1284
InventoryData.EntryId Property 1284
InventoryData.ReorderLevel Property 1285
InventoryData.UnitsInStock Property 1285
InventoryData.UnitsOnOrder Property 1285
KitConfigData Class 1285
KitConfigData.KitConfigData Constructor 1285
KitConfigData Methods 1286
KitConfigData.GetGroupIndex Method 1286
KitConfigData Properties 1286
KitConfigData.BasketItemId Property 1286
KitConfigData.Groups Property 1286
KitConfigData.Id Property 1287
KitConfigData.PriceModifier Property 1287
KitData Class 1287
KitData.KitData Constructor 1287
KitData Properties 1288
KitData.OptionGroups Property 1288
MediaData Class 1288
MediaData Properties 1288
MediaData.Alt Property 1289
MediaData.FileName Property 1289
MediaData.FilePath Property 1289
MediaData.Height Property 1289
MediaData.Id Property 1289
MediaData.MimeType Property 1290
MediaData.Sequence Property 1290
MediaData.Title Property 1290
MediaData.Width Property 1290
MediaGalleryData Class 1291
MediaGalleryData Properties 1291
MediaGalleryData.Images Property 1291
MediaGalleryData.ObjectId Property 1291

OptionGroupData Class 1292
OptionGroupData Constructor 1292
OptionGroupData.OptionGroupData Constructor () 1292
OptionGroupData.OptionGroupData Constructor (long, string, string, string, long) 1292
OptionGroupData.OptionGroupData Constructor (string, string) 1293
OptionGroupData Properties 1293
OptionGroupData.Image Property 1293
OptionGroupData.Options Property 1293
OptionGroupItemCollection Class 1293
OptionGroupItemCollection.OptionGroupItemCollection Constructor 1294
OptionGroupItemCollection Methods 1294
OptionGroupItemCollection.AddRange Method 1294
OptionGroupItemCollection.Find Method 1294
OptionGroupItemCollection.FindAll Method 1295
OptionGroupItemCollection.FindIndex Method 1295
OptionGroupItemData Class 1295
OptionGroupItemData Constructor 1295
OptionGroupItemData.OptionGroupItemData Constructor () 1296
OptionGroupItemData.OptionGroupItemData Constructor (long, long, string, decimal) 1296
OptionGroupItemData.OptionGroupItemData Constructor (string, decimal) 1296
OptionGroupItemData Properties 1296
OptionGroupItemData.CurrencyId Property 1297
OptionGroupItemData.ExtraText Property 1297
OptionGroupItemData.Name Property 1297
OptionGroupItemData.PriceModification Property 1297
OptionGroupItemData.PriceModificationType Property 1297
OrderData Class 1298
OrderData Properties 1298
OrderData.BillingAddressId Property 1299
OrderData.ClientIP Property 1299
OrderData.CouponTotal Property 1299
OrderData.CouponUsed Property 1299
OrderData.Currency Property 1299
OrderData.Customer Property 1300
OrderData.CustomerType Property 1300
OrderData.DateCompleted Property 1300
OrderData.DateCreated Property 1300
OrderData.DateRequired Property 1300
OrderData.HandlingTotal Property 1301

OrderData.HttpHost Property 1301
OrderData.Id Property 1301
OrderData.OrderTotal Property 1301
OrderData.Parts Property 1301
OrderData.ShippingTotal Property 1301
OrderData.SpecialInstructions Property 1302
OrderData.StageId Property 1302
OrderData.StageName Property 1302
OrderData.Status Property 1302
OrderData.Subtotal Property 1302
OrderData.TaxTotal Property 1303
OrderData.VisitorId Property 1303
OrderLineData Class 1303
OrderLineData Properties 1303
OrderLineData.Configuration Property 1304
OrderLineData.ItemType Property 1304
OrderLineData.LineItemId Property 1304
OrderLineData.OrderPartId Property 1304
OrderLineData.PriceEach Property 1304
OrderLineData.PriceSubTotal Property 1305
OrderLineData.PriceTotal Property 1305
OrderLineData.ProductId Property 1305
OrderLineData.ProductLanguage Property 1305
OrderLineData.Quantity Property 1305
OrderPartData Class 1306
OrderPartData Properties 1306
OrderPartData.DateShipped Property 1306
OrderPartData.HandlingTotal Property 1307
OrderPartData.Lines Property 1307
OrderPartData.OrderId Property 1307
OrderPartData.PartId Property 1307
OrderPartData.PartTotal Property 1307
OrderPartData.ShippingAddressId Property 1308
OrderPartData.ShippingTotal Property 1308
OrderPartData.Status Property 1308
OrderPartData.Subtotal Property 1308
OrderPartData.SupplierId Property 1308
OrderPartData.TaxTotal Property 1309
OrderPartData.TrackingNumber Property 1309

PackageData Class 1309
PackageData Constructor 1309
PackageData.PackageData Constructor () 1310
PackageData.PackageData Constructor (string, Dimensions, Weight) 1310
PackageData.PackageData Constructor (string, Dimensions, Weight, long) 1310
PackageData Properties 1310
PackageData.Dimensions Property 1311
PackageData.Id Property 1311
PackageData.MaxWeight Property 1311
PackageData.Name Property 1311
PaymentAuthorization Class 1311
PaymentAuthorization Properties 1312
PaymentAuthorization.AuthorizedOn Property 1312
PaymentAuthorization.CapturedOn Property 1312
PaymentAuthorization.CreditedOn Property 1312
PaymentAuthorization.Gateway Property 1312
PaymentAuthorization.Message Property 1313
PaymentAuthorization.RecurringTransactionId Property 1313
PaymentAuthorization.TransactionId Property 1313
PaymentAuthorization.VoidedOn Property 1313
PipelineStageData Class 1313
PipelineStageData Properties 1314
PipelineStageData.Name Property 1314
PipelineStageData.Sequence Property 1314
PipelineStageData.StatusId Property 1314
PostalCodeTaxRateData Class 1314
PostalCodeTaxRateData Constructor 1315
PostalCodeTaxRateData.PostalCodeTaxRateData Constructor () 1315
PostalCodeTaxRateData.PostalCodeTaxRateData Constructor (string, long, long, decimal) 1315
PostalCodeTaxRateData.PostalCodeTaxRateData Constructor (string, long, long, decimal, long) 1316
PostalCodeTaxRateData Properties 1316
PostalCodeTaxRateData.PostalCode Property 1316
PostalCodeTaxRateData.RegionId Property 1316
PricingData Class 1316
PricingData Constructor 1317
PricingData.PricingData Constructor () 1317
PricingData.PricingData Constructor (List<CurrencyPricingData>) 1317
PricingData.PricingData Constructor (long, decimal, decimal) 1317
PricingData Methods 1318

PricingData.GetCurrencyById Method 1318
PricingData.GetCurrencyByIso Method 1318
PricingData.GetCurrencyByIsoNumber Method 1319
PricingData.GetListPrice Method 1319
PricingData.GetListPriceByCurrencyId Method 1319
PricingData.GetListPriceByIsoNumber Method 1319
PricingData.GetSalePrice Method 1319
PricingData.GetSalePriceByCurrencyId Method 1320
PricingData.GetSalePriceByIsoNumber Method 1320
PricingData.Validate Method 1320
PricingData Properties 1320
PricingData.CurrencyPricelist Property 1320
PricingData.IsRecurringPrice Property 1321
PricingData.Recurrence Property 1321
ProductData Class 1321
ProductData Constructor 1321
ProductData.ProductData Constructor () 1321
ProductData.ProductData Constructor (int) 1322
ProductData Properties 1322
ProductData.Variants Property 1322
ProductTypeAttributeData Class 1322
ProductTypeAttributeData Constructor 1323
ProductTypeAttributeData.ProductTypeAttributeData Constructor () 1323
ProductTypeAttributeData.ProductTypeAttributeData Constructor (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus) 1323
ProductTypeAttributeData Properties 1323
ProductTypeAttributeData.ActiveStatus Property 1324
ProductTypeAttributeData.DataType Property 1324
ProductTypeAttributeData.DefaultValue Property 1324
ProductTypeAttributeData.DisplayOrder Property 1324
ProductTypeAttributeData.Id Property 1324
ProductTypeAttributeData.Name Property 1325
ProductTypeData Class 1325
ProductTypeData Properties 1325
ProductTypeData.Attributes Property 1325
ProductTypeData.DefaultThumbnails Property 1326
ProductTypeData.EntryClass Property 1326
RecommendationItemData Class 1326
RecommendationItemData Constructor 1326

RecommendationItemData.RecommendationItemData Constructor () 1327
RecommendationItemData.RecommendationItemData Constructor (long, long, long) 1327
RecommendationItemData.RecommendationItemData Constructor (long, long, long, EkEnumeration.RecommendationType) 1327
RecommendationItemData Properties 1328
RecommendationItemData.EntryLanguage Property 1328
RecommendationItemData.EntryType Property 1328
RecommendationItemData.Html Property 1328
RecommendationItemData.Image Property 1328
RecommendationItemData.ImageThumbnail Property 1329
RecommendationItemData.Pricing Property 1329
RecommendationItemData.Quicklink Property 1329
RecommendationItemData.Summary Property 1329
RecommendationItemData.Title Property 1329
RecommendationItemData.Type Property 1330
RegionData Class 1330
RegionData Constructor 1330
RegionData.RegionData Constructor () 1330
RegionData.RegionData Constructor (string, long, string, bool) 1331
RegionData Methods 1331
RegionData.ToString Method 1331
RegionData.Validate Method 1331
RegionData Properties 1331
RegionData.Code Property 1332
RegionData.CountryId Property 1332
RegionData.Enabled Property 1332
RegionData.Id Property 1332
RegionData.Name Property 1332
RegionTaxRateData Class 1333
RegionTaxRateData Constructor 1333
RegionTaxRateData.RegionTaxRateData Constructor () 1333
RegionTaxRateData.RegionTaxRateData Constructor (long, long, decimal) 1333
RegionTaxRateData.RegionTaxRateData Constructor (long, long, decimal, long) 1334
RegionTaxRateData Properties 1334
RegionTaxRateData.RegionData Property 1334
ShippingBaseData Class 1334
ShippingBaseData.ShippingBaseData Constructor 1334
ShippingBaseData Properties 1335
ShippingBaseData.Id Property 1335

ShippingBaseData.Name Property 1335
ShippingBaseData.ProviderService Property 1335
ShippingOptionData Class 1335
ShippingOptionData.ShippingOptionData Constructor 1336
ShippingOptionData Properties 1336
ShippingOptionData.CurrencyId Property 1336
ShippingOptionData.ShippingFee Property 1336
TaxClassData Class 1336
TaxClassData Constructor 1337
TaxClassData.TaxClassData Constructor () 1337
TaxClassData.TaxClassData Constructor (string) 1337
TaxClassData.TaxClassData Constructor (string, long) 1338
TaxClassData Properties 1338
TaxClassData.Id Property 1338
TaxClassData.Name Property 1338
TaxRateData Class 1338
TaxRateData Constructor 1339
TaxRateData.TaxRateData Constructor () 1339
TaxRateData.TaxRateData Constructor (long, long, long, decimal) 1339
TaxRateData.TaxRateData Constructor (long, long, long, decimal, long) 1339
TaxRateData Properties 1340
TaxRateData.Id Property 1340
TaxRateData.Rate Property 1340
TaxRateData.TaxClassId Property 1340
TaxRateData.TypeId Property 1340
TaxRateData.TypeItemId Property 1341
TaxTypeData Class 1341
TaxTypeData Constructor 1341
TaxTypeData.TaxTypeData Constructor () 1341
TaxTypeData.TaxTypeData Constructor (string, int) 1342
TaxTypeData.TaxTypeData Constructor (string, int, long) 1342
TaxTypeData Properties 1342
TaxTypeData.Id Property 1342
TaxTypeData.Name Property 1343
TaxTypeData.Precedence Property 1343
ThumbnailData Class 1343
ThumbnailData Properties 1343
ThumbnailData.Height Property 1344
ThumbnailData.Width Property 1344

ThumbnailDefaultData Class 1344
ThumbnailDefaultData Properties 1344
ThumbnailDefaultData.FileNameFormat Property 1345
ThumbnailDefaultData.Height Property 1345
ThumbnailDefaultData.Id Property 1345
ThumbnailDefaultData.MimeType Property 1345
ThumbnailDefaultData.Title Property 1345
ThumbnailDefaultData.Width Property 1346
TierPriceData Class 1346
TierPriceData Constructor 1346
TierPriceData.TierPriceData Constructor () 1347
TierPriceData.TierPriceData Constructor (int, decimal) 1347
TierPriceData.TierPriceData Constructor (int, decimal, long) 1347
TierPriceData Methods 1347
TierPriceData.Clone Method 1347
TierPriceData.CompareTo Method 1348
TierPriceData.Validate Method 1348
TierPriceData Properties 1348
TierPriceData.DataState Property 1348
TierPriceData.Id Property 1348
TierPriceData.Quantity Property 1349
TierPriceData.RecurrenceId Property 1349
TierPriceData.SalePrice Property 1349
ProductCouponEntryData Class 1349
ProductCouponEntryData Constructor 1350
ProductCouponEntryData.ProductCouponEntryData Constructor () 1350
ProductCouponEntryData.ProductCouponEntryData Constructor (long, EkEnumeration.CMSObjectTypes, long, string, bool, EkEnumeration.CatalogEntryType) 1350
ProductCouponEntryData Properties 1350
ProductCouponEntryData.EntryType Property 1351
WarehouseData Class 1351
WarehouseData Constructor 1351
WarehouseData.WarehouseData Constructor () 1351
WarehouseData.WarehouseData Constructor (string, AddressData) 1352
WarehouseData Properties 1352
WarehouseData.Address Property 1352
WarehouseData.Id Property 1352
WarehouseData.IsDefaultWarehouse Property 1352
WarehouseData.Name Property 1353

Weight Class 1353
Weight Constructor 1353
Weight.Weight Constructor () 1353
Weight.Weight Constructor (WeightUnit, float) 1354
Weight Methods 1354
Weight.Clone Method 1354
Weight.CompareTo Method 1354
Weight.ConvertUnits Method 1354
Weight Properties 1355
Weight.Amount Property 1355
Weight.Units Property 1355
ShippingMethodData Class 1355
ShippingMethodData.ShippingMethodData Constructor 1356
ShippingMethodData Properties 1356
ShippingMethodData.DisplayOrder Property 1356
ShippingMethodData.IsActive Property 1356
Structs, Records, Enums 1356
Ektron.Cms.Commerce.AddressProperty Enumeration 1357
Ektron.Cms.Commerce.CountryProperty Enumeration 1357
Ektron.Cms.Commerce.CouponProperty Enumeration 1358
Ektron.Cms.Commerce.CurrencyProperty Enumeration 1358
Ektron.Cms.Commerce.CustomerProperty Enumeration 1358
Ektron.Cms.Commerce.EntryAttributeProperty Enumeration 1359
Ektron.Cms.Commerce.EntryGroupItemProperty Enumeration 1359
Ektron.Cms.Commerce.EntryGroupProperty Enumeration 1359
Ektron.Cms.Commerce.EntryProperty Enumeration 1360
Ektron.Cms.Commerce.ExchangeRateProperty Enumeration 1360
Ektron.Cms.Commerce.LinearUnit Enumeration 1360
Ektron.Cms.Commerce.OrderProperty Enumeration 1360
Ektron.Cms.Commerce.PackageProperty Enumeration 1361
Ektron.Cms.Commerce.ProductTypeAttributeProperty Enumeration 1361
Ektron.Cms.Commerce.ProductTypeProperty Enumeration 1361
Ektron.Cms.Commerce.RegionProperty Enumeration 1362
Ektron.Cms.Commerce.ShippingMethodProperty Enumeration 1362
Ektron.Cms.Commerce.TaxClassProperty Enumeration 1362
Ektron.Cms.Commerce.TaxRateProperty Enumeration 1363
Ektron.Cms.Commerce.TaxRateType Enumeration 1363
Ektron.Cms.Commerce.TaxTypeProperty Enumeration 1363
Ektron.Cms.Commerce.WarehouseProperty Enumeration 1364

Ektron.Cms.Commerce.WeightUnit Enumeration 1364

**Ektron.Cms.Common Namespace 1364**

Classes 1364

UrlAliasAutoData Class 1365

UrlAliasAutoData Properties 1365

UrlAliasAutoData.AliasName Property 1366

UrlAliasAutoData.AutoAliasType Property 1366

UrlAliasAutoData.AutoId Property 1366

UrlAliasAutoData.Example Property 1366

UrlAliasAutoData.ExcludedPath Property 1367

UrlAliasAutoData.FileExtension Property 1367

UrlAliasAutoData.FormattedSelectedPath Property 1367

UrlAliasAutoData.HostName Property 1367

UrlAliasAutoData.IsEnabled Property 1367

UrlAliasAutoData.LanguageId Property 1368

UrlAliasAutoData.PageNameType Property 1368

UrlAliasAutoData.ReplacementCharacter Property 1368

UrlAliasAutoData.SelectedPath Property 1368

UrlAliasAutoData.SiteId Property 1369

UrlAliasAutoData.SourceId Property 1369

UrlAliasAutoData.SourceName Property 1369

UrlAliasAutoData.Target Property 1369

UrlAliasManualData Class 1370

UrlAliasManualData Properties 1370

UrlAliasManualData.AliasId Property 1370

UrlAliasManualData.AliasName Property 1371

UrlAliasManualData.AliasPageName Property 1371

UrlAliasManualData.ContentId Property 1371

UrlAliasManualData.ContentLanguage Property 1371

UrlAliasManualData.ContentTitle Property 1372

UrlAliasManualData.DisplayAlias Property 1372

UrlAliasManualData.FileExtension Property 1372

UrlAliasManualData.HostName Property 1372

UrlAliasManualData.IsDefault Property 1373

UrlAliasManualData.IsEnabled Property 1373

UrlAliasManualData.LibraryId Property 1373

UrlAliasManualData.QueryString Property 1373

UrlAliasManualData.QueryStringAction Property 1373

UrlAliasManualData.SiteId Property 1374

UrlAliasManualData.Target Property 1374
UrlAliasRegExData Class 1374
UrlAliasRegExData Properties 1375
UrlAliasRegExData.Expression Property 1375
UrlAliasRegExData.ExpressionMap Property 1375
UrlAliasRegExData.ExpressionName Property 1375
UrlAliasRegExData.IsEnabled Property 1376
UrlAliasRegExData.Priority Property 1376
UrlAliasRegExData.RegExId Property 1376
UrlAliasRegExData.SiteId Property 1377
UrlAliasRegExData.TransformedUrl Property 1377

**CommunityUserData Class 1377**
CommunityUserData Properties 1377
CommunityUserData.Tags Property 1378

**ContentData Class 1378**
ContentData Properties 1378
ContentData.ApprovalMethod Property 1380
ContentData.Approver Property 1381
ContentData.Comment Property 1381
ContentData.ContType Property 1381
ContentData.DateCreated Property 1382
ContentData.DisplayDateCreated Property 1382
ContentData.DisplayEndDate Property 1382
ContentData.DisplayGoLive Property 1383
ContentData.DisplayLastEditDate Property 1383
ContentData.EditorFirstName Property 1383
ContentData.EditorLastName Property 1383
ContentData.EditorUserNames Property 1384
ContentData.EndDate Property 1384
ContentData.EndDateAction Property 1384
ContentData.FolderId Property 1385
ContentData.FolderName Property 1385
ContentData.GoLive Property 1385
ContentData.HistoryId Property 1385
ContentData.HyperLink Property 1386
ContentData.InheritedFrom Property 1386
ContentData.IsInherited Property 1386
ContentData.IsPublished Property 1386

ContentData.IsSearchable Property 1387
ContentData.IsXmlInherited Property 1387
ContentData.LastEditDate Property 1387
ContentData.ManualAlias Property 1387
ContentData.ManualAliasId Property 1388
ContentData.MetaData Property 1388
ContentData.Path Property 1388
ContentData.Permissions Property 1389
ContentData.Status Property 1389
ContentData.StyleSheet Property 1389
ContentData.TemplateConfiguration Property 1390
ContentData.UserId Property 1390
ContentData.UserName Property 1390
ContentData.XmlConfiguration Property 1390
ContentData.XmlInheritedFrom Property 1391

**ContentRatingLabels Class 1391**
ContentRatingLabels Properties 1391
ContentRatingLabels.AlreadyRatedMessage Property 1392
ContentRatingLabels.BadLabel Property 1392
ContentRatingLabels.CharsRemainingLabel Property 1392
ContentRatingLabels.CookiesRequiredMessage Property 1392
ContentRatingLabels.FeedbackHeader Property 1393
ContentRatingLabels.GoodLabel Property 1393
ContentRatingLabels.RatingLevelLabel Property 1393
ContentRatingLabels.ThankYouMessage Property 1393
ContentRatingLabels.TotalRatingsLabel Property 1393
ContentRatingLabels.UserCommentsHeader Property 1394
ContentRatingLabels.ValidationMessage Property 1394

**FormData Class 1394**
FormData Properties 1395
FormData.Autofill Property 1395
FormData.Description Property 1396
FormData.MailCc Property 1396
FormData.MailFrom Property 1396
FormData.MailPreamble Property 1396
FormData.MailSubject Property 1397
FormData.MailTo Property 1397
FormData.PollRevision Property 1397

FormData.SendXmlPacket Property 1397
FormData.StoreDataTo Property 1398
FormData.Validation Property 1398

**FormFieldDataItem Class 1399**
FormFieldDataItem Properties 1399
FormFieldDataItem.DataValue Property 1399
FormFieldDataItem.FieldName Property 1399

**FormFieldDatalist Class 1400**
FormFieldDatalist Properties 1400
FormFieldDatalist.CaptionXPath Property 1400
FormFieldDatalist.DatalistItems Property 1401
FormFieldDatalist.DataNamespaces Property 1401
FormFieldDatalist.DataSelect Property 1401
FormFieldDatalist.DataSrc Property 1401
FormFieldDatalist.Language Property 1402
FormFieldDatalist.Name Property 1402
FormFieldDatalist.ValueXPath Property 1402

**FormFieldDatalistItem Class 1402**
FormFieldDatalistItem Properties 1403
FormFieldDatalistItem.DataValue Property 1403
FormFieldDatalistItem.DisplayValue Property 1403
FormFieldDatalistItem.IsDefault Property 1404
FormFieldDatalistItem.IsEnabled Property 1404

**FormFieldDefinition Class 1404**
FormFieldDefinition Properties 1405
FormFieldDefinition.Basetype Property 1405
FormFieldDefinition.ContentType Property 1405
FormFieldDefinition.DatalistName Property 1406
FormFieldDefinition.Datatype Property 1406
FormFieldDefinition.DisplayName Property 1406
FormFieldDefinition.FieldName Property 1407
FormFieldDefinition.Language Property 1407
FormFieldDefinition.XPath Property 1407

**FormFieldList Class 1407**
FormFieldList Properties 1408
FormFieldList.Datalists Property 1408
FormFieldList.Fields Property 1408

**FormRequestData Class** **1409**
FormRequestData Properties 1409
FormRequestData.AutoFill Property 1410
FormRequestData.FieldList Property 1410
FormRequestData.Id Property 1410
FormRequestData.IsPoll Property 1410
FormRequestData.Language Property 1411
FormRequestData.PostBackData Property 1411
FormRequestData.PreFillFielddata Property 1411
FormRequestData.TagRequired Property 1412

**FormSubmittedData Class** **1412**
FormSubmittedData Properties 1412
FormSubmittedData.DataItems Property 1413
FormSubmittedData.DateSubmitted Property 1413
FormSubmittedData.FormDataID Property 1413
FormSubmittedData.UserId Property 1413
FormSubmittedData.UserInfo Property 1414

**TaxonomyRequest Class** **1414**
TaxonomyRequest Properties 1414
TaxonomyRequest.Depth Property 1415
TaxonomyRequest.FolderId Property 1415
TaxonomyRequest.IncludeItems Property 1415
TaxonomyRequest.Preview Property 1416
TaxonomyRequest.ReadCount Property 1416
TaxonomyRequest.TaxonomyIdList Property 1416
TaxonomyRequest.TaxonomyItemType Property 1417
TaxonomyRequest.TaxonomyPath Property 1417

**UriFavorite Class** **1417**
UriFavorite Properties 1418
UriFavorite.FavoriteDesc Property 1418
UriFavorite.FavoriteId Property 1418
UriFavorite.FavoriteName Property 1418
UriFavorite.FavoriteUri Property 1419
UriFavorite.UserId Property 1419

**UserData Class** **1419**
UserData Properties 1420
UserData.AcceptedTerms Property 1420

UserData.AdUserName Property 1421
UserData.DateCreated Property 1421
UserData.GroupId Property 1421
UserData.GroupName Property 1421
UserData.IsMemberShip Property 1422
UserData.LanguageId Property 1422
UserData.LanguageName Property 1422
UserData.LastLoginDate Property 1423
UserData.LoginAttempts Property 1423
UserData.Rank Property 1423
UserData.Signature Property 1423
UserData.UserPreference Property 1424

**UserRequestData Class** **1424**
UserRequestData Properties 1424
UserRequestData.Group Property 1425
UserRequestData.OrderBy Property 1425
UserRequestData.OrderDirection Property 1426
UserRequestData.RequiredFlag Property 1426
UserRequestData.SearchText Property 1426
UserRequestData.Type Property 1427

# Index **a**

# Welcome

Welcome to Ektron CMS400.NET API Online Help. This documentation helps you understand the Server Control API (see page 7), Plug-in Extension API (see page 524), .NET Assemblies (see page 572), Web Services API (see page 955) and the Data Classes (see page 1201) associated with Ektron CMS400.NET. Understanding these APIs allows you to enhance and extend your Ektron CMS400.NET site.

- The Server Control API (see page 7) section explains the methods and properties available in code behind for Ektron's CMS400.NET Server Controls. Use these methods and properties when you want to customize the use of a server control beyond the drag and drop functionality provided.
- The Plug-in Extension API (see page 524) documentation details the methods and properties that are available when creating a custom plug-in. Plug-in Extensions expose event hooks in Ektron CMS400.NET. As a developer, you can utilize these event hooks in Visual Studio C# or VB.NET to create customized events for your site.
- The .NET Assemblies (see page 572) are used to access functionality typically provided in the Workarea. For example, adding content, deleting a user or retrieving information about a specific folder. By using the .NET Assemblies you can create custom functionality while keeping within the Ektron Inc. Business Rules.
- The Web Services API (see page 955) exposes many of the same methods as the .NET Assemblies for use with SOAP over HTTP.
- The Data Classes (see page 1201) section describes the properties that allow you to get or set information for each item in Ektron CMS400.NET. For example, the ContentData data class allows you to get or set information about content, such as the date created, content type and the folder that contains the content.

Copyright (c) Ektron Inc. 2008

# How Do I...

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

How Do I...

**Blogs**

create a Web page from which a user can add a blog post? (see page 794)

create a Web page from which a user can retrieve blog information by specifying a blog ID, a date and how to handle post information? (see page 797)

**Content**

create a Web page that allows users to add content? (see page 809)

create a Web page that allows users to delete content by entering the content's ID? (see page 814)

create a Web page from which a user can view a content block and its title by entering the content ID into a text box? (see page 821)

**ContentRating**

create a Web page that retrieves a content rating element used to rate content? (see page 832)

**Folder**

create a Web page from which a folder can be added to the CMS400.NET? (see page 653)

create a Web page from which a folder can be deleted from CMS400.NET by entering its ID? (see page 655)

create a Web page from which a folder ID can be entered and list of child folders returned? (see page 658)

**Library**

create a Web page that displays a graphical list of all images stored in the library for a specific folder in the Workarea? (see page 666)

**Metadata**

create a Web page from which you can add a new metadata definition? (see page 684)

create a Web page from which all metadata for a specific content ID can be retrieved? (see page 686)

Copyright (c) Ektron Inc. 2008

**SiteMap**

create a Web page from which the breadcrumb path for a given folder can be retrieved? (see page 701)

create a Web page that displays the site map for a given folder? (see page 703)

**Threaded Discussions**

create a Web page from which a forum can be added to a discussion board? (see page 850)

create a Web page from which forum data can be retrieved? (see page 859)

**Users and Membership Users**

create a Web page from which a new membership user can be added? (see page 923)

create a Web page that retrieves user details for a given username? (see page 935)

create a Web page that programmatically logs a user into CMS400.NET? (see page 948)

# List of Developer Resources

### List of Developer Resources

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

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

#### Ektron's DevCenter

http://dev.ektron.com

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

#### User Discussion Boards

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

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

#### Ektron Blogs

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

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

#### Ektron Fileshare

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

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

#### Tutorials

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

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

**External Resources**

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

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

**Knowledge Base Articles**

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

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

**Ektron CMS400.NET Developer Manual**

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

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

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

**Ektron's Developer Training Course**

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

Why should you become a Certified Ektron Developer?

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

**Developer Site Provided with Ektron CMS400.NET Starter Site Install**

http://localhost/CMS400Developer/default.aspx

Replace localhost with location of your site.

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

**Ektron Support and Product Pages**

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

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

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

**Supporting Documents**

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

Ektron CMS400.NET Administrator Manual

Ektron CMS400.NET User Manual

Ektron CMS400.NET Quick Start Manual

Ektron CMS400.NET Setup Manual

# Ektron Namespace

# Server Controls API

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

Ektron CMS400.NET's server controls let you insert many standard methods and properties within 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 7) | The CMS namespace contains the Server Control classes. |

# Ektron.Cms Namespace

The CMS namespace contains the Server Control classes.

**Namespaces**

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

# Ektron.Cms.Controls Namespace

This is namespace Ektron.Cms.Controls.

**Classes**

| | Name | Description |
|---|---|---|
| | ActiveTopics (see page 14) | The ActiveTopics server control displays either the most active topics or the most recent topics. The most active are determined by how many new posts are added to a topic or how many replies a post receives. The most recent are determined by the date of the post. |
| | AnalyticsTracker (see page 18) | Use the AnalyticsTracker server control to track statistics about visits to your Web site. |
| | AssetControl (see page 20) | The AssetControl server control, when viewed on a Web form, displays a drag and drop box for users to upload assets to CMS400.NET or update an existing asset. |
| | Blog (see page 24) | The Blog server control allows you to quickly add a blog to a Web form. It has all of the items commonly found on a blog page in one server control. |
| | BlogArchive (see page 32) | The BlogArchive server control displays a list of past months or years that have posts. |

| | | |
|---|---|---|
| | BlogCalendar (see page 35) | The BlogCalendar server control allows a user to display a calendar on a Web page and associate it with a blog. |
| | BlogCategories (see page 37) | The BlogCategories server control displays a blog's categories on a Web form. |
| | BlogEntries (see page 39) | The BlogEntries server control allows you to display a list of multiple blog posts on a Web form. |
| | BlogPost (see page 47) | The BlogPost server control allows you to display an individual blog post on a page. |
| | BlogRecentPosts (see page 49) | The BlogRecentPosts server control displays a list of recent blog posts links on a Web form. |
| | BlogRoll (see page 52) | The BlogRoll server control displays the blog roll for a blog on a Web form. |
| | BlogRSS (see page 54) | The BlogRSS server control displays the icon for the blog's RSS feed on the Web form. |
| | BreadCrumb (see page 56) | The BreadCrumb server control displays history breadcrumbs. Your site visitors can use the breadcrumb trail to jump to previously visited pages. |
| | BreadCrumbItem (see page 63) | Class to contain compass item data. |
| | BusinessRules (see page 66) | Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example:<br>• **Condition** - a site visitor visits your site for the first time<br>• **Action** - add a cookie to the site visitor's system and pop up a welcome message<br>With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 66) |
| | Calendar (see page 69) | Summary description for Calendar. |
| | Cart (see page 73) | This is class Ektron.Cms.Controls.Cart. |
| | Checkout (see page 78) | This is class Ektron.Cms.Controls.Checkout. |
| | Collection (see page 88) | Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces. |
| | CommunityDocuments (see page 98) | This is class Ektron.Cms.Controls.CommunityDocuments. |
| | CommunityGroupBrowser (see page 104) | Summary description for Message board |
| | CommunityGroupList (see page 109) | Summary description for Message board |
| | CommunityGroupMembers (see page 116) | Summary description for Message board |
| | CommunityGroupProfile (see page 122) | |
| | CommunitySearch (see page 127) | This is class Ektron.Cms.Controls.CommunitySearch. |
| | ContentBlock (see page 140) | Summary description for ContentBlock. |
| | ContentFlagging (see page 147) | |
| | ContentList (see page 151) | The ContentList server control is used to display a list of content blocks on a Web page. In contrast to a List Summary, where content must be in a specified folder, the ContentList server control uses content from any CMS400.NET folder. |
| | ContentRating (see page 161) | Use the ContentRating server control to place a rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. A visitor can rank the content on a numerical scale of one through nine, as well as submit a text comment. |
| | ContentReview (see page 169) | The ContentReview server control has two ways it can be used. The first, places a star based rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. An item is rated when a visitor clicks on a star. Depending on the XSLT chosen, visitors can also submit review comments.<br>The second way the ContentReview control can be used is to retrieve and display visitor reviews and comments. In this mode, the control can display reviews of a content item or reviews by... more (see page 169) |
| | CurrencySelect (see page 175) | Summary description for CurrencySelect control |
| | DefaultCatalogIDUITypeEditor (see page 178) | Summary description for DefaultCatalogIDUITypeEditor. |

| | | |
|---|---|---|
| | DesignTimeDiagnostic (see page 178) | The DesignTimeDiagnostic server control is used to verify the connection to Ektron's server controls Web service. When added to a Web form, this control provides the following information:<br>• **Web Service Path** - displays the server control Web service path in your web.config file<br>• **Web Service Exists** - calls a method in the Web service that returns True when the Web service exists. When it does not, it returns False.<br>• **Test Login (see page 337) Control** - displays the Login (see page 337) server control to show that the Web service is connected and working properly. When not connecting properly, an error message is displayed. |
| | DhtmlMenu (see page 181) | The DhtmlMenu server control lets you display a menu with drop down boxes and submenus. This lets developers add a menu to a page without taking up a lot space. |
| | Directory (see page 189) | The Directory server control lets you customize the behavior of the Taxonomy (see page 842) feature. You place this control on any Web form to display a taxonomy. |
| | EktronCatalogPart (see page 202) | When placed inside a Microsoft CatalogZone, the EktronCatalogPart allows users to choose the type of CMS400.NET control that is added to a WebPartZone. |
| | EktronCommunityCatalogPart (see page 209) | The EktronCommunityCatalogPart Server Control gets a list of Discussion Board (Forums) and Blogs from your site. Then, it allows site users to choose from any available Discussion Boards and Blogs and add them to a WebPartZone. |
| | EktronWebPartZone (see page 211) | Similar to a Microsoft WebPartZone, the EktronWebPartZone server control provides a location for content to be displayed on a Web page. The difference between the two is formatting. The EktronWebPartZone is designed to work specifically with Ektron Server Controls, while Microsoft WebPartZones are more generic. |
| | EkWebControl (see page 212) | The base class from which all Ektron CMS400 Server Controls are derived. |
| | EkWebPart (see page 222) | This is class Ektron.Cms.Controls.EkWebPart. |
| | EkXsltBase (see page 224) | Summary description for EkXsltBase. |
| | ExplorerDragDrop (see page 226) | Defines the Explorer Drag Drop control, used to upload assets via Desktop drag drop to the CMS. |
| | Favorites (see page 229) | Summary description for Favorites |
| | FlexMenu (see page 235) | The FlexMenu server control displays a menu on a Web form. Although it looks and acts like a Smart Menu (see page 358), its architecture is substantially different. Specifically, a FlexMenu creates XML. So, you can modify its behavior using an XSLT file, and change its appearance using a cascading style sheet (.css) file. |
| | FolderBreadcrumb (see page 241) | The FolderBreadcrumb server control displays sitemap breadcrumbs. This type of trail is typically made up of the folder path for the content. It is important to note that the FolderBreadcrumb server control does not simply read your folder structure and display its path. It reads breadcrumb information set in the Workarea's folder properties for each folder. Administrators define a folder's breadcrumbs on the Breadcrumb tab located in the Workarea folder properties. *See Also:* The Administrator Manual > "Managing Content (see page 788)" > "Managing Content (see page 788) Folders" > "Breadcrumbs"<br>For additional information on this server control, see the Developer Manual section "Introduction to... more (see page 241) |
| | FormBlock (see page 246) | The FormBlock server control displays a content block associated with a form. |
| | FormFieldItem (see page 253) | Defines a field within the form. |
| | Forum (see page 256) | The Forum server control displays a discussion board on a Web page. |
| | Friends (see page 270) | Summary description for Friends (Colleague Server Control API (see page 639)) |
| | HtmlEditDisplay (see page 277) | This is class Ektron.Cms.Controls.HtmlEditDisplay. |
| | HtmlEditor (see page 280) | This server control provides the ability to edit specific CMS content. The eWebEditPro editor can easily be placed onto a page for content editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor.<br>The control, at run time, produces the fields listed below. They are posted with the content and can be processed in the receiving page.<br>• [ControlName]_Action = the action to take - save, check in, undo checkout<br>• [ControlName]_ContentTitle = the content title<br>• [ControlName]_ContentId = the ID of the content<br>• [ControlName]_Language = the language of the content<br>The content title caption is... more (see page 280) |
| | ImageControl (see page 295) | The ImageControl server control, when viewed on a Web form, displays an image stored within the CMS400.NET document management feature. When a user is logged in and has permission to edit the image, he can right click the image and select edit. This creates a drag and drop box that the user can use to update the file. |

| | | |
|---|---|---|
| | IndexSearch (see page 298) | Use the IndexSearch server control to display a Smart Form (see page 836)'s search screen on any Web form. This lets a user search your site based on XML parameters. |
| | Invite (see page 316) | |
| | LanguageAPI (see page 319) | Control that can be used to override the default system language assignment behavior. |
| | LanguageSelect (see page 323) | The LanguageSelect server control displays a language selection dropdown list on an Ektron CMS400.NET Web page. The dropdown box lists all languages selected on the Settings > Configuration > Language (see page 219) settings screen. |
| | ListSummary (see page 325) | Summary description for ListSummary. |
| | Login (see page 337) | Summary description for Login. |
| | Map (see page 340) | Map control |
| | Membership (see page 350) | The Membership server control creates a form on a Web page that allows a user to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user properties and any custom user properties that exist. It also provides a way to have new membership users read and accept the terms and conditions for using Discussion Boards (Forums) when needed. |
| | Menu (see page 358) | The Menu server control calls a menu to be displayed on a page. Using the Menu server control, you can manipulate a menu by using the DisplayXslt (see page 360) property. |
| | MessageBoard (see page 361) | Summary description for Message board |
| | Messaging (see page 368) | |
| | MetaData (see page 375) | The Metadata server control lets you add the metadata from content blocks to a Web page. This lets developers add metadata quickly without having to type it in. You can add metadata from a single content block, multiple content blocks, or by dynamically passing a content ID from a URL. |
| | MetaDataList (see page 378) | Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces. |
| | MyAccount (see page 390) | This is class Ektron.Cms.Controls.MyAccount. |
| | OrderList (see page 395) | Summary description for OrderList control |
| | PersonalizationManager (see page 400) | The PersonalizationManager server control allows site users to personalize, add, edit and hide content on a Web page. The level of privileges assigned to a site user in CMS400.NET determines what personalization functionality is available when the control is active. The Personalization manager only appears when a user is logged in to the site. |
| | PhotoGallery (see page 406) | This is class Ektron.Cms.Controls.PhotoGallery. |
| | Poll (see page 406) | The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page. |
| | PostHistory (see page 409) | The PostHistory server control displays a list of posts for a given user. The posts are displayed in order by date and contain the following: Topic - the topic that contains the post Posted - the date the post was created Content (see page 788) - the content with in the post |
| | PostLoginEventArgs (see page 412) | Holds the Post-Login (see page 337) event arguments. |
| | Product (see page 412) | Summary description for Product |
| | ProductList (see page 418) | This is class Ektron.Cms.Controls.ProductList. |
| | ProductSearch (see page 426) | The Web Search (see page 455) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 694)" > "Web Site (see page 694) Search (see page 455)." |
| | Recommendation (see page 442) | |
| | RssAggregator (see page 448) | Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces. |
| | RssItemBase (see page 453) | Container for Title (see page 454), Link (see page 454), Description (see page 454) and DatePosted (see page 454) article information. |
| | Search (see page 455) | Search object returns the control and results. |
| | Sitemap (see page 476) | The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site. By choosing the starting point of the sitemap, the max levels to display and applying a style class, you can customize the sitemap. |
| | SmartMenu (see page 481) | Class extends the Menu (see page 358)-Sever-Control, adding CSS based menus. |
| | SocialBar (see page 487) | Summary description for ListSummary (see page 325). |
| | TagCloud (see page 493) | The TagCloud server control allows you to visually depict a weighted list of tags assigned to users or community groups. The weight of each tag is determined by the amount of times a tag is assigned and represented by font size. The large the font size the more times a tag is assigned. Tags (see page 771) are keywords that are associated with a user or community group in CMS400.NET. |
| | UserProfile (see page 499) | |

| | | |
|---|---|---|
| | WebSearch (see page 505) | The Web Search (see page 455) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 694)" > "Web Site (see page 694) Search (see page 455)." |

# Classes

The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | ActiveTopics (see page 14) | The ActiveTopics server control displays either the most active topics or the most recent topics. The most active are determined by how many new posts are added to a topic or how many replies a post receives. The most recent are determined by the date of the post. |
| | AnalyticsTracker (see page 18) | Use the AnalyticsTracker server control to track statistics about visits to your Web site. |
| | AssetControl (see page 20) | The AssetControl server control, when viewed on a Web form, displays a drag and drop box for users to upload assets to CMS400.NET or update an existing asset. |
| | Blog (see page 24) | The Blog server control allows you to quickly add a blog to a Web form. It has all of the items commonly found on a blog page in one server control. |
| | BlogArchive (see page 32) | The BlogArchive server control displays a list of past months or years that have posts. |
| | BlogCalendar (see page 35) | The BlogCalendar server control allows a user to display a calendar on a Web page and associate it with a blog. |
| | BlogCategories (see page 37) | The BlogCategories server control displays a blog's categories on a Web form. |
| | BlogEntries (see page 39) | The BlogEntries server control allows you to display a list of multiple blog posts on a Web form. |
| | BlogPost (see page 47) | The BlogPost server control allows you to display an individual blog post on a page. |
| | BlogRecentPosts (see page 49) | The BlogRecentPosts server control displays a list of recent blog posts links on a Web form. |
| | BlogRoll (see page 52) | The BlogRoll server control displays the blog roll for a blog on a Web form. |
| | BlogRSS (see page 54) | The BlogRSS server control displays the icon for the blog's RSS feed on the Web form. |
| | BreadCrumb (see page 56) | The BreadCrumb server control displays history breadcrumbs. Your site visitors can use the breadcrumb trail to jump to previously visited pages. |
| | BreadCrumbItem (see page 63) | Class to contain compass item data. |
| | BusinessRules (see page 66) | Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example:<br>• **Condition** - a site visitor visits your site for the first time<br>• **Action** - add a cookie to the site visitor's system and pop up a welcome message<br>With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 66) |
| | Calendar (see page 69) | Summary description for Calendar. |
| | Cart (see page 73) | This is class Ektron.Cms.Controls.Cart. |
| | Checkout (see page 78) | This is class Ektron.Cms.Controls.Checkout. |
| | Collection (see page 88) | Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces. |
| | CommunityDocuments (see page 98) | This is class Ektron.Cms.Controls.CommunityDocuments. |
| | CommunityGroupBrowser (see page 104) | Summary description for Message board |
| | CommunityGroupList (see page 109) | Summary description for Message board |
| | CommunityGroupMembers (see page 116) | Summary description for Message board |
| | CommunityGroupProfile (see page 122) | |
| | CommunitySearch (see page 127) | This is class Ektron.Cms.Controls.CommunitySearch. |

| | | |
|---|---|---|
| | ContentBlock (see page 140) | Summary description for ContentBlock. |
| | ContentFlagging (see page 147) | |
| | ContentList (see page 151) | The ContentList server control is used to display a list of content blocks on a Web page. In contrast to a List Summary, where content must be in a specified folder, the ContentList server control uses content from any CMS400.NET folder. |
| | ContentRating (see page 161) | Use the ContentRating server control to place a rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. A visitor can rank the content on a numerical scale of one through nine, as well as submit a text comment. |
| | ContentReview (see page 169) | The ContentReview server control has two ways it can be used. The first, places a star based rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. An item is rated when a visitor clicks on a star. Depending on the XSLT chosen, visitors can also submit review comments.<br>The second way the ContentReview control can be used is to retrieve and display visitor reviews and comments. In this mode, the control can display reviews of a content item or reviews by... more (see page 169) |
| | CurrencySelect (see page 175) | Summary description for CurrencySelect control |
| | DefaultCatalogIDUITypeEditor (see page 178) | Summary description for DefaultCatalogIDUITypeEditor. |
| | DesignTimeDiagnostic (see page 178) | The DesignTimeDiagnostic server control is used to verify the connection to Ektron's server controls Web service. When added to a Web form, this control provides the following information:<br>• **Web Service Path** - displays the server control Web service path in your web.config file<br>• **Web Service Exists** - calls a method in the Web service that returns True when the Web service exists. When it does not, it returns False.<br>• **Test Login (see page 337) Control** - displays the Login (see page 337) server control to show that the Web service is connected and working properly. When not connecting properly, an error message is displayed. |
| | DhtmlMenu (see page 181) | The DhtmlMenu server control lets you display a menu with drop down boxes and submenus. This lets developers add a menu to a page without taking up a lot space. |
| | Directory (see page 189) | The Directory server control lets you customize the behavior of the Taxonomy (see page 842) feature. You place this control on any Web form to display a taxonomy. |
| | EktronCatalogPart (see page 202) | When placed inside a Microsoft CatalogZone, the EktronCatalogPart allows users to choose the type of CMS400.NET control that is added to a WebPartZone. |
| | EktronCommunityCatalogPart (see page 209) | The EktronCommunityCatalogPart Server Control gets a list of Discussion Board (Forums) and Blogs from your site. Then, it allows site users to choose from any available Discussion Boards and Blogs and add them to a WebPartZone. |
| | EktronWebPartZone (see page 211) | Similar to a Microsoft WebPartZone, the EktronWebPartZone server control provides a location for content to be displayed on a Web page. The difference between the two is formatting. The EktronWebPartZone is designed to work specifically with Ektron Server Controls (see page 7), while Microsoft WebPartZones are more generic. |
| | EkWebControl (see page 212) | The base class from which all Ektron CMS400 Server Controls (see page 7) are derived. |
| | EkWebPart (see page 222) | This is class Ektron.Cms.Controls.EkWebPart. |
| | EkXsltBase (see page 224) | Summary description for EkXsltBase. |
| | ExplorerDragDrop (see page 226) | Defines the Explorer Drag Drop control, used to upload assets via Desktop drag drop to the CMS. |
| | Favorites (see page 229) | Summary description for Favorites |
| | FlexMenu (see page 235) | The FlexMenu server control displays a menu on a Web form. Although it looks and acts like a Smart Menu (see page 358), its architecture is substantially different. Specifically, a FlexMenu creates XML. So, you can modify its behavior using an XSLT file, and change its appearance using a cascading style sheet (.css) file. |
| | FolderBreadcrumb (see page 241) | The FolderBreadcrumb server control displays sitemap breadcrumbs. This type of trail is typically made up of the folder path for the content. It is important to note that the FolderBreadcrumb server control does not simply read your folder structure and display its path. It reads breadcrumb information set in the Workarea's folder properties for each folder. Administrators define a folder's breadcrumbs on the Breadcrumb tab located in the Workarea folder properties. *See Also:* The Administrator Manual > "Managing Content (see page 788)" > "Managing Content (see page 788) Folders" > "Breadcrumbs"<br>For additional information on this server control, see the Developer Manual section "Introduction to... more (see page 241) |
| | FormBlock (see page 246) | The FormBlock server control displays a content block associated with a form. |
| | FormFieldItem (see page 253) | Defines a field within the form. |
| | Forum (see page 256) | The Forum server control displays a discussion board on a Web page. |

| | | |
|---|---|---|
| | Friends (see page 270) | Summary description for Friends (Colleague Server Control API (see page 639)) |
| | HtmlEditDisplay (see page 277) | This is class Ektron.Cms.Controls.HtmlEditDisplay. |
| | HtmlEditor (see page 280) | This server control provides the ability to edit specific CMS content. The eWebEditPro editor can easily be placed onto a page for content editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor.<br>The control, at run time, produces the fields listed below. They are posted with the content and can be processed in the receiving page.<br>• [ControlName]_Action = the action to take - save, check in, undo checkout<br>• [ControlName]_ContentTitle = the content title<br>• [ControlName]_ContentId = the ID of the content<br>• [ControlName]_Language = the language of the content<br>The content title caption is... more (see page 280) |
| | ImageControl (see page 295) | The ImageControl server control, when viewed on a Web form, displays an image stored within the CMS400.NET document management feature. When a user is logged in and has permission to edit the image, he can right click the image and select edit. This creates a drag and drop box that the user can use to update the file. |
| | IndexSearch (see page 298) | Use the IndexSearch server control to display a Smart Form (see page 836)'s search screen on any Web form. This lets a user search your site based on XML parameters. |
| | Invite (see page 316) | |
| | LanguageAPI (see page 319) | Control that can be used to override the default system language assignment behavior. |
| | LanguageSelect (see page 323) | The LanguageSelect server control displays a language selection dropdown list on an Ektron CMS400.NET Web page. The dropdown box lists all languages selected on the Settings > Configuration > Language (see page 219) settings screen. |
| | ListSummary (see page 325) | Summary description for ListSummary. |
| | Login (see page 337) | Summary description for Login. |
| | Map (see page 340) | Map control |
| | Membership (see page 350) | The Membership server control creates a form on a Web page that allows a user to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user properties and any custom user properties that exist. It also provides a way to have new membership users read and accept the terms and conditions for using Discussion Boards (Forums) when needed. |
| | Menu (see page 358) | The Menu server control calls a menu to be displayed on a page. Using the Menu server control, you can manipulate a menu by using the DisplayXslt (see page 360) property. |
| | MessageBoard (see page 361) | Summary description for Message board |
| | Messaging (see page 368) | |
| | MetaData (see page 375) | The Metadata server control lets you add the metadata from content blocks to a Web page. This lets developers add metadata quickly without having to type it in. You can add metadata from a single content block, multiple content blocks, or by dynamically passing a content ID from a URL. |
| | MetaDataList (see page 378) | Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces. |
| | MyAccount (see page 390) | This is class Ektron.Cms.Controls.MyAccount. |
| | OrderList (see page 395) | Summary description for OrderList control |
| | PersonalizationManager (see page 400) | The PersonalizationManager server control allows site users to personalize, add, edit and hide content on a Web page. The level of privileges assigned to a site user in CMS400.NET determines what personalization functionality is available when the control is active. The Personalization manager only appears when a user is logged in to the site. |
| | PhotoGallery (see page 406) | This is class Ektron.Cms.Controls.PhotoGallery. |
| | Poll (see page 406) | The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page. |
| | PostHistory (see page 409) | The PostHistory server control displays a list of posts for a given user. The posts are displayed in order by date and contain the following: Topic - the topic that contains the post Posted - the date the post was created Content (see page 788) - the content with in the post |
| | PostLoginEventArgs (see page 412) | Holds the Post-Login (see page 337) event arguments. |
| | Product (see page 412) | Summary description for Product |
| | ProductList (see page 418) | This is class Ektron.Cms.Controls.ProductList. |
| | ProductSearch (see page 426) | The Web Search (see page 455) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 694)" > "Web Site (see page 694) Search (see page 455)." |

| | | |
|---|---|---|
| | Recommendation (see page 442) | |
| | RssAggregator (see page 448) | Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces. |
| | RssItemBase (see page 453) | Container for Title (see page 454), Link (see page 454), Description (see page 454) and DatePosted (see page 454) article information. |
| | Search (see page 455) | Search object returns the control and results. |
| | Sitemap (see page 476) | The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site. By choosing the starting point of the sitemap, the max levels to display and applying a style class, you can customize the sitemap. |
| | SmartMenu (see page 481) | Class extends the Menu (see page 358)-Sever-Control, adding CSS based menus. |
| | SocialBar (see page 487) | Summary description for ListSummary (see page 325). |
| | TagCloud (see page 493) | The TagCloud server control allows you to visually depict a weighted list of tags assigned to users or community groups. The weight of each tag is determined by the amount of times a tag is assigned and represented by font size. The large the font size the more times a tag is assigned. Tags (see page 771) are keywords that are associated with a user or community group in CMS400.NET. |
| | UserProfile (see page 499) | |
| | WebSearch (see page 505) | The Web Search (see page 455) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 694)" > "Web Site (see page 694) Search (see page 455)." |

## ActiveTopics Class

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

### Class Hierarchy

### C#

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

### Visual Basic

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

### File

ActiveTopics.cs

### ActiveTopics Methods

The methods of the ActiveTopics class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 14) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 15) | 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 15) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### ActiveTopics.Fill Method

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

**C#**

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

**Visual Basic**

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

### ActiveTopics.ToString Method

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

**C#**

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

**Visual Basic**

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

### ActiveTopics.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

| Parameters | Description |
|---|---|
| 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 15) | The ID of the board from which to get entries. |
| | CacheInterval (see page 16) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, set the CacheInterval property to 300 (three hundred). |
| | DisplayMode (see page 16) | Choose between Active or Recent. In Active mode, the server control displays the most active posts. In Recent mode, the server control displays the latest posts. The default is Active. |
| | DynamicForumParameter (see page 17) | Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically. |
| | DynamicGroupParameter (see page 17) | Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically. |
| | DynamicThreadParameter (see page 17) | Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically. |
| | MaxNumber (see page 17) | The maximum number of topics listed. The default is 10. |
| | URLPath (see page 18) | The URL path to the Forum (see page 256) server control's Web page. For example: |

### ActiveTopics.BoardID Property

The ID of the board from which to get entries.

**C#**

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

**Visual Basic**

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

### ActiveTopics.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### ActiveTopics.DisplayMode Property

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

**C#**

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

**Visual Basic**

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

### ActiveTopics.DynamicForumParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically.

**C#**

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

**Visual Basic**

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

### ActiveTopics.DynamicGroupParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically.

**C#**

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

**Visual Basic**

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

### ActiveTopics.DynamicThreadParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically.

**C#**

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

**Visual Basic**

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

### ActiveTopics.MaxNumber Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
```

```
[Description("The maximum number of topics to return.")]
public int MaxNumber;
```

**Visual Basic**

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

### ActiveTopics.URLPath Property

The URL path to the Forum (see page 256) server control's Web page. For example:

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

or

```
/[Site Root]/forum.aspx
```

If your Forum (see page 256) page and your Active Topics page are in the same folder, you can just enter the name of the page. For example:

```
forum.aspx
```

**C#**

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

**Visual Basic**

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

## AnalyticsTracker Class

Use the AnalyticsTracker server control to track statistics about visits to your Web site.

**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

### AnalyticsTracker Methods

The methods of the AnalyticsTracker class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | 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. |

## AnalyticsTracker.Wrap Method

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

**C#**

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

**Visual Basic**

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

### Parameters

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

### Returns

Html string wrapped with a toolbar

## AnalyticsTracker Properties

The properties of the AnalyticsTracker class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DefaultContentID (see page 19) | The Content (see page 788) ID to flag as "visited" in the event that no other ID is specified by QueryString (e.g. "?id=2"). Set to zero (0) if you do not wish to use a Default Content (see page 788) ID. |
| | DynamicParameter (see page 20) | Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically. Set to "" (empty) to always display the default ContentID (static.) |
| | EnableAnalytics (see page 20) | Determines if the Analytics (see page 714) feature is enabled for this page and (if identified) the content item. Choices are:<br>• **ConfigSpecified** - Take the value from the web.config file's enableAnalytics property.<br>• **True** - Enable Analytic tracking<br>• **False** - Disable Analytic tracking |

## AnalyticsTracker.DefaultContentID Property

The Content (see page 788) ID to flag as "visited" in the event that no other ID is specified by QueryString (e.g. "?id=2"). Set to zero (0) if you do not wish to use a Default Content (see page 788) ID.

**C#**

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

**Visual Basic**

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

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

### AnalyticsTracker.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically. Set to "" (empty) to always display the default ContentID (static.)

**C#**

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

**Visual Basic**

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

### AnalyticsTracker.EnableAnalytics Property

Determines if the Analytics (see page 714) feature is enabled for this page and (if identified) the content item. Choices are:

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

**C#**

```
public ConfigOrBool EnableAnalytics;
```

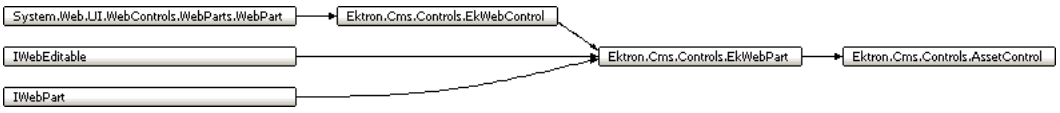
**Visual Basic**

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

## AssetControl Class

The AssetControl server control, when viewed on a Web form, displays a drag and drop box for users to upload assets to CMS400.NET or update an existing asset.

**Class Hierarchy**




**C#**

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

**Visual Basic**

```
Public Class AssetControl
Inherits EkWebPart
```

**File**

DocumentControl.cs

## AssetControl Methods

The methods of the AssetControl class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 21) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 21) | Overrides the system ToString function. |

## AssetControl.Fill Method

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

**C#**

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

**Visual Basic**

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

## AssetControl.ToString Method

Overrides the system ToString function.

**C#**

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

**Visual Basic**

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

**Returns**

String

## AssetControl Properties

The properties of the AssetControl class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DefaultAssetID (see page 22) | The ID of the asset you want to update. This property is used when the UploadType (see page 24) property is set to Update. |
| | DefaultFolderID (see page 22) | The ID of the folder where assets are added. This property is used when the UploadType (see page 24) property is set to Add. Root is 0. |
| | DynamicParameter (see page 22) | Gets or Sets the QueryString parameter to read a Content (see page 788) ID or Folder (see page 652) ID dynamically. Content (see page 788) ID will be read if mode is Update. Folder (see page 652) ID will be read if mode is Add. Set to String.Empty if you wish to always display the default DocumentID or FolderID (static.) |
| | IsImage (see page 23) | Setting this control to 1 (one) restricts the control so only images can be uploaded.<br>• **1** (one) - restrict the control to uploading images only.<br>• **0** (zero) - upload all types of assets. |

| | | |
|---|---|---|
| | OverrideExtension (see page 23) | Allows you to restrict the type of asset that can be uploaded by its extension. For example, to restrict the control to uploading Word documents, enter doc in the property.<br>**Important:** When using this property, enter only the extension's letters not the wildcard (*) or the dot (.).<br>You can add multiple extensions by creating a comma separated list of extensions. Ektron recommends limiting the list to five extensions. |
| | TaxonomyID (see page 24) | The ID of the taxonomy where assets are added. |
| | UploadType (see page 24) | Selects whether the control will add a document to a folder or overwrite an existing document. Select **Add** to add assets and use the DefaultFolderID (see page 22) property. Select **Update** to update assets. |

### AssetControl.DefaultAssetID Property

The ID of the asset you want to update. This property is used when the UploadType (see page 24) property is set to Update.

**C#**

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

**Visual Basic**

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

### AssetControl.DefaultFolderID Property

The ID of the folder where assets are added. This property is used when the UploadType (see page 24) property is set to Add. Root is 0.

**C#**

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

**Visual Basic**

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

### AssetControl.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 788) ID or Folder (see page 652) ID dynamically.

Content (see page 788) ID will be read if mode is Update. Folder (see page 652) ID will be read if mode is Add. Set to String.Empty if you wish to always display the default DocumentID or FolderID (static.)

**C#**

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

**Visual Basic**

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

### AssetControl.IsImage Property

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

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

**C#**

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

**Visual Basic**

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

### AssetControl.OverrideExtension Property

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

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

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

**C#**

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("Allows you to restrict the type of asset that can be uploaded by its extension.
For example, to restrict the control to uploading Word documents, enter doc in the property.
```

```
Important: When using this property, enter only the extension?s letters not the wildcard (*)
or the dot (.).")]
public string OverrideExtension;
```

**Visual Basic**

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

### AssetControl.TaxonomyID Property

The ID of the taxonomy where assets are added.

**C#**

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

**Visual Basic**

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

### AssetControl.UploadType Property

Selects whether the control will add a document to a folder or overwrite an existing document. Select **Add** to add assets and use the DefaultFolderID (see page 22) property. Select **Update** to update assets.

**C#**

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

**Visual Basic**

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

## Blog Class

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

### Class Hierarchy

**C#**

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

**Visual Basic**

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

**File**

Blog.cs

## Blog Methods

The methods of the Blog class are listed here.

### Public Methods

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

## Blog.Fill Method

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

**C#**

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

**Visual Basic**

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

## Blog.ToString Method

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

**C#**

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

**Visual Basic**

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

## Blog.Wrap Method

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

**C#**

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

### Visual Basic

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

### Parameters

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

### Returns

Html string wrapped with a toolbar

## Blog Properties

The properties of the Blog class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | ArchiveMode (see page 26) | Select whether the archive appears in month format or year format. The default is month. |
| | BlogID (see page 27) | The ID of the blog to display when the blog is not associated with a specific user. To display a blog associated with a user, leave this property set to 0 (zero) and enter the user's ID in the DefaultUserID (see page 28) property. |
| | CacheInterval (see page 27) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300. |
| | DateToStart (see page 28) | The date of the last blog posts you want to appear. For example, if you want to display blog posts for January 1, 2007 and before, you enter 1/1/2007. |
| | DefaultUserID (see page 28) | The ID of the user who owns the blog to be displayed. This is used when the server control displays a user's blog. To display a blog not associated with a user, leave this property set to 0 (zero) and enter the blog's ID in the BlogID (see page 27) property. |
| | DynamicParameter (see page 28) | Gets or Sets the QueryString parameter to read a Blog (see page 24) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 24) (static.) |
| | DynamicUserParameter (see page 29) | Gets or sets the QueryString parameter to read a User (see page 913) ID dynamically. Set to String.Empty if you wish to always display the default user's blog (static.) |
| | EditorHeight (see page 29) | Sets the height of the blog editor in pixels. |
| | EditorWidth (see page 29) | Sets the width of the blog editor in pixels. |
| | JavascriptEditorHTMLMode (see page 30) | Set to True to allow users to edit their posts and replies in HTML. |
| | MaxResults (see page 30) | Set the maximum number of posts to display. The default is -3.<br>• If set to 0 (zero), there is no limit.<br>• If set to -1, all posts for the day are shown.<br>• If set to -2, all post for the month are shown.<br>• If set to -3, the control uses the # of Post Visible setting in the Workarea. |
| | PostParameter (see page 30) | Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog. |
| | RecentPosts (see page 31) | Sets the number of recent posts to display. Integer value, must be greater than 0 (Zero). |
| | ShowHeader (see page 31) | Shows the title and tagline when set to True. |
| | ShowRSS (see page 31) | Displays the icon for the RSS feed when set to True. |

## Blog.ArchiveMode Property

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

### C#

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

```
[Localizable(true)]
public EkEnumeration.BlogArchiveMode ArchiveMode;
```

**Visual Basic**

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

### Blog.BlogID Property

The ID of the blog to display when the blog is not associated with a specific user. To display a blog associated with a user, leave this property set to 0 (zero) and enter the user's ID in the DefaultUserID (see page 28) property.

**C#**

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

**Visual Basic**

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

### Blog.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### Blog.DateToStart Property

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

**C#**

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

**Visual Basic**

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

### Blog.DefaultUserID Property

The ID of the user who owns the blog to be displayed. This is used when the server control displays a user's blog. To display a blog not associated with a user, leave this property set to 0 (zero) and enter the blog's ID in the BlogID (see page 27) property.

**C#**

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

**Visual Basic**

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

### Blog.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 24) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 24) (static.)

**C#**

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

**Visual Basic**

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

```
<DefaultValue("blogid")>
Public Property DynamicParameter() As string
```

### Blog.DynamicUserParameter Property

Gets or sets the QueryString parameter to read a User (see page 913) ID dynamically. Set to String.Empty if you wish to always display the default user's blog (static.)

**C#**

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

**Visual Basic**

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

### Blog.EditorHeight Property

Sets the height of the blog editor in pixels.

**C#**

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

**Visual Basic**

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

### Blog.EditorWidth Property

Sets the width of the blog editor in pixels.

**C#**

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

**Visual Basic**

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

### Blog.JavascriptEditorHTMLMode Property

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

**C#**

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

**Visual Basic**

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

### Blog.MaxResults Property

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

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

**C#**

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

**Visual Basic**

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

### Blog.PostParameter Property

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

**C#**

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

**Visual Basic**

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

### Blog.RecentPosts Property

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

**C#**

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

**Visual Basic**

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

### Blog.ShowHeader Property

Shows the title and tagline when set to True.

**C#**

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

**Visual Basic**

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

### Blog.ShowRSS Property

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

**C#**

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

**Visual Basic**

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

```
<Description("Displays the icon for the RSS feed when set to True.")>
Public Property ShowRSS() As bool
```

## BlogArchive Class

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

### Class Hierarchy

### C#

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

### Visual Basic

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

### File

BlogArchive.cs

## BlogArchive Methods

The methods of the BlogArchive class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 32) | Overrideable method that can be used to grab data from the back end for each server control. |
| | Wrap (see page 32) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## BlogArchive.Fill Method

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

### C#

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

### Visual Basic

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

## BlogArchive.Wrap Method

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

### C#

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

### Visual Basic

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

### Parameters

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

### Returns

Html string wrapped with a toolbar

## BlogArchive Properties

The properties of the BlogArchive class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | ArchiveMode (see page 33) | Select whether the archive appears in month format or year format. The default is month. |
| | BlogID (see page 33) | The ID of the blog in CMS400.NET. Can be selected via the CMS Explorer in design time. |
| | CacheInterval (see page 33) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DynamicParameter (see page 34) | Gets or Sets the QueryString parameter to read a Blog (see page 24) ID dynamically. Set to "" if you wish to always display the default blog. |
| | Text (see page 34) | 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 34) | Allows a developer to specify control's tag. Default is assumed "". |

## BlogArchive.ArchiveMode Property

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

**C#**

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

**Visual Basic**

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

## BlogArchive.BlogID Property

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

**C#**

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

**Visual Basic**

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

## BlogArchive.CacheInterval Property

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

**C#**

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

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

**Visual Basic**

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

### BlogArchive.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 24) ID dynamically. Set to "" if you wish to always display the default blog.

**C#**

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

**Visual Basic**

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

### BlogArchive.Text Property

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

**C#**

```
public override string Text;
```

**Visual Basic**

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

### BlogArchive.WrapTag Property

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

**C#**

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

**Visual Basic**

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

# BlogCalendar Class

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

### Class Hierarchy

### C#

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

### Visual Basic

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

### File

BlogCalendar.cs

## BlogCalendar Methods

The methods of the BlogCalendar class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 35) | Overrideable method that can be used to grab data from the back end for each server control. |
| | Wrap (see page 35) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## BlogCalendar.Fill Method

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

### C#

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

### Visual Basic

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

## BlogCalendar.Wrap Method

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

### C#

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

### Visual Basic

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

### Parameters

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

### Returns

Html string wrapped with a toolbar.

## BlogCalendar Properties

The properties of the BlogCalendar class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | BlogID (see page 36) | The ID of the blog for which to display a calendar. Can be selected via the CMS Explorer in design time. |
| | CacheInterval (see page 36) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | Text (see page 36) | 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 37) | Allows a developer to specify control's tag. Default is assumed " ". |

## BlogCalendar.BlogID Property

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

**C#**

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

**Visual Basic**

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

## BlogCalendar.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

## BlogCalendar.Text Property

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

**C#**

```
public override string Text;
```

**Visual Basic**

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

### BlogCalendar.WrapTag Property

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

**C#**

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

**Visual Basic**

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

## BlogCategories Class

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

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

BlogCategories.cs

### BlogCategories Methods

The methods of the BlogCategories class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 37) | 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. |

### 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 |
|---|---|---|
| | BlogID (see page 38) | The ID of the blog from which to display a list of categories. |
| | CacheInterval (see page 39) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | Text (see page 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 " ". |

### BlogCategories.BlogID Property

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

**C#**

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

**Visual Basic**

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

### BlogCategories.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### BlogCategories.Text Property

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

**C#**

```
public override string Text;
```

**Visual Basic**

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

### BlogCategories.WrapTag Property

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

**C#**

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

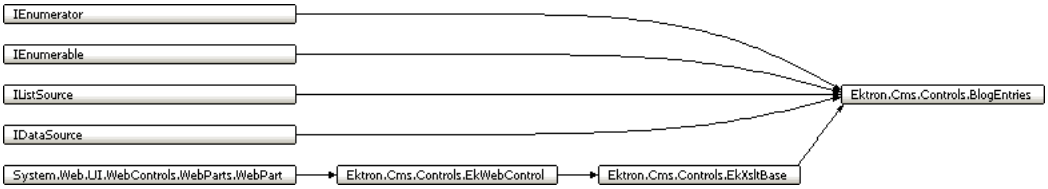
**Visual Basic**

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

## BlogEntries Class

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

### Class Hierarchy




**C#**

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

**Visual Basic**

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

**File**

BlogEntries.cs

## BlogEntries Methods

The methods of the BlogEntries class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 40) | Method that can be overridden and used to grab data from the back end for each server control. |
| | GetRssFeed (see page 40) | This method returns a blog post RSS feed. |
| | GetView (see page 41) | Gets the view data using dataset |
| | GetViewNames (see page 41) | gets the view names |
| | ToString (see page 41) | 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 41) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## BlogEntries.Fill Method

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

**C#**

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

**Visual Basic**

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

## BlogEntries.GetRssFeed Method

This method returns a blog post RSS feed.

**C#**

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

**Visual Basic**

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

**Returns**

String

### BlogEntries.GetView Method

Gets the view data using dataset

**C#**

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

**Visual Basic**

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

**Parameters**

| Parameters | Description |
|---|---|
| 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 |
|---|---|---|
| | BlogID (see page 42) | The ID of the blog to display when the blog is not associated with a specific user. To display a blog associated with a user, leave this property set to 0 (zero) and enter the user's ID in the DefaultUserID (see page 44) property. |
| | BlogPostParameter (see page 43) | Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog. |
| | CacheInterval (see page 43) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300. |
| | DateToStart (see page 43) | The date of the last blog posts you want to appear. For example, if you want to display blog posts for January 1, 2007 and before, you enter 1/1/2007. |
| | DefaultUserID (see page 44) | The ID of the user who owns the blog to be displayed. This is used when the server control displays a user's blog. To display a blog not associated with a user, leave this property set to 0 (zero) and enter the blog's ID in the BlogID (see page 42) property. |
| | DisplayXslt (see page 44) | The Xslt used to render the blog entry list (optional). |
| | DynamicParameter (see page 44) | Gets or Sets the QueryString parameter to read a Blog (see page 24) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 24) (static.) |
| | DynamicUserParameter (see page 45) | Gets or sets the QueryString parameter to read a User (see page 913) ID dynamically. Set to String.Empty if you wish to always display the default user's blog (static.) |
| | EditorHeight (see page 45) | Sets the height of the blog editor in pixels. |
| | EditorWidth (see page 45) | Sets the width of the blog editor in pixels. |
| | JavascriptEditorHTMLMode (see page 46) | Set to True to allow users to edit their posts and replies in HTML. |
| | MaxResults (see page 46) | Set the maximum number of posts to display. The default is -3.<br>• If set to 0 (zero), there is no limit.<br>• If set to -1, all posts for the day are shown.<br>• If set to -2, all post for the month are shown.<br>• If set to -3, the control uses the # of Post Visible setting in the Workarea. |
| | ShowHeader (see page 46) | Shows the title and tagline when set to True. |
| | XmlDoc (see page 47) | Returns the Xslt object for the collection. |

## BlogEntries.BlogID Property

The ID of the blog to display when the blog is not associated with a specific user. To display a blog associated with a user, leave this property set to 0 (zero) and enter the user's ID in the DefaultUserID (see page 44) property.

**C#**

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

**Visual Basic**

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

### BlogEntries.BlogPostParameter Property

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

**C#**

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

**Visual Basic**

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

### BlogEntries.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### BlogEntries.DateToStart Property

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

**C#**

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

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

**Visual Basic**

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

### BlogEntries.DefaultUserID Property

The ID of the user who owns the blog to be displayed. This is used when the server control displays a user's blog. To display a blog not associated with a user, leave this property set to 0 (zero) and enter the blog's ID in the BlogID (see page 42) property.

**C#**

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

**Visual Basic**

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

### BlogEntries.DisplayXslt Property

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

**C#**

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

**Visual Basic**

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

### BlogEntries.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 24) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 24) (static.)

**C#**

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

```
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("blogid")]
public string DynamicParameter;
```

**Visual Basic**

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

### BlogEntries.DynamicUserParameter Property

Gets or sets the QueryString parameter to read a User (see page 913) ID dynamically. Set to String.Empty if you wish to always display the default user's blog (static.)

**C#**

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

**Visual Basic**

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

### BlogEntries.EditorHeight Property

Sets the height of the blog editor in pixels.

**C#**

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

**Visual Basic**

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

### BlogEntries.EditorWidth Property

Sets the width of the blog editor in pixels.

**C#**

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

```
public int EditorWidth;
```

**Visual Basic**

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

### BlogEntries.JavascriptEditorHTMLMode Property

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

**C#**

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

**Visual Basic**

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

### BlogEntries.MaxResults Property

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

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

**C#**

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

**Visual Basic**

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

### BlogEntries.ShowHeader Property

Shows the title and tagline when set to True.

**C#**

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

**Visual Basic**

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

### BlogEntries.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

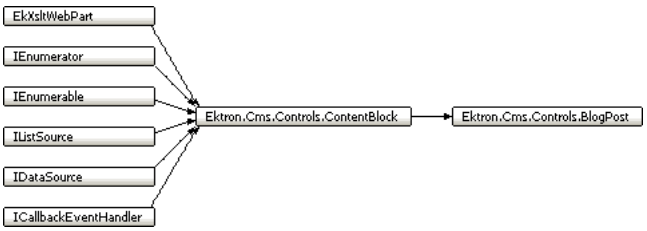
**Visual Basic**

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

## BlogPost Class

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

**Class Hierarchy**




**C#**

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

**Visual Basic**

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

**File**

BlogPost.cs

### BlogPost Methods

The methods of the BlogPost class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Wrap (see page 48) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## BlogPost.Wrap Method

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

**C#**

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

**Visual Basic**

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

### Parameters

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

### Returns

Html string wrapped with a toolbar

## BlogPost Properties

The properties of the BlogPost class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DefaultContentID (see page 48) | The ID of a default blog post that appears where you inserted this server control if no other content block is identified, or is not available. |
| | ShowType (see page 49) | Shows a blog post's content and its comments or just the blog post's comments. The default is Content (see page 788).<br>• **Content (see page 788)** - Show a blog post's content and its comments.<br>• **Description** - Show a blog post's comments only. |

## BlogPost.DefaultContentID Property

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

**C#**

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

**Visual Basic**

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

### BlogPost.ShowType Property

Shows a blog post's content and its comments or just the blog post's comments. The default is Content (see page 788).

- **Content (see page 788)** - Show a blog post's content and its comments.
- **Description** - Show a blog post's comments only.

**C#**

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

**Visual Basic**

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

## BlogRecentPosts Class

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

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

BlogRecentPosts.cs

### BlogRecentPosts Methods

The methods of the BlogRecentPosts class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 50) | Overrideable method that can be used to grab data from the back end for each server control. |
| | Wrap (see page 50) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### BlogRecentPosts.Fill Method

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

**C#**

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

**Visual Basic**

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

### BlogRecentPosts.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

Html string wrapped with a toolbar

### BlogRecentPosts Properties

The properties of the BlogRecentPosts class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | BlogID (see page 50) | The ID of the blog from which to select recent posts. Can be selected via the CMS Explorer in design time. |
| | CacheInterval (see page 51) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | NumberofPosts (see page 51) | Sets the number of recent posts to display. Integer value, must be greater than 0 (zero). |
| | Text (see page 51) | 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 51) | Allows a developer to specify control's tag. Default is assumed " ". |

### BlogRecentPosts.BlogID Property

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

**C#**

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

**Visual Basic**

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

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

### BlogRecentPosts.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### BlogRecentPosts.NumberofPosts Property

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

**C#**

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

**Visual Basic**

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

### BlogRecentPosts.Text Property

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

**C#**

```
public override string Text;
```

**Visual Basic**

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

### BlogRecentPosts.WrapTag Property

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

**C#**

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

**Visual Basic**

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

## BlogRoll Class

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

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

BlogRoll.cs

### BlogRoll Methods

The methods of the BlogRoll class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 52) | Overrideable method that can be used to grab data from the back end for each server control. |
| | Wrap (see page 52) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### BlogRoll.Fill Method

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

**C#**

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

**Visual Basic**

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

### BlogRoll.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

Html string wrapped with a toolbar

## BlogRoll Properties

The properties of the BlogRoll class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | BlogID (see page 53) | The ID of the blog in CMS400.NET. Can be selected via the CMS Explorer in design time. |
| | CacheInterval (see page 53) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | Text (see page 54) | 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 54) | Allows a developer to specify control's tag. Default is assumed " ". |

## BlogRoll.BlogID Property

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

**C#**

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

**Visual Basic**

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

## BlogRoll.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### BlogRoll.Text Property

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

**C#**

```
public override string Text;
```

**Visual Basic**

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

### BlogRoll.WrapTag Property

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

**C#**

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

**Visual Basic**

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

## BlogRSS Class

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

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

BlogRSS.cs

### BlogRSS Methods

The methods of the BlogRSS class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 55) | Overrideable method that can be used to grab data from the back end for each server control. |

## BlogRSS.Fill Method

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

**C#**

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

**Visual Basic**

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

## BlogRSS Properties

The properties of the BlogRSS class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | BlogID (see page 55) | The ID of the blog to be output as an RSS feed. |
| | CacheInterval (see page 55) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | Text (see page 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. |
| | WrapTag (see page 56) | Allows a developer to specify control's tag. Default is assumed " ". |

## BlogRSS.BlogID Property

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

**C#**

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

**Visual Basic**

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

## BlogRSS.CacheInterval Property

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

**C#**

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

```
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

**Visual Basic**

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

### BlogRSS.Text Property

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

**C#**

```
public override string Text;
```

**Visual Basic**

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

### BlogRSS.WrapTag Property

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

**C#**

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

**Visual Basic**

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

## BreadCrumb Class

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

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

BreadCrumb.cs

## BreadCrumb Methods

The methods of the BreadCrumb class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 57) | Method that can be overridden and used to grab data from the back end for each server control. |
| | GenerateTrail_HTML (see page 57) | Generates Trail HTML for both vertical and horizontal styles<br>This demonstrates usage of a HtmlTextWriter. |
| | ResetTrail (see page 57) | This is the overview for the ResetTrail method overload. |

## BreadCrumb.Fill Method

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

**C#**

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

**Visual Basic**

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

## BreadCrumb.GenerateTrail_HTML Method

Generates Trail HTML for both vertical and horizontal styles

This demonstrates usage of a HtmlTextWriter.

**C#**

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

**Visual Basic**

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

**Parameters**

| Parameters | Description |
|---|---|
| output | HtmlTextWriter in which to write. |

## ResetTrail Method

This is the overview for the ResetTrail method overload.

**Description**

This method is overloaded. The ResetTrail method:

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

**Overload List**

| | Name | Description |
|---|---|---|
| | BreadCrumb.ResetTrail () (see page 58) | Resets and removes all current Trail items. |

| | | |
|---|---|---|
| | BreadCrumb.ResetTrail (ArrayList) (see page 58) | Resets the trail with an array list of new trail items. |
| | BreadCrumb.ResetTrail (BreadCrumbItem) (see page 58) | 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 | An array list of trail items |

### BreadCrumb.ResetTrail Method (BreadCrumbItem)

Resets the trail to the given trail item.

**C#**

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

**Visual Basic**

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

**Parameters**

| Parameters | Description |
|---|---|
| item | A trail item. |

### BreadCrumb Properties

The properties of the BreadCrumb class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CurrentPageIndicator (see page 59) | If desired, enter one or more symbols or characters that identify the current page in the breadcrumb trail. |
| | DefaultContentID (see page 59) | Gets or sets the default content ID for this control. |
| | DisplayStyle (see page 60) | Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal. |

| | | |
|---|---|---|
| | DisplayTitle (see page 60) | Enter text to describe this Web form when it appears in the breadcrumb trail. For example, if the Web form whose properties you are defining is used for all Human Resources pages on your site, enter Human Resources. |
| | DynamicParameter (see page 60) | Gets or Sets the QueryString parameter to read a content ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.) |
| | IconAlt (see page 60) | If you define an image in the IconPath (see page 61) property, enter any “Alt” text that should appear when a site visitor hovers the cursor over that image. |
| | IconPath (see page 61) | If you want the breadcrumb trail to display an image to identify this Web form, enter the path to the image.<br>**Important:** The image location must be relative to the Web root. |
| | LinkLastItem (see page 61) | Determine whether the last breadcrumb item appears as a hyperlink on this Web form. |
| | LinkTarget (see page 61) | Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.<br>• **_Self** = opens in same window<br>• **_Top** = opens in parent window<br>• **_Blank** = opens in new window<br>• **_Parent** = opens in the parent frame |
| | MaxItems (see page 61) | The maximum number of items in the breadcrumb trail on this Web form. (0 = Unlimited) |
| | MetadataName (see page 62) | Gets or sets the metadata of the content block for BreadCrumb (see page 56) display title. If not specified it will use content block title. |
| | Mode (see page 62) | Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal.<br>• **Normal** = breadcrumb trail is hyperlinked<br>• **DisplayOnly** = breadcrumb trail is plain text |
| | separator (see page 62) | Enter one or more characters to separate the items in a breadcrumb trail on this Web form. The default character is the greater than sign (>). |

### BreadCrumb.CurrentPageIndicator Property

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

**C#**

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

**Visual Basic**

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

### BreadCrumb.DefaultContentID Property

Gets or sets the default content ID for this control.

**C#**

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

**Visual Basic**

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

### BreadCrumb.DisplayStyle Property

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

**C#**

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

**Visual Basic**

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

### BreadCrumb.DisplayTitle Property

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

**C#**

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

**Visual Basic**

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

### BreadCrumb.DynamicParameter Property

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

**C#**

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

**Visual Basic**

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

### BreadCrumb.IconAlt Property

If you define an image in the IconPath (see page 61) property, enter any “Alt” text that should appear when a site visitor hovers the cursor over that image.

**C#**

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

**Visual Basic**

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

### BreadCrumb.IconPath Property

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

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

**C#**

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

**Visual Basic**

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

### BreadCrumb.LinkLastItem Property

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

**C#**

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

**Visual Basic**

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

### BreadCrumb.LinkTarget Property

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

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

**C#**

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

**Visual Basic**

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

### BreadCrumb.MaxItems Property

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

**C#**

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

**Visual Basic**

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

### BreadCrumb.MetadataName Property

Gets or sets the metadata of the content block for BreadCrumb (see page 56) display title. If not specified it will use content block title.

**C#**

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

**Visual Basic**

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

### BreadCrumb.Mode Property

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

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

**C#**

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

**Visual Basic**

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

### BreadCrumb.separator Property

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

**C#**

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

**Visual Basic**

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

## BreadCrumbItem Class

Class to contain compass item data.

**Class Hierarchy**

**C#**

```
public class BreadCrumbItem;
```

**Visual Basic**

```
Public Class BreadCrumbItem
```

**File**

BreadCrumbItem.cs

### BreadCrumbItem Constructor

This is the overview for the BreadCrumbItem constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | BreadCrumbItem.BreadCrumbItem () (see page 63) | Default no parameter constructor. |
| | BreadCrumbItem.BreadCrumbItem (string, string, string, string, string) (see page 63) | All parameter constructor. |

### BreadCrumbItem.BreadCrumbItem Constructor ()

Default no parameter constructor.

**C#**

```
public BreadCrumbItem();
```

**Visual Basic**

```
Public Sub New()
```

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

All parameter constructor.

**C#**

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

**Visual Basic**

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

### Parameters

| Parameters | Description |
|---|---|
| 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 64) | Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 64)() do not match, The ToString (see page 64)() function is also overridden to allow this to work. |
| | GetHashCode (see page 64) | Overridden otherwise VS.NET gives a warning. |
| | ToString (see page 64) | Returns a concatenated string of all class variables.ToString(). |

## BreadCrumbItem.Equals Method

Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 64)() do not match, The ToString (see page 64)() 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 |
|---|---|---|
| | CaseSensitiveUrlComparison (see page 65) | 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 65) | Gets / Sets the title for use with the display in the compass. |
| | IconAlt (see page 65) | Gets / Sets the text to use as the item alt text. |
| | IconPath (see page 65) | Gets / Sets the root relative path to the item icon. |
| | RawUrl (see page 66) | Gets / Sets the URL to use for the item link. |
| | Target (see page 66) | Gets / Sets the target of the item. |

### BreadCrumbItem.CaseSensitiveUrlComparison Property

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

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

**C#**

```
public static bool CaseSensitiveUrlComparison;
```

**Visual Basic**

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

### BreadCrumbItem.DisplayTitle Property

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

**C#**

```
public string DisplayTitle;
```

**Visual Basic**

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

### BreadCrumbItem.IconAlt Property

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

**C#**

```
public string IconAlt;
```

**Visual Basic**

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

### BreadCrumbItem.IconPath Property

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

**C#**

```
public string IconPath;
```

**Visual Basic**

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

### BreadCrumbItem.RawUrl Property

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

**C#**

```
public string RawUrl;
```

**Visual Basic**

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

### BreadCrumbItem.Target Property

Gets / Sets the target of the item.

**C#**

```
public string Target;
```

**Visual Basic**

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

## BusinessRules Class

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

BusinessRules.cs

**Description**

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

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

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

**.NET Assemblies BusinessRule API vs. BusinessRules Server Control API**

The .NET Assemblies BusinessRule (see page 640) API (see page 639) allows you to work with results of a Business Rule. The BusinessRules server control API (see page 639) allows you to customize how and where a Business Rule is used on a

Web page.

To learn about the .NET Assemblies BusinessRule (see page 640) API (see page 639), click here (see page 640).

## BusinessRules Methods

The methods of the BusinessRules class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 67) | Overrideable method that can be used to grab data from the back end for each server control. |
| | Wrap (see page 67) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## BusinessRules.Fill Method

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

**C#**

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

**Visual Basic**

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

## BusinessRules.Wrap Method

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

**C#**

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

**Visual Basic**

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

### Parameters

| Parameters | Description |
|---|---|
| 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 67) | User (see page 913) Defined parameters that can be evaluated in rules. |
| | Result (see page 68) | An array of boolean values representing the results of the evaluations. |
| | RuleID (see page 68) | The ID number of the Ruleset to evaluate. |
| | Text (see page 68) | 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 69) | Overridden and hidden - should not be wrapped by span |

## BusinessRules.Parameters Property

User (see page 913) Defined parameters that can be evaluated in rules.

**C#**

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

**Visual Basic**

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

### BusinessRules.Result Property

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

**C#**

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

**Visual Basic**

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

### BusinessRules.RuleID Property

The ID number of the Ruleset to evaluate.

**C#**

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

**Visual Basic**

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

### BusinessRules.Text Property

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

**C#**

```
public override string Text;
```

**Visual Basic**

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

### BusinessRules.WrapTag Property

Overridden and hidden - should not be wrapped by span

**C#**

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

**Visual Basic**

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

## Calendar Class

Summary description for Calendar.

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

Calendar.cs

### Calendar Methods

The methods of the Calendar class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 69) | Method that can be overridden and used to grab data from the back end for each server control. |
| | Wrap (see page 70) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### Calendar.Fill Method

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

**C#**

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

**Visual Basic**

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

## Calendar.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

| Parameters | Description |
|---|---|
| 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 70) | The default ID of the Calendar (see page 69) to be displayed. |
| | DisplayType (see page 71) | Specify the the Calendar (see page 69) view. Choices are:<br>• "Month"<br>• "Day"<br>• "Inline"<br>• "MonthUpWithEvent" |
| | DynamicParameter (see page 71) | The QueryString to check for a dynamic ID. |
| | EndDate (see page 72) | Specify the end date the Calendar (see page 69) displays. |
| | HighlightEventID (see page 72) | Specify the event ID which the Calendar (see page 69) will highlight. |
| | HighlightEventTypeID (see page 72) | Specify the event type and the Calendar (see page 69) highlights that type of event(s). |
| | ShowXML (see page 72) | Text (see page 73) property will hold raw calendar XML. The Calendar (see page 69) is not visible at runtime. |
| | StartDate (see page 73) | Specify the start date the calendar displays. |
| | Text (see page 73) | Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display. |

## Calendar.DefaultCalendarID Property

The default ID of the Calendar (see page 69) to be displayed.

**C#**

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

```
public long DefaultCalendarID;
```

**Visual Basic**

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

### Calendar.DisplayType Property

Specify the the Calendar (see page 69) 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 69) 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 69) will highlight.

**C#**

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

**Visual Basic**

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

### Calendar.HighlightEventTypeID Property

Specify the event type and the Calendar (see page 69) highlights that type of event(s).

**C#**

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

**Visual Basic**

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

### Calendar.ShowXML Property

Text (see page 73) property will hold raw calendar XML. The Calendar (see page 69) is not visible at runtime.

**C#**

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

**Visual Basic**

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

### Calendar.StartDate Property

Specify the start date the calendar displays.

**C#**

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

**Visual Basic**

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

### Calendar.Text Property

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

**C#**

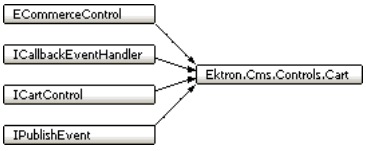
```
public override string Text;
```

**Visual Basic**

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

## Cart Class

**Class Hierarchy**




**C#**

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

**Visual Basic**

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

**File**

Cart.cs

**Description**

This is class Ektron.Cms.Controls.Cart.

## Cart Properties

The properties of the Cart class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DisplayXslt (see page 74) | Specify an external XSLT file to render the control. By default, the control uses Cart.xsl. This file is located in: |
| | DynamicProductParameter (see page 75) | The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors clicks a product's title, this parameter is passed with the product's ID to the product details page. |
| | EnableCoupons (see page 75) | Set to true to allow a site visitor to enter coupon codes for discounts.<br>• **True** = coupon entry is available.<br>• **False** = coupon entry is not available. |
| | LoadingImage (see page 76) | Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**. |
| | Stylesheet (see page 76) | Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | TemplateCheckout (see page 76) | The template location that contains the Checkout (see page 78) server control. This is the page a site visitor is sent to when the he clicks the Checkout (see page 78) button. If the template file is located in the same folder as the Web form containing this server control, just enter its name. Otherwise, the path to the file can be relative or absolute. |
| | TemplateProduct (see page 77) | The template location that contains the Product (see page 412) server control. This is the page a site visitor is sent to when he clicks the name of a product in his cart. This page provides details about the product. When this property is left blank, the server control uses the Product (see page 412)'s QuickLink information. If the template file is located in the same folder as the Web form containing this server control, just enter its name. Otherwise, the path to the file can be relative or absolute. |
| | TemplateShopping (see page 77) | The URL to navigate to when the site visitor clicks the Continue Shopping button. This URL could lead to a template containing a ProductSearch (see page 426) or ProductList (see page 418) server control that would allow a site visitor to select additional products. |
| | UserControlUrl (see page 77) | The URL of the User (see page 913) Control that manages the display of the Cart (see page 73). This path can be relative or absolute. A sample user control is located in: |

## Cart.DisplayXslt Property

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

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

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
```

```
uses \"Cart.xsl.\"")]
[TypeConverter(typeof(XsltCartTypeConverter))]
[DefaultValue("Cart.xsl")]
public override string DisplayXslt;
```

**Visual Basic**

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

### Cart.DynamicProductParameter Property

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

**C#**

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

**Visual Basic**

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

### Cart.EnableCoupons Property

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

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

**C#**

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

**Visual Basic**

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

### Cart.LoadingImage Property

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

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

### Cart.Stylesheet Property

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

**C#**

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(CartCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/Cart.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(CartCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/Cart.css")>
Public Property Stylesheet() As string
```

### Cart.TemplateCheckout Property

The template location that contains the Checkout (see page 78) server control. This is the page a site visitor is sent to when the he clicks the Checkout (see page 78) button. If the template file is located in the same folder as the Web form containing this server control, just enter its name. Otherwise, the path to the file can be relative or absolute.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The template location that contains the Checkout server control. This is the
page a site visitor is sent to when the he clicks the Checkout button. If the template file is
located in the same folder as the Web form containing this server control, just enter its
name. Otherwise, the path to the file can be relative or absolute.")]
[DefaultValue("")]
public string TemplateCheckout;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The template location that contains the Checkout server control. This is the
page a site visitor is sent to when the he clicks the Checkout button. If the template file is
```

```
located in the same folder as the Web form containing this server control, just enter its
name. Otherwise, the path to the file can be relative or absolute.")>
<DefaultValue("")>
Public Property TemplateCheckout() As string
```

### Cart.TemplateProduct Property

The template location that contains the Product (see page 412) server control. This is the page a site visitor is sent to when he clicks the name of a product in his cart. This page provides details about the product. When this property is left blank, the server control uses the Product (see page 412)'s QuickLink information. If the template file is located in the same folder as the Web form containing this server control, just enter its name. Otherwise, the path to the file can be relative or absolute.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The template location that contains the Product server control. This is the page
a site visitor is sent to when he clicks the name of a product in his cart. This page provides
details about the product.")]
[DefaultValue("")]
public string TemplateProduct;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The template location that contains the Product server control. This is the page
a site visitor is sent to when he clicks the name of a product in his cart. This page provides
details about the product.")>
<DefaultValue("")>
Public Property TemplateProduct() As string
```

### Cart.TemplateShopping Property

The URL to navigate to when the site visitor clicks the Continue Shopping button. This URL could lead to a template containing a ProductSearch (see page 426) or ProductList (see page 418) server control that would allow a site visitor to select additional products.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL to navigate to, when the user clicks the 'Continue Shopping' button.")]
[DefaultValue("")]
public string TemplateShopping;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL to navigate to, when the user clicks the 'Continue Shopping' button.")>
<DefaultValue("")>
Public Property TemplateShopping() As string
```

### Cart.UserControlUrl Property

The URL of the User (see page 913) Control that manages the display of the Cart (see page 73). This path can be relative or absolute. A sample user control is located in:

```
[siteroot]\Workarea\ControlTemplates\Commerce\Cart\CartUserControl.ascx.
```

**C#**

```
[Category("Ektron")]
[Description("The URL of the User Control that manages the display of the Cart. This path can
be relative or absolute. A sample user control is located in
<siteroot>\\Workarea\\ControlTemplates\\Commerce\\Cart\\CartUserControl.ascx.")]
[TypeConverter(typeof(string))]
```

```
[DefaultValue("")]
public string UserControlUrl;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("The URL of the User Control that manages the display of the Cart. This path can
be relative or absolute. A sample user control is located in
<siteroot>\\Workarea\\ControlTemplates\\Commerce\\Cart\\CartUserControl.ascx.")>
<TypeConverter(typeof(string))>
<DefaultValue("")>
Public Property UserControlUrl() As string
```

## Checkout Class

**Class Hierarchy**

**C#**

```
[DefaultProperty("DefaultCheckoutID"), ToolboxData("<{0}:Checkout
runat=server></{0}:Checkout>")]
public class Checkout : ECommerceControl, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultCheckoutID"), ToolboxData("<{0}:Checkout
runat=server></{0}:Checkout>")>
Public Class Checkout
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

**File**

Checkout.cs

**Description**

This is class Ektron.Cms.Controls.Checkout.

### Checkout Enumerations

The enumerations of the Checkout class are listed here.

**Enumerations**

| | Name | Description |
|---|---|---|
| | ControlPhase (see page 78) | This is record Ektron.Cms.Controls.Checkout.ControlPhase. |

### Ektron.Cms.Controls.Checkout.ControlPhase Enumeration

**C#**

```
public enum ControlPhase {
}
```

**Visual Basic**

```
Public Enum ControlPhase
End Enum
```

**File**

Checkout.cs

### Description

This is record Ektron.Cms.Controls.Checkout.ControlPhase.

## Checkout Methods

The methods of the Checkout class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 79) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 79) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 79) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## Checkout.Fill Method

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

**C#**

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

**Visual Basic**

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

## Checkout.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

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

**Visual Basic**

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

## Checkout.Wrap Method

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

**C#**

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

**Visual Basic**

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

### Parameters

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

### Returns

## Checkout Properties

The properties of the Checkout class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Cart (see page 80) | Provides access to the current Cart (Basket (see page 1215)) object. |
| | CurrentPhase (see page 81) | Reports or selects the current page (or phase) of the control, particularly useful if wizard mode is enabled. |
| | Customer (see page 81) | Provides access to the current CustomerData (see page 1255) object. |
| | DefaultCountryId (see page 81) | The ID of the default country that appears in the Billing Address and Shipping Address. Set this ID to the country from which the majority of your customers will be making purchases. If the site visitor is from a different country, he can change it when editing the Billing or Shipping Address. |
| | DisplayXslt (see page 82) | Specify an external XSLT file to render the control. By default, the control uses Checkout.xsl. This file is located in: |
| | DynamicPhaseParameter (see page 82) | The name of the parameter on the query string that identifies desired phase ID. |
| | FriendlyErrorMessage (see page 82) | The message that is shown when an unhandled error occurs. Details are sent to the event log. The default message is |
| | IsSSLRequired (see page 83) | When set to True, switches to an SSL encrypted URL. For the SSL encryption to work, you must have an SSL certificate installed. Information on SSL can be found at the following Web sites:<br>• http://www.tech-faq.com/understanding-ssl.shtml<br>• http://www.verisign.com/ssl/ssl-information-center/how-ssl-security-works/index.html<br>**Note:** Ektron, Inc. is not associated with TechFAQ or VeriSign. However, both these sites offer very good, detailed explanations of Secure Sockets Layer. |
| | LoadingImage (see page 83) | The image to display while the control is fetching data. The default is: |
| | PaymentMethod (see page 83) | Specifies the class to call to collect the payment information. Change this to accept other types of payments, such as eChecks. |
| | ShowErrorDetails (see page 84) | When true, displays detailed error message, otherwise shows friendly error message (default). |
| | Stylesheet (see page 84) | Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | TemplateCart (see page 84) | The URL path of the template that contains the Cart (see page 80) server control. This path can be relative or absolute. When a path is entered, a link appears in the Review Order part of the process that allows a user to navigate back to the template containing the Cart (see page 80) server control. |
| | TemplateOrderHistory (see page 85) | The URL path of the template that contains the OrderList (see page 395) server control. This path can be relative or absolute. When a path is entered, a link appears at the end of checkout process that allows a user to navigate to the template containing the OrderList (see page 395) server control. |
| | TemplateRecoverPassword (see page 85) | The URL path of the template that helps site visitors recover their password. This path can be relative or absolute. When a path is entered and a site visitor has not logged in, a link appears at the beginning of checkout process that allows him to navigate to a template containing information to recover his password. The Membership (see page 350) server control contains an option for recovering passwords. |
| | TemplateShopping (see page 86) | The URL path of the template that allows the site visitor to continue shopping. When a path is entered, a link appears at the end of checkout process that allows a user to navigate to the template that would allow them to continue shopping; for example, the template containing a ProductList (see page 418) or ProductSearch (see page 426) server control. |
| | UseMsAjax (see page 86) | If true, uses Microsoft Ajax to perform callback. Otherwise, the control uses Ektron Library (see page 664) callback (default). |
| | ValidatePhase (see page 86) | When true (default) ensures current phase is allowed. Set to false to override, allowing any phase order. |

## Checkout.Cart Property

Provides access to the current Cart (Basket (see page 1215)) object.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(Ektron.Cms.Commerce.Basket))]
[Description("")]
[Bindable(true)]
[DefaultValue("null")]
public Ektron.Cms.Commerce.Basket Cart;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(Ektron.Cms.Commerce.Basket))>
<Description("")>
```

```
<Bindable(true)>
<DefaultValue("null")>
Public Property Cart() As Ektron.Cms.Commerce.Basket
```

### Checkout.CurrentPhase Property

Reports or selects the current page (or phase) of the control, particularly useful if wizard mode is enabled.

**C#**

```
[Category("Ektron")]
[Browsable(false)]
[TypeConverter(typeof(ControlPhase))]
[Description("Reports or selects the current page (or phase) of the control, particularly
useful if wizard mode is enabled.")]
[DefaultValue("")]
public ControlPhase CurrentPhase;
```

**Visual Basic**

```
<Category("Ektron")>
<Browsable(false)>
<TypeConverter(typeof(ControlPhase))>
<Description("Reports or selects the current page (or phase) of the control, particularly
useful if wizard mode is enabled.")>
<DefaultValue("")>
Public Property CurrentPhase() As ControlPhase
```

### Checkout.Customer Property

Provides access to the current CustomerData (see page 1255) object.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(Ektron.Cms.Commerce.Basket))]
[Description("")]
[Bindable(true)]
[DefaultValue("null")]
public Ektron.Cms.Commerce.CustomerData Customer;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(Ektron.Cms.Commerce.Basket))>
<Description("")>
<Bindable(true)>
<DefaultValue("null")>
Public Property Customer() As Ektron.Cms.Commerce.CustomerData
```

### Checkout.DefaultCountryId Property

The ID of the default country that appears in the Billing Address and Shipping Address. Set this ID to the country from which the majority of your customers will be making purchases. If the site visitor is from a different country, he can change it when editing the Billing or Shipping Address.

**C#**

```
[Category("Ektron")]
[Description("The ID of the default country that appears in the Billing Address and Shipping
Address. Set this ID to the country from which the majority of your customers will be making
purchases. If the site visitor is from a different country, he can change it when editing the
Billing or Shipping Address..")]
public int DefaultCountryId;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("The ID of the default country that appears in the Billing Address and Shipping
```

```
Address. Set this ID to the country from which the majority of your customers will be making
purchases. If the site visitor is from a different country, he can change it when editing the
Billing or Shipping Address..")>
Public Property DefaultCountryId() As Integer
```

### Checkout.DisplayXslt Property

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

```
[SiteRoot]\Workarea\Xslt\Commerce\Checkout\Standard
```

**C#**

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

**Visual Basic**

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

### Checkout.DynamicPhaseParameter Property

The name of the parameter on the query string that identifies desired phase ID.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The name of the parameter on the query string that identifies desired phase
ID.")]
[DefaultValue("phase")]
public string DynamicPhaseParameter;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The name of the parameter on the query string that identifies desired phase
ID.")>
<DefaultValue("phase")>
Public Property DynamicPhaseParameter() As string
```

### Checkout.FriendlyErrorMessage Property

The message that is shown when an unhandled error occurs. Details are sent to the event log. The default message is

```
We're sorry, an error occurred while processing your request. Please try again later...
```

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The message that is shown when an unhandled error occurs. Details are sent to
the event log. The default message is \"We're sorry, an error occurred while processing your
request. Please try again later...\"")]
[DefaultValue("We're sorry, an error occurred...")]
public string FriendlyErrorMessage;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The message that is shown when an unhandled error occurs. Details are sent to
the event log. The default message is \"We're sorry, an error occurred while processing your
request. Please try again later...\"")>
<DefaultValue("We're sorry, an error occurred...")>
Public Property FriendlyErrorMessage() As string
```

### Checkout.IsSSLRequired Property

When set to True, switches to an SSL encrypted URL. For the SSL encryption to work, you must have an SSL certificate installed. Information on SSL can be found at the following Web sites:

- http://www.tech-faq.com/understanding-ssl.shtml
- http://www.verisign.com/ssl/ssl-information-center/how-ssl-security-works/index.html

**Note:** Ektron, Inc. is not associated with TechFAQ or VeriSign. However, both these sites offer very good, detailed explanations of Secure Sockets Layer.

**C#**

```
[Category("Ektron")]
[Description("When set to True, switches to an SSL encrypted URL. For the SSL encryption to
work, you must have an SSL certificate installed.")]
[TypeConverter(typeof(bool))]
[DefaultValue(false)]
public bool IsSSLRequired;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("When set to True, switches to an SSL encrypted URL. For the SSL encryption to
work, you must have an SSL certificate installed.")>
<TypeConverter(typeof(bool))>
<DefaultValue(false)>
Public Property IsSSLRequired() As bool
```

### Checkout.LoadingImage Property

The image to display while the control is fetching data. The default is:

```
[siteroot]\Workarea\images\application\ajax-loader_circle_lg.gif.
```

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The image to display while the control is fetching data. The default is:
[siteroot]\\Workarea\\images\\application\\ajax-loader_circle_lg.gif.")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The image to display while the control is fetching data. The default is:
[siteroot]\\Workarea\\images\\application\\ajax-loader_circle_lg.gif.")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

### Checkout.PaymentMethod Property

Specifies the class to call to collect the payment information. Change this to accept other types of payments, such as eChecks.

**C#**

```
[Category("Ektron")]
[Browsable(false)]
[Description("Specifies the PaymentMethod used for the order. Change this to accept other
types of payments, such as eChecks.")]
public PaymentMethod PaymentMethod;
```

**Visual Basic**

```
<Category("Ektron")>
<Browsable(false)>
<Description("Specifies the PaymentMethod used for the order. Change this to accept other
types of payments, such as eChecks.")>
Public Property PaymentMethod() As PaymentMethod
```

### Checkout.ShowErrorDetails Property

When true, displays detailed error message, otherwise shows friendly error message (default).

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When true, displays detailed error message, otherwise shows friendly error
message (default).")]
[DefaultValue("false")]
public bool ShowErrorDetails;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When true, displays detailed error message, otherwise shows friendly error
message (default).")>
<DefaultValue("false")>
Public Property ShowErrorDetails() As bool
```

### Checkout.Stylesheet Property

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

**C#**

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(CheckoutCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/Checkout.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(CheckoutCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/Checkout.css")>
Public Property Stylesheet() As string
```

### Checkout.TemplateCart Property

The URL path of the template that contains the Cart (see page 80) server control. This path can be relative or absolute. When a path is entered, a link appears in the Review Order part of the process that allows a user to navigate back to the template containing the Cart (see page 80) server control.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The URL path of the template that contains the Cart server control. This path
can be relative or absolute. When a path is entered, a link appears in the Review Order part
of the process that allows a user to navigate back to the template containing the Cart server
control.")]
[DefaultValue("")]
public string TemplateCart;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The URL path of the template that contains the Cart server control. This path
can be relative or absolute. When a path is entered, a link appears in the Review Order part
of the process that allows a user to navigate back to the template containing the Cart server
control.")>
<DefaultValue("")>
Public Property TemplateCart() As string
```

### Checkout.TemplateOrderHistory Property

The URL path of the template that contains the OrderList (see page 395) server control. This path can be relative or absolute. When a path is entered, a link appears at the end of checkout process that allows a user to navigate to the template containing the OrderList (see page 395) server control.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The URL path of the template that contains the OrderList server control. This
path can be relative or absolute. When a path is entered, a link appears at the end of
checkout process that allows a user to navigate to the template containing the OrderList
server control.")]
[DefaultValue("")]
public string TemplateOrderHistory;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The URL path of the template that contains the OrderList server control. This
path can be relative or absolute. When a path is entered, a link appears at the end of
checkout process that allows a user to navigate to the template containing the OrderList
server control.")>
<DefaultValue("")>
Public Property TemplateOrderHistory() As string
```

### Checkout.TemplateRecoverPassword Property

The URL path of the template that helps site visitors recover their password. This path can be relative or absolute. When a path is entered and a site visitor has not logged in, a link appears at the beginning of checkout process that allows him to navigate to a template containing information to recover his password. The Membership (see page 350) server control contains an option for recovering passwords.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The URL path of the template that helps site visitors recover their password.
This path can be relative or absolute. When a path is entered and a site visitor has not
logged in, a link appears at the beginning of checkout process that allows him to navigate to
a template containing information to recover his password. The Membership server control
contains an option for recovering passwords.")]
[DefaultValue("")]
```

```
public string TemplateRecoverPassword;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The URL path of the template that helps site visitors recover their password.
This path can be relative or absolute. When a path is entered and a site visitor has not
logged in, a link appears at the beginning of checkout process that allows him to navigate to
a template containing information to recover his password. The Membership server control
contains an option for recovering passwords.")>
<DefaultValue("")>
Public Property TemplateRecoverPassword() As string
```

### Checkout.TemplateShopping Property

The URL path of the template that allows the site visitor to continue shopping. When a path is entered, a link appears at the end of checkout process that allows a user to navigate to the template that would allow them to continue shopping; for example, the template containing a ProductList (see page 418) or ProductSearch (see page 426) server control.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("The URL path of the template that allows the site visitor to continue shopping.
When a path is entered, a link appears at the end of checkout process that allows a user to
navigate to the template that would allow them to continue shopping; for example, the template
containing a ProductList or ProductSearch server control.")]
[DefaultValue("")]
public string TemplateShopping;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("The URL path of the template that allows the site visitor to continue shopping.
When a path is entered, a link appears at the end of checkout process that allows a user to
navigate to the template that would allow them to continue shopping; for example, the template
containing a ProductList or ProductSearch server control.")>
<DefaultValue("")>
Public Property TemplateShopping() As string
```

### Checkout.UseMsAjax Property

If true, uses Microsoft Ajax to perform callback. Otherwise, the control uses Ektron Library (see page 664) callback (default).

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, uses Microsoft Ajax to perform callback. Otherwise, the control uses
Ektron Library callback (default).")]
[DefaultValue("false")]
public bool UseMsAjax;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, uses Microsoft Ajax to perform callback. Otherwise, the control uses
Ektron Library callback (default).")>
<DefaultValue("false")>
Public Property UseMsAjax() As bool
```

### Checkout.ValidatePhase Property

When true (default) ensures current phase is allowed. Set to false to override, allowing any phase order.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When true (default) ensures current phase is allowed. Set to false to override,
allowing any phase order")]
[DefaultValue("true")]
public bool ValidatePhase;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When true (default) ensures current phase is allowed. Set to false to override,
allowing any phase order")>
<DefaultValue("true")>
Public Property ValidatePhase() As bool
```

## Checkout Delegates

The delegates of the Checkout class are listed here.

**Public Delegates**

| Name | Description |
|---|---|
| PhaseChangeEventHandler (see page 87) | For the PhaseChange (see page 88) event. |
| PostLoginEventHandler (see page 87) | For the Post-Login (see page 337) event. |

## Checkout.PhaseChangeEventHandler Delegate

For the PhaseChange (see page 88) event.

**C#**

```
public delegate void PhaseChangeEventHandler(object sender, PhaseChangeEventArgs e);
```

**Visual Basic**

```
Public Delegate Sub PhaseChangeEventHandler(object sender, PhaseChangeEventArgs e) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| sender | |
| e | |

## Checkout.PostLoginEventHandler Delegate

For the Post-Login (see page 337) event.

**C#**

```
public delegate void PostLoginEventHandler(object sender, PostLoginEventArgs e);
```

**Visual Basic**

```
Public Delegate Sub PostLoginEventHandler(object sender, PostLoginEventArgs e) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| sender | |
| e | |

## Checkout Events

The events of the Checkout class are listed here.

### Public Events

| | Name | Description |
|---|---|---|
| | PhaseChange (see page 88) | Fires when the current phase (state) is about to change. Allows supervising phase-changes, in order to allow or prevent changing as needed. |
| | PostLogin (see page 88) | Fires when an Login (see page 337) event occurs. Allows inspecting or modifying callback parameters. |

### Checkout.PhaseChange Event

Fires when the current phase (state) is about to change. Allows supervising phase-changes, in order to allow or prevent changing as needed.

**C#**

```
public event PhaseChangeEventHandler PhaseChange;
```

**Visual Basic**

```
Public Event PhaseChange As PhaseChangeEventHandler
```

### Checkout.PostLogin Event

Fires when an Login (see page 337) event occurs. Allows inspecting or modifying callback parameters.

**C#**

```
public event PostLoginEventHandler PostLogin;
```

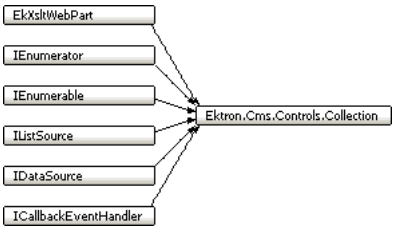
**Visual Basic**

```
Public Event PostLogin As PostLoginEventHandler
```

## Collection Class

Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.

### Class Hierarchy




**C#**

```
[DefaultProperty("DefaultCollectionID"), ToolboxData("<{0}:Collection
runat=server></{0}:Collection>"), PersistChildren(true)]
public class Collection : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultCollectionID"), ToolboxData("<{0}:Collection
runat=server></{0}:Collection>"), PersistChildren(true)>
Public Class Collection
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

**File**

Collection.cs

## Collection Methods

The methods of the Collection class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | CopyTo (see page 89) | 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 89) | Calls the database and updates the contents of the collection if the data is out of sync with the specified parameters. |
| | GetEnumerator (see page 89) | Gets a System.Collections.IEnumerator for the collection. |
| | GetRssFeed (see page 90) | This method returns a Collection (see page 88) RSS feed. |
| | GetView (see page 90) | Gets the view data using a dataset. |
| | GetViewNames (see page 90) | Gets the view names. |
| | MoveNext (see page 90) | Moves up the index of the Collection (see page 88). Returns false if the end of the collection has been surpassed. |
| | Reset (see page 91) | Brings the index of the collection back to zero. Does not clear the collection. |
| | this (see page 91) | Index operator into a strongly typed dataset holding some common properties for a Collection (see page 88) item. |
| | ToString (see page 91) | Wraps the resulting collection with the Ektron toolbars, which will only display when a user is logged in. |
| | Wrap (see page 91) | Overrideable method that's used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## Collection.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

**C#**

```
public void CopyTo(Array array, int i);
```

**Visual Basic**

```
Public Function CopyTo(array As Array, i As Integer) As void
```

## Collection.Fill Method

Calls the database and updates the contents of the collection if the data is out of sync with the specified parameters.

**C#**

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

**Visual Basic**

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

## Collection.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

**C#**

```
public IEnumerator GetEnumerator();
```

**Visual Basic**

```
Public Function GetEnumerator() As IEnumerator
```

### Collection.GetRssFeed Method

This method returns a Collection (see page 88) 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 88). 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 88) 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 |
|---|---|---|
| | AddText (see page 92) | Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 788)' could be changed to 'Add Report') |
| | CacheInterval (see page 93) | Sets the cache interval in second(s). |
| | CollectionDescription (see page 93) | Returns the description of the given folder ID. |
| | CollectionTitle (see page 93) | Returns the name of the given folder ID. |

| | | |
|---|---|---|
| | ContentParameter (see page 93) | Checks the QueryString for this value and replaces the collection with that content block when specified. Leave blank to always display the collection. |
| | Count (see page 94) | Indicates the number of elements in your Collection (see page 88). |
| | Current (see page 94) | Gets the current object in the collection. |
| | DefaultCollectionID (see page 94) | Gets or Sets the Default CollectionID for this Control. |
| | DynamicParameter (see page 94) | The parameter to query for a Dynamic Collection (see page 88). |
| | EkItems (see page 95) | Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 88). Holds more information than the exposed strongly typed this[i] index operator. |
| | EnablePaging (see page 95) | Enables paging if the total number of records exceeds the MaxResult value. |
| | GetHtml (see page 95) | If set to true, retrieves the content (Html body) of every content block in the collection for display. |
| | IncludeIcons (see page 96) | Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser. |
| | IsSynchronized (see page 96) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | LinkTarget (see page 96) | Defines the way a link acts when a link is clicked. Choices are:<br>• **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named.<br>• **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target.<br>• **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.<br>• **_top** - This target makes the link load in... more (see page 96) |
| | MarkupLanguage (see page 97) | Enter the name of a markup file that manages the display of this server control. |
| | MaxResults (see page 97) | Maximum number of results to be returned. If set to zero, there is no limit. |
| | Random (see page 97) | If set to true, grabs a single content block randomly from the collection. Set to false for all data in the collection. |
| | SuppressWrapperTags (see page 97) | Suppresses the output of span/div tags around the control. |
| | SyncRoot (see page 98) | Gets an object that's used to synchronize the collection. |
| | XmlDoc (see page 98) | Returns the Xslt object for the collection. |

### Collection.AddText Property

Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 788)' could be changed to 'Add Report')

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows overriding the derived controls default Add-menu button text (i.e. 'Add
HTML Content' could be changed to 'Add Report')")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Visible)
]
[Browsable(true)]
public virtual string AddText;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows overriding the derived controls default Add-menu button text (i.e. 'Add
HTML Content' could be changed to 'Add Report')")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Visible)
>
<Browsable(true)>
```

```
Public virtual Property AddText() As string
```

### Collection.CacheInterval Property

Sets the cache interval in second(s).

**C#**

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

**Visual Basic**

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

### Collection.CollectionDescription Property

Returns the description of the given folder ID.

**C#**

```
public string CollectionDescription;
```

**Visual Basic**

```
Public Property CollectionDescription() As string
```

### Collection.CollectionTitle Property

Returns the name of the given folder ID.

**C#**

```
public string CollectionTitle;
```

**Visual Basic**

```
Public Property CollectionTitle() As string
```

### Collection.ContentParameter Property

Checks the QueryString for this value and replaces the collection with that content block when specified. Leave blank to always display the collection.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and replaces the collection with that
content block when specified. Leave blank to always display the collection.")]
[Personalizable(), WebBrowsable()]
public string ContentParameter;
```

**Visual Basic**

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

```
content block when specified. Leave blank to always display the collection.")>
<Personalizable(), WebBrowsable()>
Public Property ContentParameter() As string
```

### Collection.Count Property

Indicates the number of elements in your Collection (see page 88).

**C#**

```
[Category("Data")]
[Description("Indicates the number of elements in your Collection.")]
public int Count;
```

**Visual Basic**

```
<Category("Data")>
<Description("Indicates the number of elements in your Collection.")>
Public Property Count() As Integer
```

### Collection.Current Property

Gets the current object in the collection.

**C#**

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

**Visual Basic**

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

### Collection.DefaultCollectionID Property

Gets or Sets the Default CollectionID for this Control.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Editor(typeof(CollectionUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("Gets or Sets the Default CollectionID for this Control.")]
[Personalizable(), WebBrowsable()]
public long DefaultCollectionID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Editor(typeof(CollectionUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("Gets or Sets the Default CollectionID for this Control.")>
<Personalizable(), WebBrowsable()>
Public Property DefaultCollectionID() As long
```

### Collection.DynamicParameter Property

The parameter to query for a Dynamic Collection (see page 88).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("The parameter to query for a Dynamic Collection.")]
[TypeConverter(typeof(DynamicCollectionTypeConverter))]
[Personalizable(), WebBrowsable()]
```

```
public string DynamicParameter;
```

**Visual Basic**

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

### Collection.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 88). Holds more information than the exposed strongly typed this[i] index operator.

**C#**

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItems;
```

**Visual Basic**

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

### Collection.EnablePaging Property

Enables paging if the total number of records exceeds the MaxResult value.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enables paging if the total number of records exceeds the MaxResult value. Use
this property with non zero value of MaxNumbers.")]
[Personalizable(), WebBrowsable()]
public bool EnablePaging;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enables paging if the total number of records exceeds the MaxResult value. Use
this property with non zero value of MaxNumbers.")>
<Personalizable(), WebBrowsable()>
Public Property EnablePaging() As bool
```

### Collection.GetHtml Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")]
[Personalizable(), WebBrowsable()]
public bool GetHtml;
```

**Visual Basic**

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

```
<Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")>
<Personalizable(), WebBrowsable()>
Public Property GetHtml() As bool
```

### Collection.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.

**C#**

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
[Personalizable(), WebBrowsable()]
public bool IncludeIcons;
```

**Visual Basic**

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
<Personalizable(), WebBrowsable()>
Public Property IncludeIcons() As bool
```

### Collection.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

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

**Visual Basic**

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

### Collection.LinkTarget Property

Defines the way a link acts when a link is clicked. Choices are:

- **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named.
- **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target.
- **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent.
- **_top** - This target makes the link load in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

**C#**

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

**Visual Basic**

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

### Collection.MarkupLanguage Property

Enter the name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

### Collection.MaxResults Property

Maximum number of results to be returned. If set to zero, there is no limit.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
[Personalizable(), WebBrowsable()]
public int MaxResults;
```

**Visual Basic**

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

### Collection.Random Property

If set to true, grabs a single content block randomly from the collection. Set to false for all data in the collection.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, grabs a single content block randomly from the collection. Set
to false for all data in the collection.")]
[Personalizable(), WebBrowsable()]
public bool Random;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, grabs a single content block randomly from the collection. Set
to false for all data in the collection.")>
<Personalizable(), WebBrowsable()>
Public Property Random() As bool
```

### Collection.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

**C#**

```
public override bool SuppressWrapperTags;
```

**Visual Basic**

```
Public override Property SuppressWrapperTags() As bool
```

### Collection.SyncRoot Property

Gets an object that's used to synchronize the collection.

**C#**

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

**Visual Basic**

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

### Collection.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

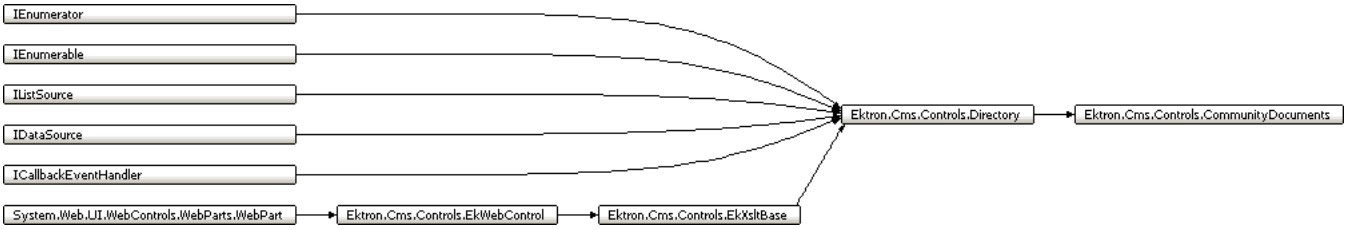
```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

## CommunityDocuments Class

**Class Hierarchy**




**C#**

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

**Visual Basic**

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

**File**

communitydocuments.cs

**Description**

This is class Ektron.Cms.Controls.CommunityDocuments.

## CommunityDocuments Methods

The methods of the CommunityDocuments class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 99) | Populates the data for the Taxonomy. |

## CommunityDocuments.Fill Method

Populates the data for the Taxonomy.

**C#**

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

**Visual Basic**

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

## CommunityDocuments Properties

The properties of the CommunityDocuments class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AddItemFolderId (see page 99) | This is a readonly property. |
| | DefaultObjectID (see page 100) | Gets or Sets the default user ID for this control. This is the user ID that will be shown if the DynamicParameter (see page 195) does not get a value. (e.g. userprofile.aspx?userid=2 will get the value 2 for a DynamicParameter (see page 195) of "id", and will display DefaultUserID otherwise.)<br>The object ID for the object that the social bar will be drawn for. |
| | DynamicContentBoxHeight (see page 100) | Get or sets the height of dynamic content. Value should be in pixels. |
| | DynamicContentBoxWidth (see page 100) | Get or sets the with of dynamic content. Value should be in pixels. |
| | DynamicContentTemplate (see page 101) | Get or sets the template of dynamic content. If specified control will not use dynamic box. |
| | EnableAjax (see page 101) | This is a readonly property. |
| | EnableSearch (see page 101) | This is a readonly property. |
| | MemberMenuActive (see page 101) | This is a readonly property. |
| | ObjectType (see page 102) | Select the type of object with which this control is associated. |
| | Recursive (see page 102) | This is a readonly property. |
| | ShowAllChecked (see page 102) | This is a readonly property. |
| | Stylesheet (see page 102) | Specify the style sheet relative or absolute. Leave blank for default. |
| | SuppressWrapperTags (see page 103) | This is a readonly property. |
| | TaxonomyDepth (see page 103) | This is a readonly property. |
| | TaxonomyId (see page 103) | This is a readonly property. |

## CommunityDocuments.AddItemFolderId Property

This is a readonly property.

**C#**

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
```

```
public override long AddItemFolderId;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property AddItemFolderId() As long
```

### CommunityDocuments.DefaultObjectID Property

Gets or Sets the default user ID for this control. This is the user ID that will be shown if the DynamicParameter (see page 195) does not get a value. (e.g. userprofile.aspx?userid=2 will get the value 2 for a DynamicParameter (see page 195) of "id", and will display DefaultUserID otherwise.)

The object ID for the object that the social bar will be drawn for.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Gets or sets the default object ID for this control. This is the object ID that
will be shown if the DynamicParameter property does not get a value.")]
public long DefaultObjectID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Gets or sets the default object ID for this control. This is the object ID that
will be shown if the DynamicParameter property does not get a value.")>
Public Property DefaultObjectID() As long
```

### CommunityDocuments.DynamicContentBoxHeight Property

Get or sets the height of dynamic content. Value should be in pixels.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(520)]
[Description("Get or sets the height of dynamic content box. Value should be in pixels.")]
public int DynamicContentBoxHeight;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(520)>
<Description("Get or sets the height of dynamic content box. Value should be in pixels.")>
Public Property DynamicContentBoxHeight() As Integer
```

### CommunityDocuments.DynamicContentBoxWidth Property

Get or sets the with of dynamic content. Value should be in pixels.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(790)]
[Description("Get or sets the width of dynamic content box. Value should be enter in pixels.")]
public int DynamicContentBoxWidth;
```

**Visual Basic**

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

```
<Category("Ektron")>
<DefaultValue(790)>
<Description("Get or sets the width of dynamic content box. Value should be enter in pixels.")>
Public Property DynamicContentBoxWidth() As Integer
```

### CommunityDocuments.DynamicContentTemplate Property

Get or sets the template of dynamic content. If specified control will not use dynamic box.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Get or sets the template of dynamic content. Leave blank to use dynamic box.")]
public string DynamicContentTemplate;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Get or sets the template of dynamic content. Leave blank to use dynamic box.")>
Public Property DynamicContentTemplate() As string
```

### CommunityDocuments.EnableAjax Property

This is a readonly property.

**C#**

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override bool EnableAjax;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property EnableAjax() As bool
```

### CommunityDocuments.EnableSearch Property

This is a readonly property.

**C#**

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override Boolean EnableSearch;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property EnableSearch() As Boolean
```

### CommunityDocuments.MemberMenuActive Property

This is a readonly property.

**C#**

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
```

```
public override bool MemberMenuActive;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property MemberMenuActive() As bool
```

### CommunityDocuments.ObjectType Property

Select the type of object with which this control is associated.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select the type of object with which this control is associated.")]
[DefaultValue(CommunityDocumentsObjectType.User)]
public CommunityDocumentsObjectType ObjectType;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select the type of object with which this control is associated.")>
<DefaultValue(CommunityDocumentsObjectType.User)>
Public Property ObjectType() As CommunityDocumentsObjectType
```

### CommunityDocuments.Recursive Property

This is a readonly property.

**C#**

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override bool Recursive;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property Recursive() As bool
```

### CommunityDocuments.ShowAllChecked Property

This is a readonly property.

**C#**

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override bool ShowAllChecked;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property ShowAllChecked() As bool
```

### CommunityDocuments.Stylesheet Property

Specify the style sheet relative or absolute. Leave blank for default.

**C#**

```
[Description("Specify the style sheet relative or absolute. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(CssTypeConverter))]
[DefaultValue("None - Use Default")]
public string Stylesheet;
```

**Visual Basic**

```
<Description("Specify the style sheet relative or absolute. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(CssTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property Stylesheet() As string
```

### CommunityDocuments.SuppressWrapperTags Property

This is a readonly property.

**C#**

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override bool SuppressWrapperTags;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property SuppressWrapperTags() As bool
```

### CommunityDocuments.TaxonomyDepth Property

This is a readonly property.

**C#**

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override int TaxonomyDepth;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
Public override Property TaxonomyDepth() As Integer
```

### CommunityDocuments.TaxonomyId Property

This is a readonly property.

**C#**

```
[Category("Ektron")]
[Description("Inherited from Taxonomy Server Control.")]
[Browsable(false)]
public override long TaxonomyId;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Inherited from Taxonomy Server Control.")>
<Browsable(false)>
```

```
Public override Property TaxonomyId() As long
```

## CommunityGroupBrowser Class

Summary description for Message board

### Class Hierarchy

### C#

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

### Visual Basic

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

### File

CommunityGroupBrowser.cs

### CommunityGroupBrowser Methods

The methods of the CommunityGroupBrowser class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 104) | Populates the data for the CommunityGroupMembers (see page 116). |
| | ToString (see page 104) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 104) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### CommunityGroupBrowser.Fill Method

Populates the data for the CommunityGroupMembers (see page 116).

### C#

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

### Visual Basic

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

### CommunityGroupBrowser.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

### C#

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

### Visual Basic

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

### CommunityGroupBrowser.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

## CommunityGroupBrowser Properties

The properties of the CommunityGroupBrowser class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AvatarHeight (see page 105) | Gets or Sets the avatar height. 0 is unlimited. |
| | AvatarWidth (see page 105) | Gets or Sets the avatar width. 0 is unlimited. |
| | CacheInterval (see page 106) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DisplayXslt (see page 106) | Select the type of review to display, or enter the path to a custom XSLT. (optional).<br>Choices are: |
| | Link (see page 107) | Gets or Sets the url for the community group links. |
| | LinkTarget (see page 107) | Gets or Sets the target attribute for the community group links. |
| | MarkupLanguage (see page 107) | Enter name of a markup file that manages the display of this server control. |
| | MaxResults (see page 107) | The number of community groups to show per page when EnablePaging is true. If set to zero, there is no limit. |
| | Stylesheet (see page 108) | Specifies the Css to which this control will provide a link. |
| | TagTemplate (see page 108) | Gets or Sets the template-url for the group tag links. |
| | TagTemplateTarget (see page 108) | Gets or Sets the target (window) attribute for the group-tag links. |
| | TaxonomyId (see page 108) | Enter the ID number of the taxonomy or category to appear in this server control. Root is 0. |
| | XmlDoc (see page 109) | Returns the Xslt object for the collection. |

## CommunityGroupBrowser.AvatarHeight Property

Gets or Sets the avatar height. 0 is unlimited.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") ]
public int AvatarHeight;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") >
Public Property AvatarHeight() As Integer
```

## CommunityGroupBrowser.AvatarWidth Property

Gets or Sets the avatar width. 0 is unlimited.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") ]
public int AvatarWidth;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") >
Public Property AvatarWidth() As Integer
```

### CommunityGroupBrowser.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### CommunityGroupBrowser.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

Choices are:

**C#**

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

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltCommunityGroupMembersDataTypeConverter))>
<DefaultValue("ListMembersTable")>
```

```
<Personalizable()>
Public override Property DisplayXslt() As string
```

### CommunityGroupBrowser.Link Property

Gets or Sets the url for the community group links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("The community group link.
{0} is the group ID and {1} is the group name.") ]
public string Link;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("The community group link.
{0} is the group ID and {1} is the group name.") >
Public Property Link() As string
```

### CommunityGroupBrowser.LinkTarget Property

Gets or Sets the target attribute for the community group links.

**C#**

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

**Visual Basic**

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

### CommunityGroupBrowser.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

### CommunityGroupBrowser.MaxResults Property

The number of community groups to show per page when EnablePaging is true. If set to zero, there is no limit.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The number of community groups to show per page. If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The number of community groups to show per page. If set to zero, there is no
```

```
limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

### CommunityGroupBrowser.Stylesheet Property

Specifies the Css to which this control will provide a link.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

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

### CommunityGroupBrowser.TagTemplate Property

Gets or Sets the template-url for the group tag links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") ]
public string TagTemplate;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") >
Public Property TagTemplate() As string
```

### CommunityGroupBrowser.TagTemplateTarget Property

Gets or Sets the target (window) attribute for the group-tag links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") ]
public ItemLinkTargets TagTemplateTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

### CommunityGroupBrowser.TaxonomyId Property

Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)]
[Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long TaxonomyId;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)>
<Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property TaxonomyId() As long
```

### CommunityGroupBrowser.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

## CommunityGroupList Class

Summary description for Message board

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

CommunityGroupList.cs

### CommunityGroupList Methods

The methods of the CommunityGroupList class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 110) | Populates the data for the CommunityGroupMembers (see page 116). |
| | ToString (see page 110) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 110) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### CommunityGroupList.Fill Method

Populates the data for the CommunityGroupMembers (see page 116).

**C#**

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

**Visual Basic**

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

### CommunityGroupList.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

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

**Visual Basic**

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

### CommunityGroupList.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

### CommunityGroupList Properties

The properties of the CommunityGroupList class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AllowGroupCreation (see page 111) | If the user has proper permissions, shows the Create Group option. |
| | AvatarHeight (see page 111) | Gets or Sets the avatar height. 0 is unlimited. |
| | AvatarWidth (see page 111) | Gets or Sets the avatar width. 0 is unlimited. |
| | CacheInterval (see page 111) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultUserID (see page 112) | When in DisplayMode (see page 112) is "MyGroups", the ID for the user that the control will use. |
| | DisplayMode (see page 112) | The type of community group list you would like to see. |
| | DisplayXslt (see page 112) | Select the type of review to display, or enter the path to a custom XSLT. (optional).<br>Choices are: |
| | DynamicUserParameter (see page 113) | When in DisplayMode (see page 112) is "MyGroups" or "MyPendingGroups", Checks the QueryString for this value and uses that value when specified. Leave blank to always use the default user ID. |

| | | |
|---|---|---|
| | LanguageId (see page 113) | |
| | Link (see page 114) | Gets or Sets the url for the community group links. |
| | LinkTarget (see page 114) | Gets or Sets the target attribute for the community group links. |
| | MarkupLanguage (see page 114) | Enter name of a markup file that manages the display of this server control. |
| | MaxResults (see page 114) | The number of community groups to show per page when EnablePaging is true. If set to zero, there is no limit. |
| | ShowMySortingOptions (see page 115) | Should the My Groups, Pending and Invited Sorting Options Be Shown? |
| | ShowSortingOptions (see page 115) | Should the Newest, Most Popular and Group Name Sorting Options Be Shown? |
| | Stylesheet (see page 115) | Specifies the Css to which this control will provide a link. |
| | TagTemplate (see page 115) | Gets or Sets the template-url for the group tag links. |
| | TagTemplateTarget (see page 116) | Gets or Sets the target (window) attribute for the group-tag links. |
| | XmlDoc (see page 116) | Returns the Xslt object for the collection. |

### CommunityGroupList.AllowGroupCreation Property

If the user has proper permissions, shows the Create Group option.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(false), Description("If the user has proper
permissions, shows the Create Group option.") ]
public bool AllowGroupCreation;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(false), Description("If the user has proper
permissions, shows the Create Group option.") >
Public Property AllowGroupCreation() As bool
```

### CommunityGroupList.AvatarHeight Property

Gets or Sets the avatar height. 0 is unlimited.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") ]
public int AvatarHeight;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") >
Public Property AvatarHeight() As Integer
```

### CommunityGroupList.AvatarWidth Property

Gets or Sets the avatar width. 0 is unlimited.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") ]
public int AvatarWidth;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") >
Public Property AvatarWidth() As Integer
```

### CommunityGroupList.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a

control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

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

**Visual Basic**

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

### CommunityGroupList.DefaultUserID Property

When in DisplayMode (see page 112) is "MyGroups", the ID for the user that the control will use.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("When in DisplayMode is \"MyGroups\", the ID for the user that the control will
use.")]
public long DefaultUserID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("When in DisplayMode is \"MyGroups\", the ID for the user that the control will
use.")>
Public Property DefaultUserID() As long
```

### CommunityGroupList.DisplayMode Property

The type of community group list you would like to see.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Name), Description("The type of
community group list you would like to see.")]
public eDisplayMode DisplayMode;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Name), Description("The type of
community group list you would like to see.")>
Public Property DisplayMode() As eDisplayMode
```

### CommunityGroupList.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

Choices are:

**C#**

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

**Visual Basic**

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

### CommunityGroupList.DynamicUserParameter Property

When in DisplayMode (see page 112) is "MyGroups" or "MyPendingGroups", Checks the QueryString for this value and uses that value when specified. Leave blank to always use the default user ID.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("When in DisplayMode is \"MyGroups\" or \"MyPendingGroups\", checks the
QueryString for this value and uses that value when specified. Leave blank to always use the
default user ID.")]
[Personalizable()]
public string DynamicUserParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("When in DisplayMode is \"MyGroups\" or \"MyPendingGroups\", checks the
QueryString for this value and uses that value when specified. Leave blank to always use the
default user ID.")>
<Personalizable()>
Public Property DynamicUserParameter() As string
```

### CommunityGroupList.LanguageId Property

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("")]
[DefaultValue(0)]
public int LanguageId;
```

**Visual Basic**

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

```
<Browsable(false)>
<Category("Ektron")>
<Description("")>
<DefaultValue(0)>
Public Property LanguageId() As Integer
```

### CommunityGroupList.Link Property

Gets or Sets the url for the community group links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("The community group link.
{0} is the group ID and {1} is the group name.") ]
public string Link;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("The community group link.
{0} is the group ID and {1} is the group name.") >
Public Property Link() As string
```

### CommunityGroupList.LinkTarget Property

Gets or Sets the target attribute for the community group links.

**C#**

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

**Visual Basic**

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

### CommunityGroupList.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

### CommunityGroupList.MaxResults Property

The number of community groups to show per page when EnablePaging is true. If set to zero, there is no limit.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The number of community groups to show per page.If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

**Visual Basic**

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

```
<Category("Ektron")>
<DefaultValue(10)>
<Description("The number of community groups to show per page.If set to zero, there is no
limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

### CommunityGroupList.ShowMySortingOptions Property

Should the My Groups, Pending and Invited Sorting Options Be Shown?

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Description("For an admin or any
user, show the My Groups, Pending and Invited sorting options.")]
public bool ShowMySortingOptions;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Description("For an admin or any
user, show the My Groups, Pending and Invited sorting options.")>
Public Property ShowMySortingOptions() As bool
```

### CommunityGroupList.ShowSortingOptions Property

Should the Newest, Most Popular and Group Name Sorting Options Be Shown?

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Description("For an admin or any
user, show the Newest, Most Popular and Group Name sorting options.")]
public bool ShowSortingOptions;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Description("For an admin or any
user, show the Newest, Most Popular and Group Name sorting options.")>
Public Property ShowSortingOptions() As bool
```

### CommunityGroupList.Stylesheet Property

Specifies the Css to which this control will provide a link.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

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

### CommunityGroupList.TagTemplate Property

Gets or Sets the template-url for the group tag links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") ]
public string TagTemplate;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") >
Public Property TagTemplate() As string
```

### CommunityGroupList.TagTemplateTarget Property

Gets or Sets the target (window) attribute for the group-tag links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") ]
public ItemLinkTargets TagTemplateTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

### CommunityGroupList.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

## CommunityGroupMembers Class

Summary description for Message board

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

CommunityGroupMembers.cs

## CommunityGroupMembers Methods

The methods of the CommunityGroupMembers class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 117) | Populates the data for the CommunityGroupMembers (see page 116). |
| | ToString (see page 117) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 117) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## CommunityGroupMembers.Fill Method

Populates the data for the CommunityGroupMembers (see page 116).

**C#**

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

**Visual Basic**

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

## CommunityGroupMembers.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

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

**Visual Basic**

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

## CommunityGroupMembers.Wrap Method

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

**C#**

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

**Visual Basic**

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

### Parameters

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

### Returns

## CommunityGroupMembers Properties

The properties of the CommunityGroupMembers class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AvatarHeight (see page 118) | Gets or Sets the avatar height. 0 is unlimited. |
| | AvatarWidth (see page 118) | Gets or Sets the avatar width. 0 is unlimited. |

| | | |
|---|---|---|
| | CacheInterval (see page 118) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultCommunityGroupID (see page 119) | The ID for the community group that the control will use. |
| | DisplayMode (see page 119) | Display Mode |
| | DisplayXslt (see page 119) | Select the type of review to display, or enter the path to a custom XSLT. (optional).<br>Choices are: |
| | DynamicParameter (see page 120) | Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary. |
| | Link (see page 120) | Gets or Sets the url for the user links. |
| | LinkTarget (see page 120) | Gets or Sets the target attribute for the user links. |
| | MarkupLanguage (see page 121) | Enter name of a markup file that manages the display of this server control. |
| | MaxResults (see page 121) | Maximum number of results to be returned. |
| | Stylesheet (see page 121) | Specifies the Css to which this control will provide a link. |
| | XmlDoc (see page 122) | Returns the Xslt object for the collection. |

## CommunityGroupMembers.AvatarHeight Property

Gets or Sets the avatar height. 0 is unlimited.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") ]
public int AvatarHeight;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") >
Public Property AvatarHeight() As Integer
```

## CommunityGroupMembers.AvatarWidth Property

Gets or Sets the avatar width. 0 is unlimited.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") ]
public int AvatarWidth;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") >
Public Property AvatarWidth() As Integer
```

## CommunityGroupMembers.CacheInterval Property

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

**C#**

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

```
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

**Visual Basic**

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

### CommunityGroupMembers.DefaultCommunityGroupID Property

The ID for the community group that the control will use.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID for the community group that the control will use.")]
public long DefaultCommunityGroupID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID for the community group that the control will use.")>
Public Property DefaultCommunityGroupID() As long
```

### CommunityGroupMembers.DisplayMode Property

Display Mode

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Members), Description("This
property enables the user to show the community members list either with the Members or
Pending members display mode")]
public eDisplayMode DisplayMode;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Members), Description("This
property enables the user to show the community members list either with the Members or
Pending members display mode")>
Public Property DisplayMode() As eDisplayMode
```

### CommunityGroupMembers.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

Choices are:

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltCommunityGroupMembersDataTypeConverter))]
```

```
[DefaultValue("ListMembersTable")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

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

### CommunityGroupMembers.DynamicParameter Property

Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default community group ID.")]
[Personalizable()]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default community group ID.")>
<Personalizable()>
Public Property DynamicParameter() As string
```

### CommunityGroupMembers.Link Property

Gets or Sets the url for the user links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("The user link. {0} is the
user ID and {1} is the user displayname.") ]
public string Link;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("The user link. {0} is the
user ID and {1} is the user displayname.") >
Public Property Link() As string
```

### CommunityGroupMembers.LinkTarget Property

Gets or Sets the target attribute for the user links.

**C#**

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

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
```

```
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

### CommunityGroupMembers.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

### CommunityGroupMembers.MaxResults Property

Maximum number of results to be returned.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

**Visual Basic**

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

### CommunityGroupMembers.Stylesheet Property

Specifies the Css to which this control will provide a link.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

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

### CommunityGroupMembers.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

## CommunityGroupProfile Class

### Class Hierarchy

**C#**

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

**Visual Basic**

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

**File**

Group.cs

### CommunityGroupProfile Methods

The methods of the CommunityGroupProfile class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 122) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 122) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 123) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### CommunityGroupProfile.Fill Method

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

**C#**

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

**Visual Basic**

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

### CommunityGroupProfile.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

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

**Visual Basic**

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

### CommunityGroupProfile.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

### CommunityGroupProfile Properties

The properties of the CommunityGroupProfile class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DefaultGroupID (see page 123) | |
| | DisplayXslt (see page 124) | |
| | DynamicParameter (see page 124) | |
| | EkItem (see page 124) | |
| | GroupAdminLink (see page 125) | Gets or Sets the url for the group admin links. |
| | GroupAdminLinkTarget (see page 125) | Gets or Sets the target attribute for the group admin links. |
| | Stylesheet (see page 125) | Specifies the Css to which this control will provide a link. |
| | TagTemplate (see page 125) | Gets or Sets the template-url for the group tag links. |
| | TagTemplateTarget (see page 126) | Gets or Sets the target (window) attribute for the group-tag links. |
| | Text (see page 126) | Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display. |
| | XmlDoc (see page 126) | Returns the Xslt object for the collection. |

### CommunityGroupProfile.DefaultGroupID Property

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The ID for the group that is used")]
[TypeConverter(typeof(GroupUITypeEditor))]
[DefaultValue(0)]
public long DefaultGroupID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The ID for the group that is used")>
<TypeConverter(typeof(GroupUITypeEditor))>
```

```
<DefaultValue(0)>
Public Property DefaultGroupID() As long
```

### CommunityGroupProfile.DisplayXslt Property

**C#**

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

**Visual Basic**

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

### CommunityGroupProfile.DynamicParameter Property

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("To make this group control dynamic, select id. When you do, this server control
is attached to the community group passed as a URL parameter.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("To make this group control dynamic, select id. When you do, this server control
is attached to the community group passed as a URL parameter.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

### CommunityGroupProfile.EkItem Property

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The CommunityGroupData Object.")]
[TypeConverter(typeof(GroupUITypeEditor))]
[Browsable(false)]
public CommunityGroupData EkItem;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The CommunityGroupData Object.")>
<TypeConverter(typeof(GroupUITypeEditor))>
<Browsable(false)>
Public Property EkItem() As CommunityGroupData
```

### CommunityGroupProfile.GroupAdminLink Property

Gets or Sets the url for the group admin links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The group admin link. {0} is the
group admin ID and {1} is the group admin displayname.") ]
public string GroupAdminLink;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The group admin link. {0} is the
group admin ID and {1} is the group admin displayname.") >
Public Property GroupAdminLink() As string
```

### CommunityGroupProfile.GroupAdminLinkTarget Property

Gets or Sets the target attribute for the group admin links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group admin links.") ]
public ItemLinkTargets GroupAdminLinkTarget;
```

**Visual Basic**

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

### CommunityGroupProfile.Stylesheet Property

Specifies the Css to which this control will provide a link.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

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

### CommunityGroupProfile.TagTemplate Property

Gets or Sets the template-url for the group tag links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") ]
```

```
public string TagTemplate;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a
community search, when a tag is clicked (example: CommunitySearch.aspx).") >
Public Property TagTemplate() As string
```

### CommunityGroupProfile.TagTemplateTarget Property

Gets or Sets the target (window) attribute for the group-tag links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") ]
public ItemLinkTargets TagTemplateTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the group tag links (_blank, _self, _parent or _top), to select opening in
the same window, a popup, etc.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

### CommunityGroupProfile.Text Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to treat the server control as a Label, taking information
from the Control and painting it on top of the Control for display. Leave blank for default
display.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public 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
```

### CommunityGroupProfile.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

# CommunitySearch Class

### Class Hierarchy

### C#

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

### Visual Basic

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

### File

CommunitySearch.cs

### Description

This is class Ektron.Cms.Controls.CommunitySearch.

## CommunitySearch Methods

The methods of the CommunitySearch class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 127) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 127) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 128) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## CommunitySearch.Fill Method

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

### C#

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

### Visual Basic

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

## CommunitySearch.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

### C#

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

### Visual Basic

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

## CommunitySearch.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

## CommunitySearch Properties

The properties of the CommunitySearch class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 129) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300. |
| | CustomPropertyID (see page 130) | ID (or comma seperated IDs) of the Custom Property to render in the information box. |
| | DisplayXslt (see page 130) | The Xslt to use to render the Control (optional) |
| | EnableFilteringResults (see page 131) | If true, matching taxonomy categories are shown, allowing the user to narrow the resuls. |
| | EnableGroupResults (see page 131) | If false, groups are prevented from being returned in the results. |
| | EnableMap (see page 131) | Set to false to hide the map functionality in this control. This hides the Location tab when the control is displayed on a Web site and the map properties in Visual Studio. |
| | EnableMouseOutHidesInfo (see page 132) | When true, mousing away from a users name hides the pop-up info box. |
| | EnableMouseOver (see page 132) | When set to False, the profile pop-up is disabled when hovering over a display name in the search results.<br>• True = Enable the profile pop-up when hovering over a display name.<br>• False = Disable the profile pop-up when hovering over a display name. |
| | EnableUnauthorizedSearching (see page 132) | When set to True, a user does not need to be logged in to perform a user search. When set to False, a user must logged in to perform a user search, otherwise the control displays a message stating that log-in is required.<br>• **True** = a user does not need to be logged in to perform a user search.<br>• **False** = a user must be logged in to perform a user search. |
| | EnableUserResults (see page 133) | If false, users are prevented from being returned in the results. |
| | FriendsOnly (see page 133) | If true, only returns friends. |
| | GroupTaxonomyID (see page 133) | The taxonomy ID to use for the group-categories (if zero, then none shown). This category must be created through the taxonomy wizard. |
| | InfoBoxShowsEmail (see page 134) | Set true to include users email address values in the pop-up information box (visible by mousing over a users name in the search results). |
| | LoadingImage (see page 134) | Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**. |
| | MapControl (see page 134) | This property allows you to set properties for the Location Tab's map. These properties are identical to the properties in the Map (see page 340) server control. The EnableMap (see page 131) property must be set to true to see the sub properties. |

| | | |
|---|---|---|
| | MarkupLanguage (see page 135) | Enter the name of an EkML file that manages the display of this server control. |
| | MaxTagCount (see page 135) | The maximum number of personal tags to show for each user in the search results. |
| | MembersOnly (see page 135) | When set to True, this server control only returns Membership (see page 350) Users. When set to False, the control returns CMS400.NET Users and Membership (see page 350) Users. |
| | PageSize (see page 136) | The maximum number of user results to show per page. |
| | ResultProfile (see page 136) | Determines whether profile information is displayed. This is based on the state of the user performing the user search. Choices are:<br>• **full** - always show user profile information.<br>• **nonprivate** - display user profile information only when a user's profile is not marked private.<br>• **nonprivate-authenticated** - display user profile information only when a user's profile is not marked private and the user performing the search is logged in.<br>• **none** - never show user profile information. |
| | SearchPrivate (see page 136) | When set to False, this property prevents users with profiles marked as private from being returned with user search results. |
| | ShowNameEmail (see page 137) | When set to True, populates the drop down list with these additional options:<br>• **First Name** - perform searches based on a user's first name.<br>• **Last Name** - perform searches based on a user's last name.<br>• **Email** - perform searches based on a user's email address.<br>When set to False, the drop down list is populated with these properties:<br>• **Tags (see page 771)** - perform searches based on a user's tags.<br>• **Display Name** - perform searches based on a user's display name. |
| | StartingTab (see page 137) | Default tab that's selected when the server control is first rendered on the Web site. Choose from **basic**, **advanced** or **location**. |
| | Stylesheet (see page 138) | Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | TemplateGroupProfile (see page 138) | The URL path of the community group template. This path can be relative or absolute. |
| | TemplateParamName (see page 138) | Gets or sets a Querystring parameter for the ID of users or groups that are returned in the search results. This parameter is passed to the template page defined in the TemplateUserProfile (see page 139) or TemplateGroupProfile (see page 138) property when a user clicks the Location (map) icon associated with a search result. |
| | TemplateTarget (see page 139) | Determines the type of window that appears on this Web form when a user clicks an a user name or tag. The default is _self.<br>• _**self** = opens in same window<br>• _**top** = opens in parent window<br>• _**blank** = opens in new window<br>• _**parent** = opens in the parent frame |
| | TemplateUserProfile (see page 139) | The URL path of the user profile template. This path can be relative or absolute. |
| | UseMsAjax (see page 139) | When true, uses Microsoft Ajax. |
| | UserTaxonomyID (see page 140) | The taxonomy ID to use for the user-categories (if zero, then none shown). This category must be created through the taxonomy wizard. |
| | XmlDoc (see page 140) | Returns the Xml object for the control. |
| | XsltDoc (see page 140) | Returns the Xsl object for the control. |

### CommunitySearch.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### CommunitySearch.CustomPropertyID Property

ID (or comma seperated IDs) of the Custom Property to render in the information box.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Specifies the custom properties to include in the users information pop-up box,
after the first and last names, in the search results (use a comma-delimiter for more than one
ID)")]
[DefaultValue("")]
public string CustomPropertyID;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Specifies the custom properties to include in the users information pop-up box,
after the first and last names, in the search results (use a comma-delimiter for more than one
ID)")>
<DefaultValue("")>
Public Property CustomPropertyID() As string
```

### CommunitySearch.DisplayXslt Property

The Xslt to use to render the Control (optional)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltCommunitySearchTypeConverter))]
[DefaultValue("CommunitySearchDirectory.xsl")]
public override string DisplayXslt;
```

**Visual Basic**

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

### CommunitySearch.EnableFilteringResults Property

If true, matching taxonomy categories are shown, allowing the user to narrow the resuls.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to true to enable showing any matching taxonomy categories, which the user
can then use to reduce the results to suit their interests")]
[DefaultValue("true")]
public bool EnableFilteringResults;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to true to enable showing any matching taxonomy categories, which the user
can then use to reduce the results to suit their interests")>
<DefaultValue("true")>
Public Property EnableFilteringResults() As bool
```

### CommunitySearch.EnableGroupResults Property

If false, groups are prevented from being returned in the results.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to true to enable groups being included in the results (default), false
prevents groups from being returned in the results")]
[DefaultValue("true")]
public bool EnableGroupResults;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to true to enable groups being included in the results (default), false
prevents groups from being returned in the results")>
<DefaultValue("true")>
Public Property EnableGroupResults() As bool
```

### CommunitySearch.EnableMap Property

Set to false to hide the map functionality in this control. This hides the Location tab when the control is displayed on a Web site and the map properties in Visual Studio.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to false to hide the map functionality in this control. This hides the
Location tab when the control is displayed on a Web site and the map properties in Visual
Studio.")]
[DefaultValue("true")]
public bool EnableMap;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to false to hide the map functionality in this control. This hides the
Location tab when the control is displayed on a Web site and the map properties in Visual
Studio.")>
<DefaultValue("true")>
Public Property EnableMap() As bool
```

### CommunitySearch.EnableMouseOutHidesInfo Property

When true, mousing away from a users name hides the pop-up info box.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, then the the pop-up information box is hidden when the mouse moves away
from a users display name in the search results. If false, then the close button must be
clicked.")]
[DefaultValue("true")]
public bool EnableMouseOutHidesInfo;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, then the the pop-up information box is hidden when the mouse moves away
from a users display name in the search results. If false, then the close button must be
clicked.")>
<DefaultValue("true")>
Public Property EnableMouseOutHidesInfo() As bool
```

### CommunitySearch.EnableMouseOver Property

When set to False, the profile pop-up is disabled when hovering over a display name in the search results.

- True = Enable the profile pop-up when hovering over a display name.
- False = Disable the profile pop-up when hovering over a display name.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When the property is set to false, prevents pop-up profile from ever being
shown.")]
[DefaultValue("true")]
public bool EnableMouseOver;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When the property is set to false, prevents pop-up profile from ever being
shown.")>
<DefaultValue("true")>
Public Property EnableMouseOver() As bool
```

### CommunitySearch.EnableUnauthorizedSearching Property

When set to True, a user does not need to be logged in to perform a user search. When set to False, a user must logged in to perform a user search, otherwise the control displays a message stating that log-in is required.

- **True** = a user does not need to be logged in to perform a user search.
- **False** = a user must be logged in to perform a user search.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description(" When the property is set to false and user is not logged in, UI displays
message that log-in is required (searching is prevented).")]
[DefaultValue("true")]
public bool EnableUnauthorizedSearching;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description(" When the property is set to false and user is not logged in, UI displays
message that log-in is required (searching is prevented).")>
<DefaultValue("true")>
Public Property EnableUnauthorizedSearching() As bool
```

### CommunitySearch.EnableUserResults Property

If false, users are prevented from being returned in the results.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to true to enable users being included in the results (default), false
prevents users from being returned in the results")]
[DefaultValue("true")]
public bool EnableUserResults;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to true to enable users being included in the results (default), false
prevents users from being returned in the results")>
<DefaultValue("true")>
Public Property EnableUserResults() As bool
```

### CommunitySearch.FriendsOnly Property

If true, only returns friends.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Set to true to limit results to the users friends, otherwise returns all matches
(default)")]
[DefaultValue("false")]
public bool FriendsOnly;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Set to true to limit results to the users friends, otherwise returns all matches
(default)")>
<DefaultValue("false")>
Public Property FriendsOnly() As bool
```

### CommunitySearch.GroupTaxonomyID Property

The taxonomy ID to use for the group-categories (if zero, then none shown). This category must be created through the taxonomy wizard.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("The taxonomy ID to use for the group-categories (if zero, then none shown). This
category must be created through the taxonomy wizard.")]
[DefaultValue("0")]
public long GroupTaxonomyID;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The taxonomy ID to use for the group-categories (if zero, then none shown). This
category must be created through the taxonomy wizard.")>
<DefaultValue("0")>
Public Property GroupTaxonomyID() As long
```

### CommunitySearch.InfoBoxShowsEmail Property

Set true to include users email address values in the pop-up information box (visible by mousing over a users name in the search results).

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, then the the users email address will be rendered in the information
box (if enabled and the viewer has the required authority).")]
[DefaultValue("false")]
public bool InfoBoxShowsEmail;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, then the the users email address will be rendered in the information
box (if enabled and the viewer has the required authority).")>
<DefaultValue("false")>
Public Property InfoBoxShowsEmail() As bool
```

### CommunitySearch.LoadingImage Property

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

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

### CommunitySearch.MapControl Property

This property allows you to set properties for the Location Tab's map. These properties are identical to the properties in the Map (see page 340) server control. The EnableMap (see page 131) property must be set to true to see the sub properties.

**C#**

```
[Browsable(false), Category("Ektron")]
[TypeConverter(typeof(Ektron.Cms.Controls.Map))]
[Description("This property allows you to set properties for the Location Tab?s map. These
properties are identical to the properties in the Map server control. The EnableMap property
must be set to true to see the sub properties.")]
public Ektron.Cms.Controls.Map MapControl;
```

**Visual Basic**

```
<Browsable(false), Category("Ektron")>
<TypeConverter(typeof(Ektron.Cms.Controls.Map))>
<Description("This property allows you to set properties for the Location Tab?s map. These
properties are identical to the properties in the Map server control. The EnableMap property
must be set to true to see the sub properties.")>
Public Property MapControl() As Ektron.Cms.Controls.Map
```

**Description**

The map control properties can be changed only through code behind and not from designer property grid

### CommunitySearch.MarkupLanguage Property

Enter the name of an EkML file that manages the display of this server control.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Enter the name of an EkML file that manages the display of this server
control.")]
[DefaultValue("")]
public string MarkupLanguage;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Enter the name of an EkML file that manages the display of this server
control.")>
<DefaultValue("")>
Public Property MarkupLanguage() As string
```

### CommunitySearch.MaxTagCount Property

The maximum number of personal tags to show for each user in the search results.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("The maximum number of personal tags to show for each user in the search
results.")]
[DefaultValue("5")]
public int MaxTagCount;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The maximum number of personal tags to show for each user in the search
results.")>
<DefaultValue("5")>
Public Property MaxTagCount() As Integer
```

### CommunitySearch.MembersOnly Property

When set to True, this server control only returns Membership (see page 350) Users. When set to False, the control returns CMS400.NET Users and Membership (see page 350) Users.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Allows limiting results for security; When true, search only returns members.
When false, returns members and CMS uers.")]
```

```
[DefaultValue("false")]
public bool MembersOnly;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Allows limiting results for security; When true, search only returns members.
When false, returns members and CMS uers.")>
<DefaultValue("false")>
Public Property MembersOnly() As bool
```

### CommunitySearch.PageSize Property

The maximum number of user results to show per page.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("The maximum number of user results to show per page.")]
[DefaultValue("10")]
public int PageSize;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The maximum number of user results to show per page.")>
<DefaultValue("10")>
Public Property PageSize() As Integer
```

### CommunitySearch.ResultProfile Property

Determines whether profile information is displayed. This is based on the state of the user performing the user search. Choices are:

- **full** - always show user profile information.
- **nonprivate** - display user profile information only when a user's profile is not marked private.
- **nonprivate-authenticated** - display user profile information only when a user's profile is not marked private and the user performing the search is logged in.
- **none** - never show user profile information.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(ResultProfileCommunitySearchTypeConverter))]
[Description(" This property will display the information of users or friends based on the
state of the user who is seeking the information.")]
[DefaultValue("nonprivate-authenticated")]
public string ResultProfile;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(ResultProfileCommunitySearchTypeConverter))>
<Description(" This property will display the information of users or friends based on the
state of the user who is seeking the information.")>
<DefaultValue("nonprivate-authenticated")>
Public Property ResultProfile() As string
```

### CommunitySearch.SearchPrivate Property

When set to False, this property prevents users with profiles marked as private from being returned with user search results.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When set to False, this property prevents users with profiles marked as private
from being returned with user search results.")]
[DefaultValue("true")]
public bool SearchPrivate;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When set to False, this property prevents users with profiles marked as private
from being returned with user search results.")>
<DefaultValue("true")>
Public Property SearchPrivate() As bool
```

### CommunitySearch.ShowNameEmail Property

When set to True, populates the drop down list with these additional options:

- **First Name** - perform searches based on a user's first name.
- **Last Name** - perform searches based on a user's last name.
- **Email** - perform searches based on a user's email address.

When set to False, the drop down list is populated with these properties:

- **Tags (see page 771)** - perform searches based on a user's tags.
- **Display Name** - perform searches based on a user's display name.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When the property is set to true, populates the drop-down list with additional
options; First Name, Last Name, email.")]
[DefaultValue("true")]
public bool ShowNameEmail;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When the property is set to true, populates the drop-down list with additional
options; First Name, Last Name, email.")>
<DefaultValue("true")>
Public Property ShowNameEmail() As bool
```

### CommunitySearch.StartingTab Property

Default tab that's selected when the server control is first rendered on the Web site. Choose from **basic**, **advanced** or **location**.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The tab to show as selected, when the control is first rendered: choose from
'basic', 'advanced' or 'location'.")]
[DefaultValue("basic")]
public string StartingTab;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The tab to show as selected, when the control is first rendered: choose from
'basic', 'advanced' or 'location'.")>
```

```
<DefaultValue("basic")>
Public Property StartingTab() As string
```

### CommunitySearch.Stylesheet Property

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

**C#**

```
[Category("Ektron")]
[Description("Allows selecting a different CSS file for the community search server control.")]
[TypeConverter(typeof(string))]
[DefaultValue("csslib/CommunitySearch.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Allows selecting a different CSS file for the community search server control.")>
<TypeConverter(typeof(string))>
<DefaultValue("csslib/CommunitySearch.css")>
Public Property Stylesheet() As string
```

### CommunitySearch.TemplateGroupProfile Property

The URL path of the community group template. This path can be relative or absolute.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL path of the community group template. This path can be relative or
absolute.")]
[DefaultValue("")]
public string TemplateGroupProfile;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL path of the community group template. This path can be relative or
absolute.")>
<DefaultValue("")>
Public Property TemplateGroupProfile() As string
```

### CommunitySearch.TemplateParamName Property

Gets or sets a Querystring parameter for the ID of users or groups that are returned in the search results. This parameter is passed to the template page defined in the TemplateUserProfile (see page 139) or TemplateGroupProfile (see page 138) property when a user clicks the Location (map) icon associated with a search result.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Gets or sets a Querystring parameter for the ID of users or groups that are
returned in the search results. This parameter is passed to the template page defined in the
TemplateUserProfile or TemplateGroupProfile property when a user clicks the Location (map)
icon associated with a search result.")]
[DefaultValue("id")]
public string TemplateParamName;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Gets or sets a Querystring parameter for the ID of users or groups that are
returned in the search results. This parameter is passed to the template page defined in the
```

```
TemplateUserProfile or TemplateGroupProfile property when a user clicks the Location (map)
icon associated with a search result.")>
<DefaultValue("id")>
Public Property TemplateParamName() As string
```

### CommunitySearch.TemplateTarget Property

Determines the type of window that appears on this Web form when a user clicks an a user name or tag. The default is _self.

- **_self** = opens in same window
- **_top** = opens in parent window
- **_blank** = opens in new window
- **_parent** = opens in the parent frame

**C#**

```
[Category("Ektron")]
[DefaultValue(ItemLinkTargets._self)]
[Description("The target attribute of the personal-tag and profile links (_blank, _self,
_parent or _top), to select opening in the same window, a popup, etc.")]
public ItemLinkTargets TemplateTarget;
```

**Visual Basic**

```
<Category("Ektron")>
<DefaultValue(ItemLinkTargets._self)>
<Description("The target attribute of the personal-tag and profile links (_blank, _self,
_parent or _top), to select opening in the same window, a popup, etc.")>
Public Property TemplateTarget() As ItemLinkTargets
```

### CommunitySearch.TemplateUserProfile Property

The URL path of the user profile template. This path can be relative or absolute.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL path of the user profile template. This path can be relative or
absolute.")]
[DefaultValue("")]
public string TemplateUserProfile;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL path of the user profile template. This path can be relative or
absolute.")>
<DefaultValue("")>
Public Property TemplateUserProfile() As string
```

### CommunitySearch.UseMsAjax Property

When true, uses Microsoft Ajax.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")]
[DefaultValue("false")]
public bool UseMsAjax;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")>
<DefaultValue("false")>
Public Property UseMsAjax() As bool
```

### CommunitySearch.UserTaxonomyID Property

The taxonomy ID to use for the user-categories (if zero, then none shown). This category must be created through the taxonomy wizard.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("The taxonomy ID to use for the user-categories (if zero, then none shown). This
category must be created through the taxonomy wizard.")]
[DefaultValue("0")]
public long UserTaxonomyID;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The taxonomy ID to use for the user-categories (if zero, then none shown). This
category must be created through the taxonomy wizard.")>
<DefaultValue("0")>
Public Property UserTaxonomyID() As long
```

### CommunitySearch.XmlDoc Property

Returns the Xml object for the control.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

### CommunitySearch.XsltDoc Property

Returns the Xsl object for the control.

**C#**

```
public override XmlDocument XsltDoc;
```

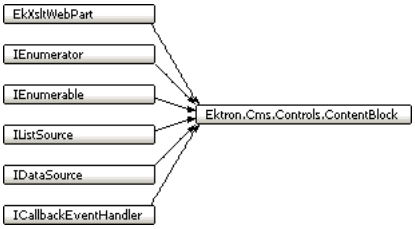
**Visual Basic**

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

## ContentBlock Class

Summary description for ContentBlock.

### Class Hierarchy




**C#**

```
[DefaultProperty("DefaultContentID"), ToolboxData("<{0}:ContentBlock
runat=server></{0}:ContentBlock>")]
public class ContentBlock : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

**Visual Basic**

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

**File**

ContentBlock.cs

## ContentBlock Methods

The methods of the ContentBlock class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | CopyTo (see page 141) | 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 142) | Method that can be overridden and used to grab data from the back end for each server control. |
| | GetEnumerator (see page 142) | Gets a System.Collections.IEnumerator for the collection. |
| | MoveNext (see page 142) | Moves up the index of the Collection (see page 88). Returns false if the end of the collection has been surpassed. |
| | Reset (see page 142) | Brings the index of the collection back to zero. Does not clear the collection. |
| | this (see page 142) | Index operator into a strongly typed dataset holding some common properties for a Collection (see page 88) item. |
| | ToString (see page 142) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 143) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## ContentBlock.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

**C#**

```
public void CopyTo(Array array, int i);
```

**Visual Basic**

```
Public Function CopyTo(array As Array, i As Integer) As void
```

### ContentBlock.Fill Method

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

**C#**

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

**Visual Basic**

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

### ContentBlock.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

**C#**

```
public IEnumerator GetEnumerator();
```

**Visual Basic**

```
Public Function GetEnumerator() As IEnumerator
```

### ContentBlock.MoveNext Method

Moves up the index of the Collection (see page 88). 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 88) 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 |
|---|---|---|
| | CacheInterval (see page 143) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | Count (see page 144) | Indicates the number of elements in your Collection (see page 88). |
| | Current (see page 144) | Gets the current object in the collection. |
| | DefaultContentID (see page 144) | Gets or Sets the Default ContentID for this Control. |
| | DisplayXslt (see page 145) | Gets or Sets the path to an Xslt that can be applied to this content block. Set to String.Empty if you wish to always display the default Content (see page 788) view. |
| | DynamicParameter (see page 145) | Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.) |
| | EkItem (see page 145) | Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 788) Block. |
| | IsSynchronized (see page 146) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | OverrideXslt (see page 146) | Ektron CMS400.NET allows an Administrator to specify up to three custom Xslt's which can be applied to a content block from the System. This allows the user to override the default ContentBlock (see page 140) Xslt and display one of these custom Xsl transforms, if specified in the system. In addition, you can use the Xslt property to perform a transform on a content block (see Xslt property.) |
| | StartingFolderID (see page 146) | For use with Web Parts only. Allows a developer to set the starting folder for the personalization/collaboration selection dialog. |
| | SyncRoot (see page 147) | Gets an object that can be used to synchronize the collection. |

### ContentBlock.CacheInterval Property

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

**C#**

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

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

**Visual Basic**

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

### ContentBlock.Count Property

Indicates the number of elements in your Collection (see page 88).

**C#**

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

**Visual Basic**

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

### ContentBlock.Current Property

Gets the current object in the collection.

**C#**

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

**Visual Basic**

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

### ContentBlock.DefaultContentID Property

Gets or Sets the Default ContentID for this Control.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control.")]
[Personalizable(), WebBrowsable()]
public long DefaultContentID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control.")>
```

```
<Personalizable(), WebBrowsable()>
Public Property DefaultContentID() As long
```

### ContentBlock.DisplayXslt Property

Gets or Sets the path to an Xslt that can be applied to this content block. Set to String.Empty if you wish to always display the default Content (see page 788) view.

**C#**

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

**Visual Basic**

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

### ContentBlock.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)

**C#**

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

**Visual Basic**

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

### ContentBlock.EkItem Property

Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 788) Block.

**C#**

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItem;
```

**Visual Basic**

```
<Browsable(false)>
```

```
Public Property EkItem() As Ektron.Cms.Common.ContentBase
```

### ContentBlock.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

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

**Visual Basic**

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

### ContentBlock.OverrideXslt Property

Ektron CMS400.NET allows an Administrator to specify up to three custom Xslt's which can be applied to a content block from the System. This allows the user to override the default ContentBlock (see page 140) Xslt and display one of these custom Xsl transforms, if specified in the system. In addition, you can use the Xslt property to perform a transform on a content block (see Xslt property.)

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[DefaultValue(XsltOverride.Default)]
[Description("Use this to specify an XSLT that will override the default display for an XML
content block.")]
[Obsolete("Specify the display XSLT path in \"DisplayXslt\" property.")]
public XsltOverride OverrideXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<DefaultValue(XsltOverride.Default)>
<Description("Use this to specify an XSLT that will override the default display for an XML
content block.")>
<Obsolete("Specify the display XSLT path in \"DisplayXslt\" property.")>
Public Property OverrideXslt() As XsltOverride
```

### ContentBlock.StartingFolderID Property

For use with Web Parts only. Allows a developer to set the starting folder for the personalization/collaboration selection dialog.

**C#**

```
[Bindable(true)]
[Category("Ektron WebParts")]
[Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")]
[DefaultValue(0)]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long StartingFolderID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron WebParts")>
<Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")>
<DefaultValue(0)>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
```

```
Public Property StartingFolderID() As long
```

### ContentBlock.SyncRoot Property

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

**C#**

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

**Visual Basic**

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

## ContentFlagging Class

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

ContentFlagging.cs

### ContentFlagging Methods

The methods of the ContentFlagging class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 147) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 147) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 148) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### ContentFlagging.Fill Method

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

**C#**

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

**Visual Basic**

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

### ContentFlagging.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

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

**Visual Basic**

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

### ContentFlagging.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

### ContentFlagging Properties

The properties of the ContentFlagging class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DefaultContentID (see page 148) | Gets or Sets the default content ID for this control. This is the content ID that will be flagged if the DynamicParameter (see page 149) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 149) of "id", and will display DefaultContentID otherwise.) |
| | DisplayXslt (see page 149) | Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. |
| | DynamicParameter (see page 149) | Gets or sets the Querystring parameter to read a content ID dynamically. To use the default content ID, leave blank. |
| | FlagImage (see page 149) | Determines the image that will be shown to allow flagging. This overrides the FlagText (see page 150) value. |
| | FlagText (see page 150) | Determines the text that will be shown to allow flagging. FlagImage (see page 149) overrides this value. |
| | Stylesheet (see page 150) | Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Select **Use Default** to use the default style sheet. |
| | XmlDoc (see page 151) | Returns the Xslt object for the collection. |

### ContentFlagging.DefaultContentID Property

Gets or Sets the default content ID for this control. This is the content ID that will be flagged if the DynamicParameter (see page 149) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 149) of "id", and will display DefaultContentID otherwise.)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentFlaggingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or sets the default content ID for this control. This is the content ID
that will be flagged if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will
get the value 2 for a DynamicParameter of \"id\", and will use DefaultContentID otherwise.)")]
public long DefaultContentID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentFlaggingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or sets the default content ID for this control. This is the content ID
that will be flagged if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will
get the value 2 for a DynamicParameter of \"id\", and will use DefaultContentID otherwise.)")>
Public Property DefaultContentID() As long
```

### ContentFlagging.DisplayXslt Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute.")]
[TypeConverter(typeof(XsltContentFlaggingTypeConverter))]
[DefaultValue("Dropdown with Comments")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute.")>
<TypeConverter(typeof(XsltContentFlaggingTypeConverter))>
<DefaultValue("Dropdown with Comments")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### ContentFlagging.DynamicParameter Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or sets the Querystring parameter to read a content ID dynamically. To use
the default content ID, leave blank.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or sets the Querystring parameter to read a content ID dynamically. To use
the default content ID, leave blank.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

### ContentFlagging.FlagImage Property

Determines the image that will be shown to allow flagging. This overrides the FlagText (see page 150) value.

**C#**

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

```
[Description("Determines the image that will be shown to allow flagging. This overrides the
FlagText value.")]
[DefaultValue("")]
[Personalizable()]
public string FlagImage;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Determines the image that will be shown to allow flagging. This overrides the
FlagText value.")>
<DefaultValue("")>
<Personalizable()>
Public Property FlagImage() As string
```

### ContentFlagging.FlagText Property

Determines the text that will be shown to allow flagging. FlagImage (see page 149) overrides this value.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Determines the text that will be shown to allow flagging. FlagImage overrides
this value.")]
[DefaultValue("Flag")]
[Personalizable()]
public string FlagText;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Determines the text that will be shown to allow flagging. FlagImage overrides
this value.")>
<DefaultValue("Flag")>
<Personalizable()>
Public Property FlagText() As string
```

### ContentFlagging.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Select **Use Default** to use the default style sheet.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Select Use Default to use the default style sheet.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Select Use Default to use the default style sheet.")>
<Personalizable()>
Public Property Stylesheet() As string
```

### ContentFlagging.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

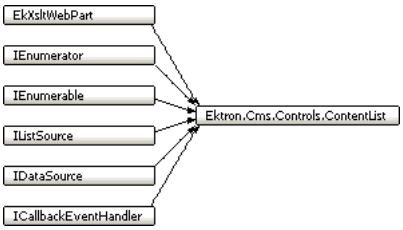
**Visual Basic**

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

## ContentList Class

The ContentList server control is used to display a list of content blocks on a Web page. In contrast to a List Summary, where content must be in a specified folder, the ContentList server control uses content from any CMS400.NET folder.

**Class Hierarchy**




**C#**

```
[DefaultProperty("ContentIds"), ToolboxData("<{0}:ContentList
runat=server></{0}:ContentList>")]
public class ContentList : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

**Visual Basic**

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

**File**

ContentList.cs

### ContentList Methods

The methods of the ContentList class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | CopyTo (see page 152) | 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 152) | Populates the data for the ContentList (see page 151). |
| | GetEnumerator (see page 152) | Gets a System.Collections.IEnumerator for the ContentList (see page 151). |
| | GetRssFeed (see page 152) | This method returns the ContentList (see page 151) RSS feed. |
| | MoveNext (see page 152) | Move to next index. |

| | | |
|---|---|---|
| | Reset (see page 153) | Resets the index. |
| | this (see page 153) | Returns the current ContentItem.ContentRow. |
| | ToString (see page 153) | Overrides the system ToString function. |
| | Wrap (see page 153) | Wrap the Html toolbar. |

### ContentList.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

**C#**

```
public void CopyTo(Array array, int index);
```

**Visual Basic**

```
Public Function CopyTo(array As Array, index As Integer) As void
```

### ContentList.Fill Method

Populates the data for the ContentList (see page 151).

**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 151).

**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 151) 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 788) 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 155) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | ContentIds (see page 155) | The comma separated list of content IDs you wish to display. |
| | Count (see page 156) | Indicates the number of elements in your ContentList (see page 151). |
| | Current (see page 156) | Gets the current index. |
| | DefaultContentID (see page 156) | Gets or Sets the Default ContentID for this Control. |
| | Direction (see page 156) | Direction for sorting results, "ascending" or "descending". |
| | DynamicParameter (see page 157) | Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.) |
| | EkItems (see page 157) | Returns the array of ContentBase. |
| | GetHtml (see page 157) | If set to true, retrieves the content (Html body) of every content block in the content list for display. |
| | IncludeIcons (see page 158) | Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser. |
| | IsSynchronized (see page 158) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | LinkTarget (see page 158) | Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.<br>• **_Self** = opens in same window<br>• **_Top** = opens in parent window<br>• **_Blank** = opens in new window<br>• **_Parent** = opens in the parent frame |
| | MarkupLanguage (see page 159) | Enter name of a markup file that manages the display of this server control. |
| | MetaTag (see page 159) | Gets or Sets the meta tag value. Once this value is set, associated content(s) will be listed. |
| | OrderByDirection (see page 159) | Determines how to order content in the list. The sort field is determined by the OrderKey (see page 160) property.<br>• ascending - items are arranged A, B, C or 1,2,3.<br>• descending - items are arranged. Z,Y,X or 3,2,1.<br>If sorting by date, descending puts the most recent first.<br>When ascending is selected and the Orderkey property is set to OrderOfTheIds, the order of the IDs are preserved. When set to descending, the order is reversed. |

| | | |
|---|---|---|
| | OrderKey (see page 160) | Sort the list by one of these values:<br>• **Title** - the order of the content's title.<br>• **DateModified** - the last date the content was modified.<br>• **DateCreated** - the date the content was created.<br>• **LastEditorFname** - the last editor's first name.<br>• **LastEditorLname** - the last editor's last name.<br>• **OrderOfTheIds** - preserves the content ID order based on the list in the ContentIds (see page 155) property. |
| | Random (see page 160) | Set to True if you want to randomly display one content block link from the content list. The content changes each time a user views the page.<br>• **True** - randomly display one content block in the specified folder.<br>• **False** - display the content list normally. |
| | SuppressWrapperTags (see page 160) | Suppresses the output of span/div tags around the control. |
| | SyncRoot (see page 161) | Gets an object that can be used to synchronize the ContentList (see page 151). |
| | XmlDoc (see page 161) | Returns the Xml object loaded for ContentList (see page 151). |

### ContentList.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### ContentList.ContentIds Property

The comma separated list of content IDs you wish to display.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The comma separated list of content IDs you wish to display.")]
[Personalizable(), WebBrowsable()]
public string ContentIds;
```

**Visual Basic**

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

### ContentList.Count Property

Indicates the number of elements in your ContentList (see page 151).

**C#**

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

**Visual Basic**

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

### ContentList.Current Property

Gets the current index.

**C#**

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

**Visual Basic**

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

### ContentList.DefaultContentID Property

Gets or Sets the Default ContentID for this Control.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control.")]
[Personalizable(), WebBrowsable()]
public long DefaultContentID;
```

**Visual Basic**

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

### ContentList.Direction Property

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

**C#**

```
[Obsolete("Use OrderByDirection Property")]
[Browsable(false)]
[Bindable(true)]
[Category("Ektron")]
[Description("The direction (ascending or descending) that you wish your sorted results to be
```

```
displayed.")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")]
public new Ektron.Cms.Controls.CmsWebService.OrderByDirection Direction;
```

**Visual Basic**

```
<Obsolete("Use OrderByDirection Property")>
<Browsable(false)>
<Bindable(true)>
<Category("Ektron")>
<Description("The direction (ascending or descending) that you wish your sorted results to be
displayed.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")>
Public new Property Direction() As Ektron.Cms.Controls.CmsWebService.OrderByDirection
```

### ContentList.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)

**C#**

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

**Visual Basic**

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

### ContentList.EkItems Property

Returns the array of ContentBase.

**C#**

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItems;
```

**Visual Basic**

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

### ContentList.GetHtml Property

If set to true, retrieves the content (Html body) of every content block in the content list for display.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, retrieves the content (Html body) of every content block in the
content list for display.")]
public bool GetHtml;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, retrieves the content (Html body) of every content block in the
content list for display.")>
Public Property GetHtml() As bool
```

### ContentList.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.

**C#**

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: Only works with ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
public bool IncludeIcons;
```

**Visual Basic**

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: Only works with ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
Public Property IncludeIcons() As bool
```

### ContentList.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

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

**Visual Basic**

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

### ContentList.LinkTarget Property

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

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

**C#**

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

**Visual Basic**

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

```
Public Property LinkTarget() As ItemLinkTargets
```

### ContentList.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

### ContentList.MetaTag Property

Gets or Sets the meta tag value. Once this value is set, associated content(s) will be listed.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Gets or Sets the meta tag value.  Once this value is set, associated content(s)
will be listed.")]
public string MetaTag;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Gets or Sets the meta tag value.  Once this value is set, associated content(s)
will be listed.")>
Public Property MetaTag() As string
```

### ContentList.OrderByDirection Property

Determines how to order content in the list. The sort field is determined by the OrderKey (see page 160) property.

- ascending - items are arranged A, B, C or 1,2,3.
- descending - items are arranged. Z,Y,X or 3,2,1.

If sorting by date, descending puts the most recent first.

When ascending is selected and the Orderkey property is set to OrderOfTheIds, the order of the IDs are preserved. When set to descending, the order is reversed.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(EkEnumeration.OrderByDirection.Ascending)]
[Description("Direction for sorting results, 'ascending' or 'descending'.")]
public Ektron.Cms.Common.EkEnumeration.OrderByDirection OrderByDirection;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(EkEnumeration.OrderByDirection.Ascending)>
<Description("Direction for sorting results, 'ascending' or 'descending'.")>
Public Property OrderByDirection() As Ektron.Cms.Common.EkEnumeration.OrderByDirection
```

### ContentList.OrderKey Property

Sort the list by one of these values:

- **Title** - the order of the content's title.
- **DateModified** - the last date the content was modified.
- **DateCreated** - the date the content was created.
- **LastEditorFname** - the last editor's first name.
- **LastEditorLname** - the last editor's last name.
- **OrderOfTheIds** - preserves the content ID order based on the list in the ContentIds (see page 155) property.

**C#**

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Controls.CmsWebService.ContentListOrderBy.OrderOfTheIds)]
[Description("Sort the list by one of these values: Title, DateModified, DateCreated,
LastEditorFname, LastEditorLname or OrderOfTheIds.")]
[Category("Ektron")]
public Ektron.Cms.Controls.CmsWebService.ContentListOrderBy OrderKey;
```

**Visual Basic**

```
<Bindable(true)>
<DefaultValue(Ektron.Cms.Controls.CmsWebService.ContentListOrderBy.OrderOfTheIds)>
<Description("Sort the list by one of these values: Title, DateModified, DateCreated,
LastEditorFname, LastEditorLname or OrderOfTheIds.")>
<Category("Ektron")>
Public Property OrderKey() As Ektron.Cms.Controls.CmsWebService.ContentListOrderBy
```

### ContentList.Random Property

Set to True if you want to randomly display one content block link from the content list. The content changes each time a user views the page.

- **True** - randomly display one content block in the specified folder.
- **False** - display the content list normally.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Set this property to True to grab a single, random piece of content from the
content list.")]
public bool Random;
```

**Visual Basic**

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

### ContentList.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

**C#**

```
public override bool SuppressWrapperTags;
```

**Visual Basic**

```
Public override Property SuppressWrapperTags() As bool
```

### ContentList.SyncRoot Property

Gets an object that can be used to synchronize the ContentList (see page 151).

**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 151).

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

## ContentRating Class

Use the ContentRating server control to place a rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. A visitor can rank the content on a numerical scale of one through nine, as well as submit a text comment.

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

ContentRating.cs

**Description**

[Obsolete()]

### ContentRating Methods

The methods of the ContentRating class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 162) | Method that can be overridden and used to grab data from the back end for each server control. |

| | | |
|---|---|---|
| | ToString (see page 162) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 162) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## ContentRating.Fill Method

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

**C#**

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

**Visual Basic**

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

## ContentRating.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

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

**Visual Basic**

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

## ContentRating.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

| Parameters | Description |
|---|---|
| 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 163) | The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback". |
| | BadLabel (see page 163) | The label corresponding to a "Bad" content rating. This defaults to "Poor". |
| | CharsRemainingLabel (see page 164) | Sets or gets the label for the number of additional characters a user may enter into the comment. The default is "Characters Remaining:" |
| | CompleteGraphDisplay (see page 164) | Described the layout manner which the graph is displayed to a user after content rating. The graph will be displayed to the right of the dialog in the event of Horizontal, below in the case of Vertical, and not displayed in the case of None. |
| | CookiesRequiredMessage (see page 164) | The message that appears when a user needs to enable cookies on their system. The default is: "We're sorry, but you must enable cookies in your browser to rate this content." |
| | DefaultContentID (see page 165) | Gets or Sets the Default ContentID for this Control. This is the ContentID that will be displayed if the DynamicParameter (see page 165) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 165) of "id", and will display DefaultContentID otherwise.) |

| | | |
|---|---|---|
| | DynamicParameter (see page 165) | Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically. Set to "" (empty) if you wish to always display the default ContentID (static.) |
| | FeedbackHeader (see page 165) | The label corresponding to how the user would rate the content block. Defaults to "How would you rate the usefulness of this content?". |
| | GoodLabel (see page 166) | The label corresponding to a "Good" content rating. This defaults to "Outstanding". |
| | GraphBarColor (see page 166) | The color of the bars in the Content (see page 788) Rating bar graph. |
| | InitialGraphDisplay (see page 166) | Describes the layout manner which the graph is displayed to a user before content rating. The graph will be displayed to the right of the dialog in the event of Horizontal, below in the case of Vertical, and not displayed in the case of None. |
| | Padding (see page 167) | Padding, in pixels, to be applied between the content rating input and display graph. |
| | RatingLevelLabel (see page 167) | The label corresponding to a "Good" content rating. This defaults to "Outstanding". |
| | ThankYouMessage (see page 168) | The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback.". |
| | TotalRatingsLabel (see page 168) | The label that appears below the graph, and lists the number of rating submissions received. The default is: "Total Ratings". |
| | UserCommentsHeader (see page 168) | The label corresponding to why the user rated content the way they did. Defaults to "Tell us why you rated the content this way". |
| | ValidationMessage (see page 169) | The label displayed to a user when they try to submit a content rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.". |

## ContentRating.AlreadyRatedMessage Property

The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback".

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_RATED)]
public string AlreadyRatedMessage;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_RATED)>
Public Property AlreadyRatedMessage() As string
```

## ContentRating.BadLabel Property

The label corresponding to a "Bad" content rating. This defaults to "Poor".

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to a \"Bad\" content rating. This defaults to
\"Poor\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_BAD)]
public string BadLabel;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to a \"Bad\" content rating. This defaults to
\"Poor\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
```

```
<DefaultValue(EkWebControl.RES_CNTRATE_BAD)>
Public Property BadLabel() As string
```

### ContentRating.CharsRemainingLabel Property

Sets or gets the label for the number of additional characters a user may enter into the comment. The default is "Characters Remaining:"

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("Sets or gets the label for the number of additional characters a user may enter
into the comment. The default is Characters Remaining:")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_CHARS)]
public string CharsRemainingLabel;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("Sets or gets the label for the number of additional characters a user may enter
into the comment. The default is Characters Remaining:")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_CHARS)>
Public Property CharsRemainingLabel() As string
```

### ContentRating.CompleteGraphDisplay Property

Described the layout manner which the graph is displayed to a user after content rating. The graph will be displayed to the right of the dialog in the event of Horizontal, below in the case of Vertical, and not displayed in the case of None.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Described the layout manner which the graph is displayed to a user after content
rating. The graph will be displayed to the right of the dialog in the event of Horizontal,
below in the case of Verical, and not displayed in the case of None.")]
[DefaultValue(Ektron.Cms.ContentAPI.RatingGraphDisplayType.Horizontal)]
public Ektron.Cms.ContentAPI.RatingGraphDisplayType CompleteGraphDisplay;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Described the layout manner which the graph is displayed to a user after content
rating. The graph will be displayed to the right of the dialog in the event of Horizontal,
below in the case of Verical, and not displayed in the case of None.")>
<DefaultValue(Ektron.Cms.ContentAPI.RatingGraphDisplayType.Horizontal)>
Public Property CompleteGraphDisplay() As Ektron.Cms.ContentAPI.RatingGraphDisplayType
```

### ContentRating.CookiesRequiredMessage Property

The message that appears when a user needs to enable cookies on their system. The default is: "We're sorry, but you must enable cookies in your browser to rate this content."

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The message that appears when a user needs to enable cookies on their system.
The default is: \"We're sorry, but you must enable cookies in your browser to rate this
content.\"")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_COOK)]
public string CookiesRequiredMessage;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The message that appears when a user needs to enable cookies on their system.
The default is: \"We're sorry, but you must enable cookies in your browser to rate this
content.\"")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_COOK)>
Public Property CookiesRequiredMessage() As string
```

### ContentRating.DefaultContentID Property

Gets or Sets the Default ContentID for this Control. This is the ContentID that will be displayed if the DynamicParameter (see page 165) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 165) of "id", and will display DefaultContentID otherwise.)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentRatingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control. This is the ContentID that
will be displayed if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will get
the value 2 for a DynamicParameter of \"id\", and will display DefaultContentID otherwise.)")]
public long DefaultContentID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentRatingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control. This is the ContentID that
will be displayed if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will get
the value 2 for a DynamicParameter of \"id\", and will display DefaultContentID otherwise.)")>
Public Property DefaultContentID() As long
```

### ContentRating.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically. Set to "" (empty) if you wish to always display the default ContentID (static.)

**C#**

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

**Visual Basic**

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

### ContentRating.FeedbackHeader Property

The label corresponding to how the user would rate the content block. Defaults to "How would you rate the usefulness of this

content?".

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to how the user would rate the content block. Defaults
to \"How would you rate the usefullness of this content?\"")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_FEED)]
public string FeedbackHeader;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to how the user would rate the content block. Defaults
to \"How would you rate the usefullness of this content?\"")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_FEED)>
Public Property FeedbackHeader() As string
```

### ContentRating.GoodLabel Property

The label corresponding to a "Good" content rating. This defaults to "Outstanding".

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to a \"Good\" content rating. This defaults to
\"Ouotstanding\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_GOOD)]
public string GoodLabel;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to a \"Good\" content rating. This defaults to
\"Ouotstanding\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_GOOD)>
Public Property GoodLabel() As string
```

### ContentRating.GraphBarColor Property

The color of the bars in the Content (see page 788) Rating bar graph.

**C#**

```
[Category("Ektron")]
[Bindable(true)]
[DefaultValue(typeof(System.Drawing.Color), "LightBlue")]
[Description("The color of the bars in the Content Rating bar graph.")]
public System.Drawing.Color GraphBarColor;
```

**Visual Basic**

```
<Category("Ektron")>
<Bindable(true)>
<DefaultValue(typeof(System.Drawing.Color), "LightBlue")>
<Description("The color of the bars in the Content Rating bar graph.")>
Public Property GraphBarColor() As System.Drawing.Color
```

### ContentRating.InitialGraphDisplay Property

Describes the layout manner which the graph is displayed to a user before content rating. The graph will be displayed to the right

of the dialog in the event of Horizontal, below in the case of Vertical, and not displayed in the case of None.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Described the layout manner which the graph is displayed to a user before
content rating. The graph will be displayed to the right of the dialog in the event of
Horizontal, below in the case of Verical, and not displayed in the case of None.")]
[DefaultValue(Ektron.Cms.ContentAPI.RatingGraphDisplayType.Horizontal)]
public Ektron.Cms.ContentAPI.RatingGraphDisplayType InitialGraphDisplay;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Described the layout manner which the graph is displayed to a user before
content rating. The graph will be displayed to the right of the dialog in the event of
Horizontal, below in the case of Verical, and not displayed in the case of None.")>
<DefaultValue(Ektron.Cms.ContentAPI.RatingGraphDisplayType.Horizontal)>
Public Property InitialGraphDisplay() As Ektron.Cms.ContentAPI.RatingGraphDisplayType
```

### ContentRating.Padding Property

Padding, in pixels, to be applied between the content rating input and display graph.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("Padding, in pixels, to be applied between the content rating input and display
graph.")]
public int Padding;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("Padding, in pixels, to be applied between the content rating input and display
graph.")>
Public Property Padding() As Integer
```

### ContentRating.RatingLevelLabel Property

The label corresponding to a "Good" content rating. This defaults to "Outstanding".

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to a \"Good\" content rating. This defaults to
\"Ouotstanding\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_LEVEL)]
public string RatingLevelLabel;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to a \"Good\" content rating. This defaults to
\"Ouotstanding\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_LEVEL)>
Public Property RatingLevelLabel() As string
```

### ContentRating.ThankYouMessage Property

The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback.".

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_THANK)]
public string ThankYouMessage;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_THANK)>
Public Property ThankYouMessage() As string
```

### ContentRating.TotalRatingsLabel Property

The label that appears below the graph, and lists the number of rating submissions received. The default is: "Total Ratings".

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label that appears below the graph, and lists the number of rating
submissions received. The default is: \"Total Ratings\"")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_TOTAL)]
public string TotalRatingsLabel;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label that appears below the graph, and lists the number of rating
submissions received. The default is: \"Total Ratings\"")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_TOTAL)>
Public Property TotalRatingsLabel() As string
```

### ContentRating.UserCommentsHeader Property

The label corresponding to why the user rated content the way they did. Defaults to "Tell us why you rated the content this way".

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label corresponding to why the user rated content the way they did. Defaults
to \"Tell us why you rated the content this way\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_CMNT)]
public string UserCommentsHeader;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to why the user rated content the way they did. Defaults
to \"Tell us why you rated the content this way\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
```

```
<DefaultValue(EkWebControl.RES_CNTRATE_CMNT)>
Public Property UserCommentsHeader() As string
```

### ContentRating.ValidationMessage Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label displayed to the user when they try to submit a content rating without
selecting a value of 1-9. Defaults to \"Please select a rating before submitting your
feedback.\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_VALID)]
public string ValidationMessage;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label displayed to the user when they try to submit a content rating without
selecting a value of 1-9. Defaults to \"Please select a rating before submitting your
feedback.\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_VALID)>
Public Property ValidationMessage() As string
```

## ContentReview Class

The ContentReview server control has two ways it can be used. The first, places a star based rating scale on any page of your Web site. The scale lets Ektron CMS400.NET collect feedback by giving site visitors the opportunity to rate a particular content item. An item is rated when a visitor clicks on a star. Depending on the XSLT chosen, visitors can also submit review comments.

The second way the ContentReview control can be used is to retrieve and display visitor reviews and comments. In this mode, the control can display reviews of a content item or reviews by a specific site or membership user.

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

ContentReview.cs

### ContentReview Methods

The methods of the ContentReview class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 170) | Method that can be overridden and used to grab data from the back end for each server control. |
| | RaiseCallbackEvent (see page 170) | |
| | ToString (see page 170) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 170) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## ContentReview.Fill Method

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

**C#**

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

**Visual Basic**

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

## ContentReview.RaiseCallbackEvent Method

**C#**

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

**Visual Basic**

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

**Parameters**

| Parameters | Description |
|---|---|
| eventArgs | string |

## ContentReview.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

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

**Visual Basic**

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

## ContentReview.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

## ContentReview Properties

The properties of the ContentReview class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DefaultContentID (see page 171) | Gets or Sets the default content ID for this control. This is the content ID that will be rated if the DynamicParameter (see page 172) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 172) of "id", and will display DefaultContentID otherwise.) |
| | DisplayXslt (see page 172) | Select the type of review to display, or enter the path to a custom XSLT. (optional).<br>Choices are:<br>• **Ajax 5 Stars** - a five star rating system utilizing Ajax for display.<br>• **Ajax 5 Stars Comment** - a five star rating system utilizing Ajax for display. In addition, when you hover over the stars, a review box pops-up below the stars. Comments that are added are submitted via an Ajax call.<br>• **Ajax 5 Stars with Increments** - a five star rating system with half star increments that utilizes Ajax for display.<br>• **5 Stars** - a five star rating system that allows... more (see page 172) |
| | DynamicParameter (see page 172) | To make this content review control dynamic, select id. When you do, this server control is attached to the content block passed as a URL parameter. When using this control to retrieve a user's reviews, set this property to UserId to make the User (see page 913)'s ID dynamic. |
| | DynamicUserParameter (see page 173) | The querystring parameter for which to get the user ID. If a value is found in this parameter, it is used, or the UserID (see page 174) property is used. Important: The GetReviews (see page 173) property must be set to User (see page 913) for the control to use this property. |
| | GetReviews (see page 173) | This property returns a list of reviews for a content item or a specified user. It is used to decide which type of review information is retrieved by the server control. To display reviews on a page, you must use an XSLT that is set up for displaying reviews. |
| | MaxReviews (see page 173) | The number of reviews to retrieve if the GetReview property is set to Content (see page 788) or User (see page 913). 0 (zero) = unlimited results. |
| | Moderate (see page 174) | Setting this property to True allows CMS400.NET users to moderate reviews.<br>Additional information on moderating reviews can be found in the Ektron CMS400.NET Administrator Manual section "Managing HTML Content (see page 788)" > "Content (see page 788) Rating" |
| | RatingsMinimum (see page 174) | Sets a minimum number of the reviews and ratings before displaying the average rating. |
| | UserID (see page 174) | The user ID for which to get reviews. If left blank, reviews from all users are returned. If a user ID is specified, only reviews for that user are returned.<br>Important: The GetReviews (see page 173) property must be set to User (see page 913) for the control to use this property. |
| | XmlDoc (see page 175) | Returns the Xslt object for the collection. |

## ContentReview.DefaultContentID Property

Gets or Sets the default content ID for this control. This is the content ID that will be rated if the DynamicParameter (see page 172) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 172) of "id", and will display DefaultContentID otherwise.)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentReviewUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the default content ID for this control. This is the content ID
that will be rated if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will
get the value 2 for a DynamicParameter of \"id\", and will review DefaultContentID
otherwise.)")]
public long DefaultContentID;
```

**Visual Basic**

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

```
<Category("Ektron")>
<Editor(typeof(ContentReviewUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the default content ID for this control. This is the content ID
that will be rated if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will
get the value 2 for a DynamicParameter of \"id\", and will review DefaultContentID
otherwise.)")>
Public Property DefaultContentID() As long
```

### ContentReview.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

Choices are:

- **Ajax 5 Stars** - a five star rating system utilizing Ajax for display.
- **Ajax 5 Stars Comment** - a five star rating system utilizing Ajax for display. In addition, when you hover over the stars, a review box pops-up below the stars. Comments that are added are submitted via an Ajax call.
- **Ajax 5 Stars with Increments** - a five star rating system with half star increments that utilizes Ajax for display.
- **5 Stars** - a five star rating system that allows visitors to add text reviews of content.
- **5 Stars with Increments** - a five star rating system with half star increments that allows visitors to add text reviews of content.
- **Review List** - displays a list of reviews for content or a user. See Also:
- **Path to Custom Xslt** - enter the path to a custom XSLT.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltContentReviewTypeConverter))]
[DefaultValue("Ajax 5 Stars with increments")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltContentReviewTypeConverter))>
<DefaultValue("Ajax 5 Stars with increments")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### ContentReview.DynamicParameter Property

To make this content review control dynamic, select id. When you do, this server control is attached to the content block passed as a URL parameter. When using this control to retrieve a user's reviews, set this property to UserId to make the User (see page 913)'s ID dynamic.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("To make this content review control dynamic, select id. When you do, this server
control is attached to the content block passed as a URL parameter. When using this control to
retrieve a user?s reviews, set this property to UserId to make the user?s ID dynamic.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

**Visual Basic**

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

```
<Category("Ektron")>
<Description("To make this content review control dynamic, select id. When you do, this server
control is attached to the content block passed as a URL parameter. When using this control to
retrieve a user?s reviews, set this property to UserId to make the user?s ID dynamic.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

### ContentReview.DynamicUserParameter Property

The querystring parameter for which to get the user ID. If a value is found in this parameter, it is used, or the UserID (see page 174) property is used. Important: The GetReviews (see page 173) property must be set to User (see page 913) for the control to use this property.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The querystring parameter for which to get the user ID. If a value is found in
this parameter, it is used, or the UserID property is used. Important: The GetReviews property
must be set to User for the control to use this property.")]
[TypeConverter(typeof(DynamicUserIDTypeConverter))]
[DefaultValue("userid")]
public string DynamicUserParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The querystring parameter for which to get the user ID. If a value is found in
this parameter, it is used, or the UserID property is used. Important: The GetReviews property
must be set to User for the control to use this property.")>
<TypeConverter(typeof(DynamicUserIDTypeConverter))>
<DefaultValue("userid")>
Public Property DynamicUserParameter() As string
```

### ContentReview.GetReviews Property

This property returns a list of reviews for a content item or a specified user. It is used to decide which type of review information is retrieved by the server control. To display reviews on a page, you must use an XSLT that is set up for displaying reviews.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("This property returns a list of reviews for a content item or a specified user.
It is used to decide which type of review information is retrieved by the server control. To
display reviews on a page, you must use an XSLT that is set up for displaying reviews.")]
[DefaultValue(ReviewTypes.none)]
public ReviewTypes GetReviews;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("This property returns a list of reviews for a content item or a specified user.
It is used to decide which type of review information is retrieved by the server control. To
display reviews on a page, you must use an XSLT that is set up for displaying reviews.")>
<DefaultValue(ReviewTypes.none)>
Public Property GetReviews() As ReviewTypes
```

### ContentReview.MaxReviews Property

The number of reviews to retrieve if the GetReview property is set to Content (see page 788) or User (see page 913). 0 (zero) = unlimited results.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The number of reviews to retrieve if the GetReview property is set to Content or
User. 0  = unlimited results.")]
[DefaultValue(0)]
public int MaxReviews;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The number of reviews to retrieve if the GetReview property is set to Content or
User. 0  = unlimited results.")>
<DefaultValue(0)>
Public Property MaxReviews() As Integer
```

### ContentReview.Moderate Property

Setting this property to True allows CMS400.NET users to moderate reviews.

Additional information on moderating reviews can be found in the Ektron CMS400.NET Administrator Manual section “Managing HTML Content (see page 788)” > “Content (see page 788) Rating”

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Setting this property to True allows CMS400.NET users to moderate reviews.")]
[DefaultValue(false)]
public bool Moderate;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Setting this property to True allows CMS400.NET users to moderate reviews.")>
<DefaultValue(false)>
Public Property Moderate() As bool
```

### ContentReview.RatingsMinimum Property

Sets a minimum number of the reviews and ratings before displaying the average rating.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets a minimum number of the reviews and ratings before displaying the average
rating.")]
[DefaultValue(0)]
public int RatingsMinimum;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets a minimum number of the reviews and ratings before displaying the average
rating.")>
<DefaultValue(0)>
Public Property RatingsMinimum() As Integer
```

### ContentReview.UserID Property

The user ID for which to get reviews. If left blank, reviews from all users are returned. If a user ID is specified, only reviews for that user are returned.

Important: The GetReviews (see page 173) property must be set to User (see page 913) for the control to use this property.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The user ID for which to get reviews. If left blank, reviews from all users are
returned. If a user ID is specified, only reviews for that user are returned. Important: The
GetReviews property must be set to User for the control to use this property.")]
[DefaultValue(0)]
public long UserID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The user ID for which to get reviews. If left blank, reviews from all users are
returned. If a user ID is specified, only reviews for that user are returned. Important: The
GetReviews property must be set to User for the control to use this property.")>
<DefaultValue(0)>
Public Property UserID() As long
```

### ContentReview.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

## CurrencySelect Class

Summary description for CurrencySelect control

**Class Hierarchy**

**C#**

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

**Visual Basic**

```
<ToolboxData("<{0}:CurrencySelect runat=server></{0}:CurrencySelect>")>
Public Class CurrencySelect
Inherits ECommerceControl
```

**File**

CurrencySelect.cs

### CurrencySelect Methods

The methods of the CurrencySelect class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 176) | Populates the data. |
| | ToString (see page 176) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 176) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### CurrencySelect.Fill Method

Populates the data.

**C#**

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

**Visual Basic**

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

### CurrencySelect.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

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

**Visual Basic**

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

### CurrencySelect.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

### CurrencySelect Properties

The properties of the CurrencySelect class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 176) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DisplayXslt (see page 177) | Specify an external XSLT file to render the control. By default, the control uses CurrencySelect.xsl. This file is located in: |
| | EkItem (see page 178) | |
| | Stylesheet (see page 178) | Specify the path to a style sheet for use with this server control. The location can be relative or absolute. Leave blank to use the default style sheet. |

### CurrencySelect.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### CurrencySelect.DisplayXslt Property

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

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

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses CurrencySelect.xsl. This file is located in \"<Site Root>\\Workarea\\Xslt\\Commerce.\"
Important: If you want to edit this file, it is strongly recommended that you make a copy,
change its name and move it outside of the Workarea folder. If you make changes to this file
and do not move it out of the Workarea, this file and the changes will be lost when you
upgrade.")]
[TypeConverter(typeof(XsltOrderTypeConverter))]
[DefaultValue("Commerce/CurrencySelect.xsl")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses CurrencySelect.xsl. This file is located in \"<Site Root>\\Workarea\\Xslt\\Commerce.\"
Important: If you want to edit this file, it is strongly recommended that you make a copy,
change its name and move it outside of the Workarea folder. If you make changes to this file
and do not move it out of the Workarea, this file and the changes will be lost when you
upgrade.")>
<TypeConverter(typeof(XsltOrderTypeConverter))>
<DefaultValue("Commerce/CurrencySelect.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### CurrencySelect.EkItem Property

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Returns Currency Data.")]
[Personalizable()]
public List<CurrencyData> EkItem;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Returns Currency Data.")>
<Personalizable()>
Public Property EkItem() As List<CurrencyData>
```

### CurrencySelect.Stylesheet Property

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

**C#**

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(CurrencySelectCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/CurrencySelect.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(CurrencySelectCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/CurrencySelect.css")>
Public Property Stylesheet() As string
```

## DefaultCatalogIDUITypeEditor Class

Summary description for DefaultCatalogIDUITypeEditor.

**Class Hierarchy**

**C#**

```
public class DefaultCatalogIDUITypeEditor : UITypeEditor;
```

**Visual Basic**

```
Public Class DefaultCatalogIDUITypeEditor
Inherits UITypeEditor
```

**File**

DefaultCatalogIDUITypeEditor.cs

## DesignTimeDiagnostic Class

The DesignTimeDiagnostic server control is used to verify the connection to Ektron's server controls Web service. When added

to a Web form, this control provides the following information:

- **Web Service Path** - displays the server control Web service path in your web.config file
- **Web Service Exists** - calls a method in the Web service that returns True when the Web service exists. When it does not, it returns False.
- **Test Login (see page 337) Control** - displays the Login (see page 337) server control to show that the Web service is connected and working properly. When not connecting properly, an error message is displayed.

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

DesignTimeDiagnostic.cs

## DesignTimeDiagnostic Methods

The methods of the DesignTimeDiagnostic class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 179) | Method that can be overridden and used to grab data from the back end for each server control. |

## DesignTimeDiagnostic.Fill Method

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

**C#**

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

**Visual Basic**

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

## DesignTimeDiagnostic Properties

The properties of the DesignTimeDiagnostic class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Authenticated (see page 180) | 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 180) | 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 180) | Set a language for viewing content blocks, collections, etc. in different languages. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser). |
| | SuppressWrapperTags (see page 180) | Suppresses the output of span/div tags around the control. |
| | WrapTag (see page 180) | Allows a developer to specify control's tag. Default is assumed "". |

### DesignTimeDiagnostic.Authenticated Property

DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in Visual Studio.NET.

**C#**

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

**Visual Basic**

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

### DesignTimeDiagnostic.Hide Property

Alias for the 'Visible' property. Visible is set to mirror this value. This property is used to hide output in design time, whereas Visible does not hide output in design time.

**C#**

```
[Browsable(false)]
public new bool Hide;
```

**Visual Basic**

```
<Browsable(false)>
Public new Property Hide() As bool
```

### DesignTimeDiagnostic.Language Property

Set a language for viewing content blocks, collections, etc. in different languages. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).

**C#**

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

**Visual Basic**

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

### DesignTimeDiagnostic.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

**C#**

```
[Browsable(false)]
public new bool SuppressWrapperTags;
```

**Visual Basic**

```
<Browsable(false)>
Public new Property SuppressWrapperTags() As bool
```

### DesignTimeDiagnostic.WrapTag Property

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

**C#**

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

**Visual Basic**

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

## DhtmlMenu Class

The DhtmlMenu server control lets you display a menu with drop down boxes and submenus. This lets developers add a menu to a page without taking up a lot space.

**Class Hierarchy**

**C#**

```
[DefaultProperty("DefaultMenuID")]
public class DhtmlMenu : Ektron.Cms.Controls.Menu;
```

**Visual Basic**

```
<DefaultProperty("DefaultMenuID")>
Public Class DhtmlMenu
Inherits Ektron.Cms.Controls.Menu
```

**File**

DhtmlMenu.cs

### DhtmlMenu Methods

The methods of the DhtmlMenu class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 181) | Calls up the database and updates the contents of the Menu (see page 358) if the data is out of sync with the specified parameters. |
| | Wrap (see page 181) | Method that can be overridden and used to wrap an Ektron Toolbar corresponding to a particular Control around some Html text. |

### DhtmlMenu.Fill Method

Calls up the database and updates the contents of the Menu (see page 358) 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 183) | Ignored. The property has no effect. |
| | Align (see page 183) | Specify the horizontal alignment for the root menu text. The default is Center. Choices are:<br>• **NotSet** - Use the default setting.<br>• **Left** - Aligns text to the left.<br>• **Center** - Aligns text to the center.<br>• **Right** - Aligns text to the right.<br>• **Justify** - Aligns text justified to both margins. |
| | BackColor (see page 183) | BackColor sets the menu background color. |
| | BorderColor (see page 183) | Sets the border color around the editor. |
| | BorderStyle (see page 184) | Sets the border style around the editor. |
| | BorderWidth (see page 184) | Sets the border width around the editor. |
| | CssClass (see page 184) | Apply a .css class to the root menu. Enter the class name from the .css defined in the Stylesheet (see page 186) property. Since pop_style.css is the default style sheet, you can use its classes without defining it in the Stylesheet (see page 186) property. |
| | DisplayXslt (see page 184) | Ignored. The property has no effect. |
| | Enabled (see page 184) | Ignored. The property has no effect. |
| | EnableIE6FormsFix (see page 185) | 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 185) | The height (in pixels or percent) of the editor, including toolbars. |
| | HorizontalDisplay (see page 185) | Popup root menu horizontally. |
| | MenuHoverOverColor (see page 186) | `HoverOverColor` sets the menu hoverover color. |
| | MenuStyle (see page 186) | Specify the custom style to use. If specified, all specified appearance properties will be ignored. |
| | Stylesheet (see page 186) | Specify the style sheet relative to site root. Leave blank for default. |
| | SubMenuBackColor (see page 186) | `SubMenuBackColor` sets the sub menu background color. |
| | SubMenuHoverOverColor (see page 187) | `SubMenuHoverOverColor` sets the menu hoverover color. |
| | SubMenuItemHeight (see page 187) | `SubMenuItemHeight` is the height in pixels of submenu items. |
| | SubMenuItemWidth (see page 187) | `SubMenuItemWidth` is the width in pixels of submenu items. |
| | SubMenuTopOffset (see page 188) | `SubMenuTopOffset` sets the horizontal offset of submenu popups relative to the parent submenu item. |
| | TabIndex (see page 188) | Ignored. The property has no effect. |
| | ToolTip (see page 188) | Ignored. The property has no effect. |
| | Width (see page 188) | The width (in pixels or percent) of the editor, including toolbars. |

### DhtmlMenu.AccessKey Property

Ignored. The property has no effect.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string AccessKey;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property AccessKey() As string
```

### DhtmlMenu.Align Property

Specify the horizontal alignment for the root menu text. The default is Center. Choices are:

- **NotSet** - Use the default setting.
- **Left** - Aligns text to the left.
- **Center** - Aligns text to the center.
- **Right** - Aligns text to the right.
- **Justify** - Aligns text justified to both margins.

**C#**

```
[Description("Specify the horizontal alignment for the root menu text. Choices are: NotSet,
Left, Center, Right or Justify.")]
[Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(HorizontalAlign))]
public HorizontalAlign Align;
```

**Visual Basic**

```
<Description("Specify the horizontal alignment for the root menu text. Choices are: NotSet,
Left, Center, Right or Justify.")>
<Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(HorizontalAlign))>
Public Property Align() As HorizontalAlign
```

### DhtmlMenu.BackColor Property

BackColor sets the menu background color.

**C#**

```
[NotifyParentProperty(true), Browsable(true), Bindable(true)]
[Description("BackColor sets the menu background color.")]
public override System.Drawing.Color BackColor;
```

**Visual Basic**

```
<NotifyParentProperty(true), Browsable(true), Bindable(true)>
<Description("BackColor sets the menu background color.")>
Public override Property BackColor() As System.Drawing.Color
```

### DhtmlMenu.BorderColor Property

Sets the border color around the editor.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override System.Drawing.Color BorderColor;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property BorderColor() As System.Drawing.Color
```

### DhtmlMenu.BorderStyle Property

Sets the border style around the editor.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override BorderStyle BorderStyle;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property BorderStyle() As BorderStyle
```

### DhtmlMenu.BorderWidth Property

Sets the border width around the editor.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override Unit BorderWidth;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property BorderWidth() As Unit
```

### DhtmlMenu.CssClass Property

Apply a .css class to the root menu. Enter the class name from the .css defined in the Stylesheet (see page 186) property. Since pop_style.css is the default style sheet, you can use its classes without defining it in the Stylesheet (see page 186) property.

**C#**

```
[Description("Specify the class name to use instead of internal style."), Category("Ektron")]
public override string CssClass;
```

**Visual Basic**

```
<Description("Specify the class name to use instead of internal style."), Category("Ektron")>
Public override Property CssClass() As string
```

### DhtmlMenu.DisplayXslt Property

Ignored. The property has no effect.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string DisplayXslt;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property DisplayXslt() As string
```

### DhtmlMenu.Enabled Property

Ignored. The property has no effect.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
```

```
public override bool Enabled;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property Enabled() As bool
```

### DhtmlMenu.EnableIE6FormsFix Property

This should only be enabled when you have a DHTML menu rendering over an HTML Select element. In IE6, the DHTML will render below the drop down. This will temporarily disable the element when the DHTML menu is rendered.

**C#**

```
[Browsable(true)]
[Description("This should only be enabled when you have a DHTML menu rendering over an HTML
Select element. In IE6, the DHTML will render below the drop down. This will temporarily
disable the element when the DHTML menu is rendered.")]
[Bindable(true)]
[Category("Ektron")]
public bool EnableIE6FormsFix;
```

**Visual Basic**

```
<Browsable(true)>
<Description("This should only be enabled when you have a DHTML menu rendering over an HTML
Select element. In IE6, the DHTML will render below the drop down. This will temporarily
disable the element when the DHTML menu is rendered.")>
<Bindable(true)>
<Category("Ektron")>
Public Property EnableIE6FormsFix() As bool
```

### DhtmlMenu.Height Property

The height (in pixels or percent) of the editor, including toolbars.

**C#**

```
[NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultHeight)]
[Description("The height (in pixels or percent) of the editor, including toolbars.")]
public override Unit Height;
```

**Visual Basic**

```
<NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultHeight)>
<Description("The height (in pixels or percent) of the editor, including toolbars.")>
Public override Property Height() As Unit
```

### DhtmlMenu.HorizontalDisplay Property

Popup root menu horizontally.

**C#**

```
[Description("Popup root menu horizontally."), Category("Ektron")]
[Browsable(true)]
[Bindable(false)]
[DefaultValue(false)]
public bool HorizontalDisplay;
```

**Visual Basic**

```
<Description("Popup root menu horizontally."), Category("Ektron")>
<Browsable(true)>
<Bindable(false)>
<DefaultValue(false)>
Public Property HorizontalDisplay() As bool
```

### DhtmlMenu.MenuHoverOverColor Property

**C#**

```
[Description("HoverOverColor sets the menu hoverover color."), Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color MenuHoverOverColor;
```

**Visual Basic**

```
<Description("HoverOverColor sets the menu hoverover color."), Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property MenuHoverOverColor() As Color
```

**Description**

`HoverOverColor` sets the menu hoverover color.

### DhtmlMenu.MenuStyle Property

Specify the custom style to use. If specified, all specified appearance properties will be ignored.

**C#**

```
[Description("Specify the custom style to use. If specified, all specified appearance
properties will be ignored."), Category("Ektron")]
public string MenuStyle;
```

**Visual Basic**

```
<Description("Specify the custom style to use. If specified, all specified appearance
properties will be ignored."), Category("Ektron")>
Public Property MenuStyle() As string
```

### DhtmlMenu.Stylesheet Property

Specify the style sheet relative to site root. Leave blank for default.

**C#**

```
[Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
public override string Stylesheet;
```

**Visual Basic**

```
<Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
Public override Property Stylesheet() As string
```

### DhtmlMenu.SubMenuBackColor Property

**C#**

```
[Description("SubMenuBackColor sets the sub menu background color."), Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color SubMenuBackColor;
```

**Visual Basic**

```
<Description("SubMenuBackColor sets the sub menu background color."), Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property SubMenuBackColor() As Color
```

**Description**

`SubMenuBackColor` sets the sub menu background color.

### DhtmlMenu.SubMenuHoverOverColor Property

**C#**

```
[Description("SubMenuHoverOverColor sets the sub menu hoverover color."), Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color SubMenuHoverOverColor;
```

**Visual Basic**

```
<Description("SubMenuHoverOverColor sets the sub menu hoverover color."), Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property SubMenuHoverOverColor() As Color
```

**Description**

`SubMenuHoverOverColor` sets the menu hoverover color.

### DhtmlMenu.SubMenuItemHeight Property

**C#**

```
[Description("SubMenuItemHeight is the height in pixels of submenu items."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(Int16Converter))]
public Int16 SubMenuItemHeight;
```

**Visual Basic**

```
<Description("SubMenuItemHeight is the height in pixels of submenu items."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(Int16Converter))>
Public Property SubMenuItemHeight() As Int16
```

**Description**

`SubMenuItemHeight` is the height in pixels of submenu items.

### DhtmlMenu.SubMenuItemWidth Property

**C#**

```
[Description("SubMenuItemWidth is the width in pixels of submenu items."), Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(Int16Converter))]
public Int16 SubMenuItemWidth;
```

**Visual Basic**

```
<Description("SubMenuItemWidth is the width in pixels of submenu items."), Category("Ektron")>
```

```
<Bindable(true)>
<TypeConverter(typeof(Int16Converter))>
Public Property SubMenuItemWidth() As Int16
```

**Description**

SubMenuItemWidth is the width in pixels of submenu items.

### DhtmlMenu.SubMenuTopOffset Property

**C#**

```
[Description("SubMenuTopOffset sets the horizontal offset of submenu popups relative to the
parent submenu item."), Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(Int16Converter))]
public Int16 SubMenuTopOffset;
```

**Visual Basic**

```
<Description("SubMenuTopOffset sets the horizontal offset of submenu popups relative to the
parent submenu item."), Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(Int16Converter))>
Public Property SubMenuTopOffset() As Int16
```

**Description**

SubMenuTopOffset sets the horizontal offset of submenu popups relative to the parent submenu item.

### DhtmlMenu.TabIndex Property

Ignored. The property has no effect.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override short TabIndex;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property TabIndex() As short
```

### DhtmlMenu.ToolTip Property

Ignored. The property has no effect.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string ToolTip;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property ToolTip() As string
```

### DhtmlMenu.Width Property

The width (in pixels or percent) of the editor, including toolbars.

**C#**

```
[NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultWidth)]
[Description("The width (in pixels or percent) of the editor, including toolbars.")]
public override Unit Width;
```

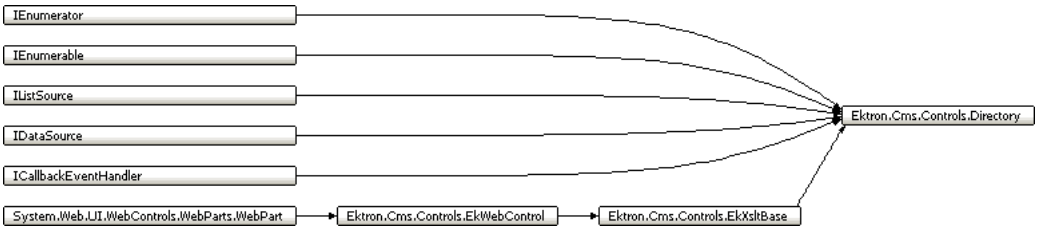
**Visual Basic**

```
<NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultWidth)>
<Description("The width (in pixels or percent) of the editor, including toolbars.")>
Public override Property Width() As Unit
```

## Directory Class

The Directory server control lets you customize the behavior of the Taxonomy (see page 842) feature. You place this control on any Web form to display a taxonomy.

**Class Hierarchy**




**C#**

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

**Visual Basic**

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

**File**

Taxonomy.cs

### Directory Methods

The methods of the Directory class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | CopyTo (see page 190) | 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 190) | Populates the data for the Taxonomy (see page 842). |
| | GetEnumerator (see page 190) | Gets a System.Collections.IEnumerator for the Taxonomy (see page 842). |
| | GetView (see page 190) | Gets the view data using dataset |
| | GetViewNames (see page 190) | gets the view names |
| | MoveNext (see page 191) | Move to next index. |
| | Reset (see page 191) | Resets the index. |
| | this (see page 191) | Returns the current ContentItem.ContentRow. |
| | ToString (see page 191) | Overrides the system ToString function. |
| | Wrap (see page 192) | Wrap the Html toolbar. |

### Directory.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

**C#**

```
public void CopyTo(Array array, int index);
```

**Visual Basic**

```
Public Function CopyTo(array As Array, index As Integer) As void
```

### Directory.Fill Method

Populates the data for the Taxonomy (see page 842).

**C#**

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

**Visual Basic**

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

### Directory.GetEnumerator Method

Gets a System.Collections.IEnumerator for the Taxonomy (see page 842).

**C#**

```
[Browsable(false)]
public IEnumerator GetEnumerator();
```

**Visual Basic**

```
<Browsable(false)>
Public Function GetEnumerator() As IEnumerator
```

### Directory.GetView Method

Gets the view data using dataset

**C#**

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

**Visual Basic**

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

**Parameters**

| Parameters | Description |
|---|---|
| viewName | |

**Returns**

DataSourceView

### Directory.GetViewNames Method

gets the view names

**C#**

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

**Visual Basic**

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

**Returns**

View names

### Directory.MoveNext Method

Move to next index.

**C#**

```
public bool MoveNext();
```

**Visual Basic**

```
Public Function MoveNext() As bool
```

**Returns**

Next Index

### Directory.Reset Method

Resets the index.

**C#**

```
public void Reset();
```

**Visual Basic**

```
Public Function Reset() As void
```

### Directory.this Indexer

Returns the current ContentItem.ContentRow.

**C#**

```
[Browsable(false)]
public Ektron.Cms.Controls.ContentItem.ContentRow this[int i];
```

**Visual Basic**

```
<Browsable(false)>
Public Sub this(i As Integer)
```

**Parameters**

| Parameters | Description |
|---|---|
| i | Index value |

**Returns**

Content (see page 788) Row

### Directory.ToString Method

Overrides the system ToString function.

**C#**

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

**Visual Basic**

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

**Returns**

String

## Directory.Wrap Method

Wrap the Html toolbar.

**C#**

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

**Visual Basic**

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

**Parameters**

| Parameters | Description |
|---|---|
| Html | |

## Directory Properties

The properties of the Directory class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AddItemFolderId (see page 193) | Set or Get and Folder (see page 652) Id where new items will be added. |
| | CacheInterval (see page 194) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | Count (see page 194) | Indicates the number of elements in your Taxonomy (see page 842). |
| | Current (see page 194) | Gets the current index. |
| | DisplayXslt (see page 194) | The Xslt to use to render the control (optional). |
| | DynamicParameter (see page 195) | Gets or Sets the QueryString parameter to read a Taxonomy (see page 842) ID dynamically. Set to String.Empty if you wish to always display the default Taxonomy (see page 842) ID (static.) |
| | EkItems (see page 195) | Returns the array of ContentBase. |
| | EnableAjax (see page 195) | Set to true to enable Ajax searches. When enabled, the MaxResults (see page 198) property determines the maximum number of results per page.<br>• **True** = Enable Ajax Search (see page 455) (default value)<br>• **False** = Original HTML Search (see page 455) |
| | EnablePaging (see page 196) | This property, in conjunction with the MaxResults (see page 198) property, lets site visitors view an unlimited number of taxonomy items while controlling the amount of screen space. To accomplish this, the taxonomy display is limited to the number set in the MaxResults (see page 198) property. |
| | EnableSearch (see page 196) | Set to true if you want a search box (shown below) to appear above this taxonomy display. |
| | IncludeIcons (see page 196) | Determines whether or not to display an icon next to your navigation lists. |
| | IsSynchronized (see page 197) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | ItemSortOrder (see page 197) | Specify the sort order of results. Choices are:<br>• **taxonomy_item _display_order** - the order of taxonomy items as set in the Workarea. For additional information, see the CMS400.NET Administrator section "Taxonomy (see page 842)" > "Additional Taxonomy (see page 842)/Category Tasks" > "Change Order of Content (see page 788) Assigned to a Taxonomy (see page 842)/Category"<br>• **content_title** - the content is listed in alphabetical order by title.<br>• **date_created** - content is listed in the order by which it was created.<br>• **last_edit_date** - content is listed in order by its last edit date. |

| | | |
|---|---|---|
| | LinkTarget (see page 197) | Defines the way a link acts when it is clicked. The choices are:<br>• **_blank** = loads the link in a new blank window. This window is not named.<br>• **_self** = loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.<br>• **_parent** = loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.<br>• **_top** = loads the link in the full body of the window. This defaults to acting like "_self" if the document is already... more (see page 197) |
| | MarkupLanguage (see page 198) | Enter name of a markup file that manages the display of this server control. |
| | MaxResults (see page 198) | Maximum number of results to be returned. If set to zero, there is no limit. |
| | Recursive (see page 198) | Determines if subfolders of the selected folders should be brought into the list. |
| | ShowAllChecked (see page 198) | When set to True a checkmark appears by default in the **show all** check box. This check box shows all categories even when they have no content assigned to them. The **show all** check box is available when a user is logged in. |
| | ShowCount (see page 199) | Indicates if the number of taxonomy items appears next to each category when displayed in the Web site. The default is: False |
| | SortDirection (see page 199) | Set the direction of the output |
| | SuppressWrapperTags (see page 199) | Suppresses the output of span/div tags around the control. As of release 7.5.2, this property is hidden by default. This property is set to false because Ajax uses  tags to rewrite the region around the tag. You cannot change the value to **True**. |
| | SyncRoot (see page 200) | Gets an object that can be used to synchronize the Taxonomy (see page 842). |
| | TaxonomyCols (see page 200) | Enter the number of columns in which this taxonomy/category will appear on the page. |
| | TaxonomyDepth (see page 200) | Enter the number of taxonomy levels to appear below each taxonomy/category. For example, if the taxonomy is Businesses > Restaurants > Pizza and you set TaxonomyDepth to 2, only Business and Restaurants appear.<br>**Important!** - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth.<br>A depth greater than 1 is only useful if you create a custom output using the Taxonomy (see page 842)'s XML. |
| | TaxonomyDescription (see page 200) | Returns taxonomy's description. |
| | TaxonomyId (see page 201) | Enter the ID number of the taxonomy or category to appear in this server control. Root is 0. |
| | TaxonomyItemCols (see page 201) | Enter the number of columns in which this taxonomy/category items(articles) will appear on the page. |
| | TaxonomyPath (see page 201) | Returns taxonomy's path. |
| | TaxonomyTitle (see page 201) | Returns the Taxonomy (see page 842)'s title. |
| | TaxonomyTreeData (see page 202) | Returns taxonomy's data. |
| | TaxonomyXml (see page 202) | Returns taxonomy's XML. (includes subcategory and items) |
| | XmlDoc (see page 202) | Returns the XML object loaded for Taxonomy (see page 842). |

## Directory.AddItemFolderId Property

Set or Get and Folder (see page 652) Id where new items will be added.

**C#**

```
[Browsable(true)]
[Bindable(true)]
[Description("Set or Get and Folder Id where new items will be added.")]
[DefaultValue(0)]
[Category("Ektron")]
public virtual long AddItemFolderId;
```

**Visual Basic**

```
<Browsable(true)>
<Bindable(true)>
<Description("Set or Get and Folder Id where new items will be added.")>
```

```
<DefaultValue(0)>
<Category("Ektron")>
Public virtual Property AddItemFolderId() As long
```

### Directory.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### Directory.Count Property

Indicates the number of elements in your Taxonomy (see page 842).

**C#**

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

**Visual Basic**

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

### Directory.Current Property

Gets the current index.

**C#**

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

**Visual Basic**

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

### Directory.DisplayXslt Property

The Xslt to use to render the control (optional).

**C#**

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

**Visual Basic**

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

### Directory.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Taxonomy (see page 842) ID dynamically. Set to String.Empty if you wish to always display the default Taxonomy (see page 842) ID (static.)

**C#**

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

**Visual Basic**

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

### Directory.EkItems Property

Returns the array of ContentBase.

**C#**

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItems;
```

**Visual Basic**

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

### Directory.EnableAjax Property

Set to true to enable Ajax searches. When enabled, the MaxResults (see page 198) property determines the maximum number of results per page.

- **True** = Enable Ajax Search (see page 455) (default value)
- **False** = Original HTML Search (see page 455)

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Description("Enable/Disable ajax
based retrival.")]
[Personalizable(), WebBrowsable()]
public virtual bool EnableAjax;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Description("Enable/Disable ajax
based retrival.")>
<Personalizable(), WebBrowsable()>
Public virtual Property EnableAjax() As bool
```

### Directory.EnablePaging Property

This property, in conjunction with the MaxResults (see page 198) property, lets site visitors view an unlimited number of taxonomy items while controlling the amount of screen space. To accomplish this, the taxonomy display is limited to the number set in the MaxResults (see page 198) property.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(0)]
[Description("Set to True to enable paging if the total records exeeds the MaxResult value.
Use this property with non zero value of MaxResults.")]
[Personalizable(), WebBrowsable()]
public bool EnablePaging;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(0)>
<Description("Set to True to enable paging if the total records exeeds the MaxResult value.
Use this property with non zero value of MaxResults.")>
<Personalizable(), WebBrowsable()>
Public Property EnablePaging() As bool
```

### Directory.EnableSearch Property

Set to true if you want a search box (shown below) to appear above this taxonomy display.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Browsable(true)]
[Description("When set to True, this property displays a search box to search thru the
taxonomy.")]
[Personalizable(), WebBrowsable()]
public virtual Boolean EnableSearch;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Browsable(true)>
<Description("When set to True, this property displays a search box to search thru the
taxonomy.")>
<Personalizable(), WebBrowsable()>
Public virtual Property EnableSearch() As Boolean
```

### Directory.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists.

**C#**

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.")]
[DefaultValue(false)]
[Category("Ektron")]
[Personalizable(), WebBrowsable()]
public bool IncludeIcons;
```

**Visual Basic**

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.")>
<DefaultValue(false)>
<Category("Ektron")>
<Personalizable(), WebBrowsable()>
Public Property IncludeIcons() As bool
```

### Directory.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

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

**Visual Basic**

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

### Directory.ItemSortOrder Property

Specify the sort order of results. Choices are:

- **taxonomy_item _display_order** - the order of taxonomy items as set in the Workarea. For additional information, see the CMS400.NET Administrator section “Taxonomy (see page 842)” > “Additional Taxonomy (see page 842)/Category Tasks” > “Change Order of Content (see page 788) Assigned to a Taxonomy (see page 842)/Category”
- **content_title** - the content is listed in alphabetical order by title.
- **date_created** - content is listed in the order by which it was created.
- **last_edit_date** - content is listed in order by its last edit date.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(TaxonomyItemSortOrder.content_title),
Description("Set the direction of the output.")]
[Personalizable(), WebBrowsable()]
public TaxonomyItemSortOrder ItemSortOrder;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(TaxonomyItemSortOrder.content_title),
Description("Set the direction of the output.")>
<Personalizable(), WebBrowsable()>
Public Property ItemSortOrder() As TaxonomyItemSortOrder
```

### Directory.LinkTarget Property

Defines the way a link acts when it is clicked. The choices are:

- **_blank** = loads the link in a new blank window. This window is not named.
- **_self** = loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- **_parent** = loads the link in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent.
- **_top** = loads the link in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

**C#**

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

**Visual Basic**

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

### Directory.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

### Directory.MaxResults Property

Maximum number of results to be returned. If set to zero, there is no limit.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to be returned. If set to zero, there is no
limit.")]
[Personalizable(), WebBrowsable()]
public int MaxResults;
```

**Visual Basic**

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

### Directory.Recursive Property

Determines if subfolders of the selected folders should be brought into the list.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set Recursive to true if you wish to grab results in subfolders of your
FolderPath.")]
public virtual bool Recursive;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set Recursive to true if you wish to grab results in subfolders of your
FolderPath.")>
Public virtual Property Recursive() As bool
```

### Directory.ShowAllChecked Property

When set to True a checkmark appears by default in the **show all** check box. This check box shows all categories even when they have no content assigned to them. The **show all** check box is available when a user is logged in.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("false"), Browsable(true)]
[Description("When set to True a checkmark appears by default in the \"show all\" check box.")]
public virtual bool ShowAllChecked;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("false"), Browsable(true)>
<Description("When set to True a checkmark appears by default in the \"show all\" check box.")>
Public virtual Property ShowAllChecked() As bool
```

### Directory.ShowCount Property

Indicates if the number of taxonomy items appears next to each category when displayed in the Web site. The default is: False

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(false)]
[Description("Indicates if the number of taxonomy items appears next to each category when
displayed in the Web site. The default is: False")]
[Personalizable(), WebBrowsable()]
public bool ShowCount;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(false)>
<Description("Indicates if the number of taxonomy items appears next to each category when
displayed in the Web site. The default is: False")>
<Personalizable(), WebBrowsable()>
Public Property ShowCount() As bool
```

### Directory.SortDirection Property

Set the direction of the output

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(OrderByDirection.Ascending),
Description("Set the direction of the output.")]
[Personalizable(), WebBrowsable()]
public OrderByDirection SortDirection;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(OrderByDirection.Ascending),
Description("Set the direction of the output.")>
<Personalizable(), WebBrowsable()>
Public Property SortDirection() As OrderByDirection
```

### Directory.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control. As of release 7.5.2, this property is hidden by default. This property is set to false because Ajax uses  tags to rewrite the region around the tag. You cannot change the value to **True**.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Browsable(false)]
[Description("Suppresses the output of span/div tags around the control. As of release 7.5.2,
this property is hidden by default. This property is set to false because Ajax uses <div> tags
to rewrite the region around the tag. You cannot change the value to <b>True</b>.")]
public override bool SuppressWrapperTags;
```

**Visual Basic**

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

```
<DefaultValue(false)>
<Browsable(false)>
<Description("Suppresses the output of span/div tags around the control. As of release 7.5.2,
this property is hidden by default. This property is set to false because Ajax uses <div> tags
to rewrite the region around the tag. You cannot change the value to <b>True</b>.")>
Public override Property SuppressWrapperTags() As bool
```

### Directory.SyncRoot Property

Gets an object that can be used to synchronize the Taxonomy (see page 842).

**C#**

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

**Visual Basic**

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

### Directory.TaxonomyCols Property

Enter the number of columns in which this taxonomy/category will appear on the page.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)]
[Description("This aligns the taxonomy categories in number of columns.")]
[Personalizable(), WebBrowsable()]
public int TaxonomyCols;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("This aligns the taxonomy categories in number of columns.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyCols() As Integer
```

### Directory.TaxonomyDepth Property

Enter the number of taxonomy levels to appear below each taxonomy/category. For example, if the taxonomy is Businesses > Restaurants > Pizza and you set TaxonomyDepth to 2, only Business and Restaurants appear.

**Important!** - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth.

A depth greater than 1 is only useful if you create a custom output using the Taxonomy (see page 842)'s XML.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)]
[Description("Enter the number of taxonomy levels to appear below each taxonomy/category. For
example, if the taxonomy is Businesses > Restaurants > Pizza and you set TaxonomyDepth to 2,
only Business and Restaurants appear.")]
public virtual int TaxonomyDepth;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("Enter the number of taxonomy levels to appear below each taxonomy/category. For
example, if the taxonomy is Businesses > Restaurants > Pizza and you set TaxonomyDepth to 2,
only Business and Restaurants appear.")>
Public virtual Property TaxonomyDepth() As Integer
```

### Directory.TaxonomyDescription Property

Returns taxonomy's description.

**C#**

```
public string TaxonomyDescription;
```

**Visual Basic**

```
Public Property TaxonomyDescription() As string
```

### Directory.TaxonomyId Property

Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)]
[Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public virtual long TaxonomyId;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)>
<Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public virtual Property TaxonomyId() As long
```

### Directory.TaxonomyItemCols Property

Enter the number of columns in which this taxonomy/category items(articles) will appear on the page.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)]
[Description("This aligns the taxonomy categories in number of columns.")]
[Personalizable(), WebBrowsable()]
public int TaxonomyItemCols;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("This aligns the taxonomy categories in number of columns.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyItemCols() As Integer
```

### Directory.TaxonomyPath Property

Returns taxonomy's path.

**C#**

```
public string TaxonomyPath;
```

**Visual Basic**

```
Public Property TaxonomyPath() As string
```

### Directory.TaxonomyTitle Property

Returns the Taxonomy (see page 842)'s title.

**C#**

```
public string TaxonomyTitle;
```

**Visual Basic**

```
Public Property TaxonomyTitle() As string
```

### Directory.TaxonomyTreeData Property

Returns taxonomy's data.

**C#**

```
public TaxonomyData TaxonomyTreeData;
```

**Visual Basic**

```
Public Property TaxonomyTreeData() As TaxonomyData
```

### Directory.TaxonomyXml Property

Returns taxonomy's XML. (includes subcategory and items)

**C#**

```
public string TaxonomyXml;
```

**Visual Basic**

```
Public Property TaxonomyXml() As string
```

### Directory.XmlDoc Property

Returns the XML object loaded for Taxonomy (see page 842).

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

## EktronCatalogPart Class

When placed inside a Microsoft CatalogZone, the EktronCatalogPart allows users to choose the type of CMS400.NET control that is added to a WebPartZone.

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

CatalogPart.cs

### EktronCatalogPart Properties

The properties of the EktronCatalogPart class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | BlogID (see page 203) | The ID of the blog in CMS400.NET. |
| | BoardID (see page 203) | The ID of the discussion board. |
| | CalendarDynamicParameter (see page 204) | To make this calendar dynamic, select calendar_id. When you do, this server control uses the calendar passed as a URL parameter. |
| | CollectionDynamicParameter (see page 204) | To make this collection dynamic, select id. When you do, this server control uses the collection passed as a URL parameter. |
| | ContentBlockDynamicParameter (see page 204) | Select id. When you do, this server control uses the content block passed as a URL parameter. |
| | DefaultCalendarID (see page 205) | The ID number of the calendar that appears where you inserted this server control. |
| | DefaultCollectionID (see page 205) | The ID of a collection that appears where you inserted this server control if no other collection is identified, or is not available. |
| | DefaultContentID (see page 205) | The ID of a content block that appears where you insert this server control. |
| | DefaultFormID (see page 206) | The ID of a FormBlock (see page 246) that appears where you inserted this server control if no other form block is identified. |
| | EnableBlogControl (see page 206) | Set to True to allow a Blog (see page 24) Control to appear in the catalog. |
| | EnableCalendarControl (see page 206) | Set to true to allow a Calendar (see page 69) control to appear in the Catalog. |
| | EnableCollectionControl (see page 206) | Set to True to allow a Collection (see page 88) control to appear in the Catalog. |
| | EnableContentBlockControl (see page 207) | Set to True to allow a Content (see page 788) Block Control to appear in the catalog. |
| | EnableDiscussionBoardsControl (see page 207) | Set to True to allow a Forum (see page 256) (Discussion Board) control to appear in the catalog. |
| | EnableFormBlockControl (see page 207) | Set to True to allow a FormBlock (see page 246) Control to appear in the catalog. |
| | EnableListSummaryControl (see page 208) | Set to True to allow a ListSummary (see page 325) Control to appear in the catalog. |
| | EnableThreadedDiscussionsControl (see page 208) | Set to True to allow a Forum (see page 256) (Discussion Board) control to appear in the catalog. |
| | FolderID (see page 208) | The folder that contains the content blocks which appear in the list summary. |
| | FormBlockDynamicParameter (see page 208) | To make this form block dynamic, select ekfrm. When you do, this server control uses the form block passed as a URL parameter. |
| | Title (see page 209) | The descriptive name of the Ektron Catalog. The title appears as clickable link in the catalog when the Ektron Catalog is not shown. |

## EktronCatalogPart.BlogID Property

The ID of the blog in CMS400.NET.

**C#**

```
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron Blog Control")]
[Description("The ID of the blog in CMS400.NET.")]
public long BlogID;
```

**Visual Basic**

```
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron Blog Control")>
<Description("The ID of the blog in CMS400.NET.")>
Public Property BlogID() As long
```

## EktronCatalogPart.BoardID Property

The ID of the discussion board.

**C#**

```
[Browsable(true), DefaultValue(0)]
[Category("Ektron Discussion Boards Control")]
[Description("The ID of the discussion board.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BoardID;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(0)>
<Category("Ektron Discussion Boards Control")>
<Description("The ID of the discussion board.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As long
```

### EktronCatalogPart.CalendarDynamicParameter Property

To make this calendar dynamic, select calendar_id. When you do, this server control uses the calendar passed as a URL parameter.

**C#**

```
[TypeConverter(typeof(DynamicCalendarTypeConverter))]
[Browsable(true), DefaultValue("")]
[Category("Ektron Calendar Control")]
[Description("To make this calendar dynamic, select calendar_id. When you do, this server
control uses the calendar passed as a URL parameter.")]
public string CalendarDynamicParameter;
```

**Visual Basic**

```
<TypeConverter(typeof(DynamicCalendarTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron Calendar Control")>
<Description("To make this calendar dynamic, select calendar_id. When you do, this server
control uses the calendar passed as a URL parameter.")>
Public Property CalendarDynamicParameter() As string
```

### EktronCatalogPart.CollectionDynamicParameter Property

To make this collection dynamic, select id. When you do, this server control uses the collection passed as a URL parameter.

**C#**

```
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Browsable(true), DefaultValue("")]
[Category("Ektron Collection Control")]
[Description("To make this collection dynamic, select id. When you do, this server control
uses the collection passed as a URL parameter.")]
public string CollectionDynamicParameter;
```

**Visual Basic**

```
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron Collection Control")>
<Description("To make this collection dynamic, select id. When you do, this server control
uses the collection passed as a URL parameter.")>
Public Property CollectionDynamicParameter() As string
```

### EktronCatalogPart.ContentBlockDynamicParameter Property

Select id. When you do, this server control uses the content block passed as a URL parameter.

**C#**

```
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Browsable(true), DefaultValue("")]
```

```
[Category("Ektron Content Block")]
[Description("Select id. When you do, this server control uses the content block passed as a
URL parameter.")]
public string ContentBlockDynamicParameter;
```

**Visual Basic**

```
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron Content Block")>
<Description("Select id. When you do, this server control uses the content block passed as a
URL parameter.")>
Public Property ContentBlockDynamicParameter() As string
```

### EktronCatalogPart.DefaultCalendarID Property

The ID number of the calendar that appears where you inserted this server control.

**C#**

```
[Editor(typeof(CalendarUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron Calendar Control")]
[Description("The ID number of the calendar that appears where you inserted this server
control.")]
public long DefaultCalendarID;
```

**Visual Basic**

```
<Editor(typeof(CalendarUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron Calendar Control")>
<Description("The ID number of the calendar that appears where you inserted this server
control.")>
Public Property DefaultCalendarID() As long
```

### EktronCatalogPart.DefaultCollectionID Property

The ID of a collection that appears where you inserted this server control if no other collection is identified, or is not available.

**C#**

```
[Browsable(true), DefaultValue(0)]
[Editor(typeof(CollectionUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Category("Ektron Collection Control")]
[Description("The ID of a collection that appears where you inserted this server control if no
other collection is identified, or is not available.")]
public long DefaultCollectionID;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(0)>
<Editor(typeof(CollectionUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Category("Ektron Collection Control")>
<Description("The ID of a collection that appears where you inserted this server control if no
other collection is identified, or is not available.")>
Public Property DefaultCollectionID() As long
```

### EktronCatalogPart.DefaultContentID Property

The ID of a content block that appears where you insert this server control.

**C#**

```
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron Content Block")]
[Description("The ID of a content block that appears where you insert this server control.")]
public long DefaultContentID;
```

**Visual Basic**

```
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron Content Block")>
<Description("The ID of a content block that appears where you insert this server control.")>
Public Property DefaultContentID() As long
```

### EktronCatalogPart.DefaultFormID Property

The ID of a FormBlock (see page 246) that appears where you inserted this server control if no other form block is identified.

**C#**

```
[Editor(typeof(FormUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron FormBlock Control")]
[Description("The ID of a FormBlock that appears where you inserted this server control if no
other form block is identified.")]
public long DefaultFormID;
```

**Visual Basic**

```
<Editor(typeof(FormUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron FormBlock Control")>
<Description("The ID of a FormBlock that appears where you inserted this server control if no
other form block is identified.")>
Public Property DefaultFormID() As long
```

### EktronCatalogPart.EnableBlogControl Property

Set to True to allow a Blog (see page 24) Control to appear in the catalog.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Blog Control")]
[Description("Set to True to allow a Blog Control to appear in the catalog.")]
public bool EnableBlogControl;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Blog Control")>
<Description("Set to True to allow a Blog Control to appear in the catalog.")>
Public Property EnableBlogControl() As bool
```

### EktronCatalogPart.EnableCalendarControl Property

Set to true to allow a Calendar (see page 69) control to appear in the Catalog.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Calendar Control")]
[Description("Set to true to allow a Calendar control to appear in the Catalog.")]
public bool EnableCalendarControl;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Calendar Control")>
<Description("Set to true to allow a Calendar control to appear in the Catalog.")>
Public Property EnableCalendarControl() As bool
```

### EktronCatalogPart.EnableCollectionControl Property

Set to True to allow a Collection (see page 88) control to appear in the Catalog.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Collection Control")]
[Description("Set to True to allow a Collection control to appear in the Catalog.")]
public bool EnableCollectionControl;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Collection Control")>
<Description("Set to True to allow a Collection control to appear in the Catalog.")>
Public Property EnableCollectionControl() As bool
```

### EktronCatalogPart.EnableContentBlockControl Property

Set to True to allow a Content (see page 788) Block Control to appear in the catalog.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Content Block")]
[Description("Set to True to allow a Content Block Control to appear in the catalog.")]
public bool EnableContentBlockControl;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Content Block")>
<Description("Set to True to allow a Content Block Control to appear in the catalog.")>
Public Property EnableContentBlockControl() As bool
```

### EktronCatalogPart.EnableDiscussionBoardsControl Property

Set to True to allow a Forum (see page 256) (Discussion Board) control to appear in the catalog.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Discussion Boards Control")]
[Description("Set to True to allow a Forum (Discussion Board) control to appear in the
catalog.")]
public bool EnableDiscussionBoardsControl;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Discussion Boards Control")>
<Description("Set to True to allow a Forum (Discussion Board) control to appear in the
catalog.")>
Public Property EnableDiscussionBoardsControl() As bool
```

### EktronCatalogPart.EnableFormBlockControl Property

Set to True to allow a FormBlock (see page 246) Control to appear in the catalog.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron FormBlock Control")]
[Description("Set to True to allow a FormBlock Control to appear in the catalog.")]
public bool EnableFormBlockControl;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron FormBlock Control")>
<Description("Set to True to allow a FormBlock Control to appear in the catalog.")>
Public Property EnableFormBlockControl() As bool
```

### EktronCatalogPart.EnableListSummaryControl Property

Set to True to allow a ListSummary (see page 325) Control to appear in the catalog.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron ListSummary")]
[Description("Set to True to allow a ListSummary Control to appear in the catalog.")]
public bool EnableListSummaryControl;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron ListSummary")>
<Description("Set to True to allow a ListSummary Control to appear in the catalog.")>
Public Property EnableListSummaryControl() As bool
```

### EktronCatalogPart.EnableThreadedDiscussionsControl Property

Set to True to allow a Forum (see page 256) (Discussion Board) control to appear in the catalog.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Discussion Boards Control")]
[Description("Set to True to allow a Forum (Discussion Board) control to appear in the
catalog.")]
public bool EnableThreadedDiscussionsControl;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Discussion Boards Control")>
<Description("Set to True to allow a Forum (Discussion Board) control to appear in the
catalog.")>
Public Property EnableThreadedDiscussionsControl() As bool
```

### EktronCatalogPart.FolderID Property

The folder that contains the content blocks which appear in the list summary.

**C#**

```
[Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron ListSummary")]
[Description("The folder that contains the content blocks which appear in the list summary.")]
public long FolderID;
```

**Visual Basic**

```
<Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron ListSummary")>
<Description("The folder that contains the content blocks which appear in the list summary.")>
Public Property FolderID() As long
```

### EktronCatalogPart.FormBlockDynamicParameter Property

To make this form block dynamic, select ekfrm. When you do, this server control uses the form block passed as a URL parameter.

**C#**

```
[TypeConverter(typeof(DynamicFormIDTypeConverter))]
[Browsable(true), DefaultValue("")]
[Category("Ektron FormBlock Control")]
[Description("To make this form block dynamic, select ekfrm. When you do, this server control
```

```
uses the form block passed as a URL parameter.")]
public string FormBlockDynamicParameter;
```

**Visual Basic**

```
<TypeConverter(typeof(DynamicFormIDTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron FormBlock Control")>
<Description("To make this form block dynamic, select ekfrm. When you do, this server control
uses the form block passed as a URL parameter.")>
Public Property FormBlockDynamicParameter() As string
```

### EktronCatalogPart.Title Property

The descriptive name of the Ektron Catalog. The title appears as clickable link in the catalog when the Ektron Catalog is not shown.

**C#**

```
[Browsable(true), DefaultValue("Ektron Catalog Part")]
[Category("Ektron")]
[Description("The descriptive name of the Ektron Catalog. The title appears as clickable link
in the catalog when the Ektron Catalog is not shown.")]
public override string Title;
```

**Visual Basic**

```
<Browsable(true), DefaultValue("Ektron Catalog Part")>
<Category("Ektron")>
<Description("The descriptive name of the Ektron Catalog. The title appears as clickable link
in the catalog when the Ektron Catalog is not shown.")>
Public override Property Title() As string
```

## EktronCommunityCatalogPart Class

The EktronCommunityCatalogPart Server Control gets a list of Discussion Board (Forums) and Blogs from your site. Then, it allows site users to choose from any available Discussion Boards and Blogs and add them to a WebPartZone.

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

CommunityCatalogPart.cs

### EktronCommunityCatalogPart Properties

The properties of the EktronCommunityCatalogPart class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | EnableBlogs (see page 210) | When set to True, the option to add a blog control to a Web part is available. |

| | | |
|---|---|---|
| | EnableDiscussionBoards (see page 210) | When set to True, the option to add a Forum (see page 256) (Discussion Board) to a Web part is available. |
| | EnableForums (see page 210) | When set to True, the option to add a Forum (see page 256) (Discussion Board) to a Web part is available. |
| | Title (see page 210) | The descriptive name of the Ektron Community (see page 718) Catalog. The title appears as clickable link in the catalog when the Ektron Catalog is not shown. The default title is Ektron Community (see page 718) Catalog Part. |

### EktronCommunityCatalogPart.EnableBlogs Property

When set to True, the option to add a blog control to a Web part is available.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron")]
[Description("When set to True, the option to add a blog control to a Web part is available.")]
public bool EnableBlogs;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron")>
<Description("When set to True, the option to add a blog control to a Web part is available.")>
Public Property EnableBlogs() As bool
```

### EktronCommunityCatalogPart.EnableDiscussionBoards Property

When set to True, the option to add a Forum (see page 256) (Discussion Board) to a Web part is available.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron")]
[Description("When set to True, the option to add a Forum (Discussion Board) to a Web part is
available.")]
public bool EnableDiscussionBoards;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron")>
<Description("When set to True, the option to add a Forum (Discussion Board) to a Web part is
available.")>
Public Property EnableDiscussionBoards() As bool
```

### EktronCommunityCatalogPart.EnableForums Property

When set to True, the option to add a Forum (see page 256) (Discussion Board) to a Web part is available.

**C#**

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron")]
[Description("When set to True, the option to add a Forum (Discussion Board) to a Web part is
available.")]
public bool EnableForums;
```

**Visual Basic**

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron")>
<Description("When set to True, the option to add a Forum (Discussion Board) to a Web part is
available.")>
Public Property EnableForums() As bool
```

### EktronCommunityCatalogPart.Title Property

The descriptive name of the Ektron Community (see page 718) Catalog. The title appears as clickable link in the catalog when

the Ektron Catalog is not shown. The default title is Ektron Community (see page 718) Catalog Part.

**C#**

```
[Browsable(true), DefaultValue("Ektron Catalog Part")]
[Category("Ektron")]
[Description("The descriptive name of the Ektron Community Catalog. The title appears as
clickable link in the catalog when the Ektron Catalog is not shown. The default title is
Ektron Community Catalog Part.")]
public override string Title;
```

**Visual Basic**

```
<Browsable(true), DefaultValue("Ektron Catalog Part")>
<Category("Ektron")>
<Description("The descriptive name of the Ektron Community Catalog. The title appears as
clickable link in the catalog when the Ektron Catalog is not shown. The default title is
Ektron Community Catalog Part.")>
Public override Property Title() As string
```

## EktronWebPartZone Class

Similar to a Microsoft WebPartZone, the EktronWebPartZone server control provides a location for content to be displayed on a Web page. The difference between the two is formatting. The EktronWebPartZone is designed to work specifically with Ektron Server Controls (see page 7), while Microsoft WebPartZones are more generic.

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

EkWebPartZone.cs

### EktronWebPartZone Properties

The properties of the EktronWebPartZone class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CmsSitePath (see page 211) | DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time. |

### EktronWebPartZone.CmsSitePath Property

DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.

**C#**

```
[Browsable(false)]
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
```

```
[Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")]
public string CmsSitePath;
```

**Visual Basic**

```
<Browsable(false)>
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")>
Public Property CmsSitePath() As string
```

## EkWebControl Class

The base class from which all Ektron CMS400 Server Controls (see page 7) are derived.

**Class Hierarchy**

**C#**

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

**Visual Basic**

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

**File**

EkWebControl.cs

### EkWebControl.EkWebControl Constructor

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

**C#**

```
public EkWebControl();
```

**Visual Basic**

```
Public Sub New()
```

### EkWebControl Methods

The methods of the EkWebControl class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 213) | Method that can be overridden and used to grab data from the back end for each server control. |
| | GetEnumDescription (see page 213) | Gets the Description attribute value for the enum value |
| | GetFilenameFromPath (see page 213) | Returns the filename (with extention), given a path and filename. |
| | GetMessage (see page 214) | This is the overview for the GetMessage method overload. |
| | LocalizeString (see page 214) | 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 212) class. |

| | | |
|---|---|---|
| | RedirectFromLoginPage (see page 215) | Provides a mechanism to return from the login page to the previous page (if specified in web.config and on the querystring). |
| | RedirectToLoginPage (see page 215) | Redirects to the login page (if specified in web.config) and the user is not currently logged in. |
| | RegisterAllDmsMenuCssAndScripts (see page 215) | Used to register all necessary DmsMenu CSS and Scripts in one shot |
| | RegisterDmsMenuCSS (see page 215) | Used to register the DmsMenu.css File for the DmsMenu |
| | RegisterDmsMenuScript (see page 216) | Used to register the DMSMenu.js file |
| | RegisterJQueryScript (see page 216) | Used to register the Ektron.js file |
| | RegisterMembershipTabScript (see page 216) | Used to Register the Membership (see page 350) tab JS and CSS files |
| | RegisterThickBoxCSS (see page 216) | Used to Register the box.css file for ThickBox |
| | RegisterThickBoxScript (see page 216) | Used to Register the thickbox.js file |
| | SearchHeader (see page 216) | This methods searches for the literal-control containing the head-definition |
| | ToString (see page 217) | 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 217) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### EkWebControl.Fill Method

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

**C#**

```
public virtual void Fill();
```

**Visual Basic**

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

### EkWebControl.GetEnumDescription Method

Gets the Description attribute value for the enum value

**C#**

```
public static string GetEnumDescription(Enum value);
```

**Visual Basic**

```
Public static Function GetEnumDescription(value As Enum) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| value | |

### EkWebControl.GetFilenameFromPath Method

Returns the filename (with extention), given a path and filename.

**C#**

```
public static string GetFilenameFromPath(string path);
```

**Visual Basic**

```
Public static Function GetFilenameFromPath(path As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| path | |

## GetMessage Method

This is the overview for the GetMessage method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | EkWebControl.GetMessage (string) (see page 214) | Returns the localized value of the supplied string. |
| | EkWebControl.GetMessage (string, string) (see page 214) | Returns the localized value of the supplied string, or the default string if it doesn't exist in the resources. |

## EkWebControl.GetMessage Method (string)

Returns the localized value of the supplied string.

**C#**

```
public string GetMessage(string resourceString);
```

**Visual Basic**

```
Public Function GetMessage(resourceString As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| resourceString | |

## EkWebControl.GetMessage Method (string, string)

Returns the localized value of the supplied string, or the default string if it doesn't exist in the resources.

**C#**

```
public string GetMessage(string resourceString, string defaultString);
```

**Visual Basic**

```
Public Function GetMessage(resourceString As string, defaultString As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| resourceString | |
| defaultString | |

## EkWebControl.LocalizeString Method

Used to localize strings in the control properties window and at render time. Strings should be defined in resource files. Any property that the user would like to be localized to the sites default language should begin with the constant STR_LOCALIZE defined in the EkWebControl (see page 212) 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 (see page 639).

### EkWebControl.RedirectFromLoginPage Method

Provides a mechanism to return from the login page to the previous page (if specified in web.config and on the querystring).

**C#**

```
public bool RedirectFromLoginPage();
```

**Visual Basic**

```
Public Function RedirectFromLoginPage() As bool
```

**Returns**

bool

### EkWebControl.RedirectToLoginPage Method

Redirects to the login page (if specified in web.config) and the user is not currently logged in.

**C#**

```
public bool RedirectToLoginPage();
```

**Visual Basic**

```
Public Function RedirectToLoginPage() As bool
```

**Returns**

bool

### EkWebControl.RegisterAllDmsMenuCssAndScripts Method

Used to register all necessary DmsMenu CSS and Scripts in one shot

**C#**

```
public void RegisterAllDmsMenuCssAndScripts();
```

**Visual Basic**

```
Public Function RegisterAllDmsMenuCssAndScripts() As void
```

### EkWebControl.RegisterDmsMenuCSS Method

Used to register the DmsMenu.css File for the DmsMenu

**C#**

```
public void RegisterDmsMenuCSS();
```

**Visual Basic**

```
Public Function RegisterDmsMenuCSS() As void
```

### EkWebControl.RegisterDmsMenuScript Method

Used to register the DMSMenu.js file

**C#**

```
public void RegisterDmsMenuScript();
```

**Visual Basic**

```
Public Function RegisterDmsMenuScript() As void
```

### EkWebControl.RegisterJQueryScript Method

Used to register the Ektron.js file

**C#**

```
public void RegisterJQueryScript();
```

**Visual Basic**

```
Public Function RegisterJQueryScript() As void
```

### EkWebControl.RegisterMembershipTabScript Method

Used to Register the Membership (see page 350) tab JS and CSS files

**C#**

```
public void RegisterMembershipTabScript();
```

**Visual Basic**

```
Public Function RegisterMembershipTabScript() As void
```

### EkWebControl.RegisterThickBoxCSS Method

Used to Register the box.css file for ThickBox

**C#**

```
public void RegisterThickBoxCSS();
```

**Visual Basic**

```
Public Function RegisterThickBoxCSS() As void
```

### EkWebControl.RegisterThickBoxScript Method

Used to Register the thickbox.js file

**C#**

```
public void RegisterThickBoxScript();
```

**Visual Basic**

```
Public Function RegisterThickBoxScript() As void
```

### EkWebControl.SearchHeader Method

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

**C#**

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

**Visual Basic**

```
Public Function SearchHeader(ByRef index As Integer) 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 new virtual string Wrap(string Html);
```

### Visual Basic

```
Public new 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 218) | DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in Visual Studio.NET. |
| | CmsSitePath (see page 218) | DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time. |
| | Error (see page 218) | An exception caught in the rendering of this Server Control. |
| | Hide (see page 219) | 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 219) | 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 219) | Set a language for viewing content blocks, collections, etc. in different languages. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser). |
| | LanguageID (see page 220) | This property lets a designer view content blocks, collections etc in different languages. |
| | loggedInUserID (see page 220) | 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 220) | DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in. |
| | LogLevel (see page 220) | "Allows control-level filtering of the logged events, from verbose to none |

| | | |
|---|---|---|
| | SuppressWrapperTags (see page 220) | Suppresses the output of span/div tags around the control. |
| | Text (see page 221) | 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 221) | Allows a developer to specify control's tag. Default is assumed "". |

### EkWebControl.Authenticated Property

DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in Visual Studio.NET.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(LoginUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(false)]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Description("DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and
permits them to login through a login dialog in VisualStudio.NET.")]
public string Authenticated;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(LoginUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(false)>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Description("DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and
permits them to login through a login dialog in VisualStudio.NET.")>
Public Property Authenticated() As string
```

### EkWebControl.CmsSitePath Property

DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.

**C#**

```
[Browsable(false)]
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")]
public string CmsSitePath;
```

**Visual Basic**

```
<Browsable(false)>
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")>
Public Property CmsSitePath() As string
```

### EkWebControl.Error Property

An exception caught in the rendering of this Server Control.

**C#**

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

```
[DefaultValue("")]
[Description("An exception caught in the rendering of this Server Control.")]
public string Error;
```

**Visual Basic**

```
<Bindable(true)>
<DefaultValue("")>
<Description("An exception caught in the rendering of this Server Control.")>
Public Property Error() As string
```

### EkWebControl.Hide Property

Alias for the 'Visible' property. Visible is set to mirror this value. This property is used to hide output in design time, whereas Visible does not hide output in design time.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Alias for the 'Visible' property. Visible is set to mirror this value. This
property is used to hide output in design time, whereas Visible does not hide output in design
time.")]
public bool Hide;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Alias for the 'Visible' property. Visible is set to mirror this value. This
property is used to hide output in design time, whereas Visible does not hide output in design
time.")>
Public Property Hide() As bool
```

### EkWebControl.IsLoggedIn Property

DESIGN TIME ONLY: Read only property that lets a developer know if the user is currently log into the site as a user.

**C#**

```
public bool IsLoggedIn;
```

**Visual Basic**

```
Public Property IsLoggedIn() As bool
```

### EkWebControl.Language Property

Set a language for viewing content blocks, collections, etc. in different languages. The language property shows results in design-time (in Visual Studio) and at run-time (in a browser).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(1033)]
[Description("Set a language for viewing content blocks, collections, etc. in different
languages. The language property shows results in design-time (in Visual Studio) and at
run-time (in a browser).")]
[Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int Language;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(1033)>
<Description("Set a language for viewing content blocks, collections, etc. in different
languages. The language property shows results in design-time (in Visual Studio) and at
run-time (in a browser).")>
```

```
<Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property Language() As Integer
```

### EkWebControl.LanguageID Property

This property lets a designer view content blocks, collections etc in different languages.

**C#**

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

**Visual Basic**

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

### EkWebControl.loggedInUserID Property

DESIGN TIME ONLY: Read only property that lets a developer know the user ID of the user who is currently logged in.

**C#**

```
public long loggedInUserID;
```

**Visual Basic**

```
Public Property loggedInUserID() As long
```

### EkWebControl.loggedInUserName Property

DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in.

**C#**

```
public string loggedInUserName;
```

**Visual Basic**

```
Public Property loggedInUserName() As string
```

### EkWebControl.LogLevel Property

"Allows control-level filtering of the logged events, from verbose to none

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(System.Diagnostics.TraceLevel))]
[Description("Allows control-level filtering of the logged events, from verbose to none")]
[DefaultValue("Error")]
public System.Diagnostics.TraceLevel LogLevel;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(System.Diagnostics.TraceLevel))>
<Description("Allows control-level filtering of the logged events, from verbose to none")>
<DefaultValue("Error")>
Public Property LogLevel() As System.Diagnostics.TraceLevel
```

### EkWebControl.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

**C#**

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

```
[DefaultValue(false)]
[Description("Suppresses the output of span/div tags around the control.")]
public virtual bool SuppressWrapperTags;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Suppresses the output of span/div tags around the control.")>
Public virtual Property SuppressWrapperTags() As bool
```

### EkWebControl.Text Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to treat the server control as a Label, taking information
from the Control and painting it on top of the Control for display. Leave blank for default
display.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual string Text;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to treat the server control as a Label, taking information
from the Control and painting it on top of the Control for display. Leave blank for default
display.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property Text() As string
```

### EkWebControl.WrapTag Property

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

**C#**

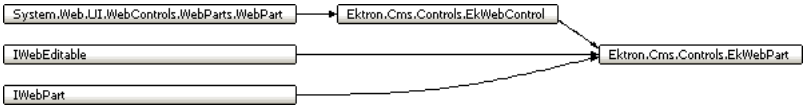
```
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("span")]
[TypeConverter(typeof(WrapTagTypeConverter))]
[Description("Allows a developer to specify control's tag. Default is assumed '<span>'.")]
public virtual string WrapTag;
```

**Visual Basic**

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

# EkWebPart Class

### Class Hierarchy




### C#

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

### Visual Basic

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

### File

EkWebPart.cs

### Description

This is class Ektron.Cms.Controls.EkWebPart.

## EkWebPart Methods

The methods of the EkWebPart class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | CreateEditorParts (see page 222) | This is CreateEditorParts, a member of class EkWebPart. |

## EkWebPart.CreateEditorParts Method

### C#

```
public EditorPartCollection CreateEditorParts();
```

### Visual Basic

```
Public Function CreateEditorParts() As EditorPartCollection
```

### Description

This is CreateEditorParts, a member of class EkWebPart.

## EkWebPart Properties

The properties of the EkWebPart class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | CatalogIconImageUrl (see page 223) | This is CatalogIconImageUrl, a member of class EkWebPart. |
| | Description (see page 223) | This is Description, a member of class EkWebPart. |
| | Subtitle (see page 223) | This is Subtitle, a member of class EkWebPart. |
| | Title (see page 223) | This is Title, a member of class EkWebPart. |
| | TitleIconImageUrl (see page 223) | This is TitleIconImageUrl, a member of class EkWebPart. |

| | | |
|---|---|---|
| | TitleUrl (see page 224) | This is TitleUrl, a member of class EkWebPart. |
| | WebBrowsableObject (see page 224) | This is WebBrowsableObject, a member of class EkWebPart. |

### EkWebPart.CatalogIconImageUrl Property

**C#**

```
public string CatalogIconImageUrl;
```

**Visual Basic**

```
Public Property CatalogIconImageUrl() As string
```

**Description**

This is CatalogIconImageUrl, a member of class EkWebPart.

### EkWebPart.Description Property

**C#**

```
public string Description;
```

**Visual Basic**

```
Public Property Description() As string
```

**Description**

This is Description, a member of class EkWebPart.

### EkWebPart.Subtitle Property

**C#**

```
public string Subtitle;
```

**Visual Basic**

```
Public Property Subtitle() As string
```

**Description**

This is Subtitle, a member of class EkWebPart.

### EkWebPart.Title Property

**C#**

```
public string Title;
```

**Visual Basic**

```
Public Property Title() As string
```

**Description**

This is Title, a member of class EkWebPart.

### EkWebPart.TitleIconImageUrl Property

**C#**

```
public string TitleIconImageUrl;
```

**Visual Basic**

```
Public Property TitleIconImageUrl() As string
```

**Description**

This is TitleIconImageUrl, a member of class EkWebPart.

### EkWebPart.TitleUrl Property

**C#**

```
public string TitleUrl;
```

**Visual Basic**

```
Public Property TitleUrl() As string
```

**Description**

This is TitleUrl, a member of class EkWebPart.

### EkWebPart.WebBrowsableObject Property

**C#**

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

**Visual Basic**

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

**Description**

This is WebBrowsableObject, a member of class EkWebPart.

## EkXsltBase Class

Summary description for EkXsltBase.

**Class Hierarchy**

**C#**

```
public abstract class EkXsltBase : EkWebControl;
```

**Visual Basic**

```
Public abstract Class EkXsltBase
Inherits EkWebControl
```

**File**

EkXsltBase.cs

### EkXsltBase.EkXsltBase Constructor

Constructor for EkXlstBase

**C#**

```
public EkXsltBase();
```

**Visual Basic**

```
Public Sub New()
```

## EkXsltBase Properties

The properties of the EkXsltBase class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | CustomXml (see page 225) | This property provides a mechanism to inject custom XML into the generated XML before being processed by the XSLT. |
| | DisplayXslt (see page 225) | The Xslt to use to render the control (optional). |
| | EnableErrorLogging (see page 225) | Enables or disables logging the error to the event log. |
| | XmlDoc (see page 226) | XmlDocument used in this control. |
| | XsltDoc (see page 226) | Xslt used in this control. |

## EkXsltBase.CustomXml Property

This property provides a mechanism to inject custom XML into the generated XML before being processed by the XSLT.

**C#**

```
[Browsable(false)]
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("This property provides a mechanism to inject custom XML into the generated XML
before being processed by the XSLT.")]
[DefaultValue("")]
public string CustomXml;
```

**Visual Basic**

```
<Browsable(false)>
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("This property provides a mechanism to inject custom XML into the generated XML
before being processed by the XSLT.")>
<DefaultValue("")>
Public Property CustomXml() As string
```

## EkXsltBase.DisplayXslt Property

The Xslt to use to render the control (optional).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltTypeConverter))]
[DefaultValue("ecmNavigation")]
[Personalizable()]
public virtual string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltTypeConverter))>
<DefaultValue("ecmNavigation")>
<Personalizable()>
Public virtual Property DisplayXslt() As string
```

## EkXsltBase.EnableErrorLogging Property

Enables or disables logging the error to the event log.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Enables or disables logging the error to the event log.")]
[DefaultValue("true")]
[Obsolete("EnableErrorLogging property is obsolete, use LogLevel", false)]
public bool EnableErrorLogging;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Enables or disables logging the error to the event log.")>
<DefaultValue("true")>
<Obsolete("EnableErrorLogging property is obsolete, use LogLevel", false)>
Public Property EnableErrorLogging() As bool
```

### EkXsltBase.XmlDoc Property

XmlDocument used in this control.

**C#**

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

**Visual Basic**

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

### EkXsltBase.XsltDoc Property

Xslt used in this control.

**C#**

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

**Visual Basic**

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

## ExplorerDragDrop Class

Defines the Explorer Drag Drop control, used to upload assets via Desktop drag drop to the CMS.

**Class Hierarchy**

**C#**

```
[DefaultProperty("Path")]
[ToolboxData("<{0}:ExplorerView runat=\"server\"></{0}:ExplorerView>")]
[AspNetHostingPermission(SecurityAction.Demand, Level = AspNetHostingPermissionLevel.Minimal)]
[System.ComponentModel.ToolboxItem(false)]
public class ExplorerDragDrop : WebControl;
```

**Visual Basic**

```
<DefaultProperty("Path")>
<ToolboxData("<{0}:ExplorerView runat=\"server\"></{0}:ExplorerView>")>
<AspNetHostingPermission(SecurityAction.Demand, Level = AspNetHostingPermissionLevel.Minimal)>
<System.ComponentModel.ToolboxItem(false)>
```

```
Public Class ExplorerDragDrop
Inherits WebControl
```

**File**

ExplorerDragDrop.cs

## ExplorerDragDrop Properties

The properties of the ExplorerDragDrop class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AssetID (see page 227) | Ektron CMS Asset (see page 789) ID. |
| | ContentLanguage (see page 227) | Ektron CMS Content (see page 788) Language. |
| | FolderID (see page 227) | Ektron CMS folder ID. |
| | IsImage (see page 228) | The ID of the taxonomy where assets are added. |
| | OverrideExtension (see page 228) | The ID of the taxonomy where assets are added. |
| | TaxonomyId (see page 228) | Ektron CMS Taxonomy (see page 842) Id. |

## ExplorerDragDrop.AssetID Property

Ektron CMS Asset (see page 789) ID.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("-1")]
[Localizable(true)]
public string AssetID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("-1")>
<Localizable(true)>
Public Property AssetID() As string
```

## ExplorerDragDrop.ContentLanguage Property

Ektron CMS Content (see page 788) Language.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(1033)]
[Localizable(true)]
public int ContentLanguage;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(1033)>
<Localizable(true)>
Public Property ContentLanguage() As Integer
```

## ExplorerDragDrop.FolderID Property

Ektron CMS folder ID.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(1)]
[Localizable(true)]
public long FolderID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(1)>
<Localizable(true)>
Public Property FolderID() As long
```

### ExplorerDragDrop.IsImage Property

The ID of the taxonomy where assets are added.

**C#**

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("Used to specify only images.")]
public int IsImage;
```

**Visual Basic**

```
<Category("Ektron")>
<DefaultValue("0")>
<Description("Used to specify only images.")>
Public Property IsImage() As Integer
```

### ExplorerDragDrop.OverrideExtension Property

The ID of the taxonomy where assets are added.

**C#**

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("Extension the uploaded file should be.")]
public string OverrideExtension;
```

**Visual Basic**

```
<Category("Ektron")>
<DefaultValue("0")>
<Description("Extension the uploaded file should be.")>
Public Property OverrideExtension() As string
```

### ExplorerDragDrop.TaxonomyId Property

Ektron CMS Taxonomy (see page 842) Id.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(-1)]
[Localizable(true)]
public long TaxonomyId;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(-1)>
<Localizable(true)>
```

```
Public Property TaxonomyId() As long
```

## Favorites Class

Summary description for Favorites

### Class Hierarchy

**C#**

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

**Visual Basic**

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

**File**

Favorites.cs

### Favorites Methods

The methods of the Favorites class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 229) | Populates the data for the Favorites (see page 229). |
| | ToString (see page 229) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 229) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### Favorites.Fill Method

Populates the data for the Favorites (see page 229).

**C#**

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

**Visual Basic**

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

### Favorites.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

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

**Visual Basic**

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

### Favorites.Wrap Method

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

**C#**

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

**Visual Basic**

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

**Parameters**

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

**Returns**

## Favorites Properties

The properties of the Favorites class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 231) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultUserID (see page 231) | The ID for the community group that the control will use. |
| | DisplayMode (see page 231) | Select the way this control displays community group information. Choices are:<br>**List** - shows a list of Favorites (see page 229) sorted in alphabetical order.<br>**Directory (see page 189)** - this selection allows users to group their favorites by folders. When this selection is chosen, the following menu items are added to the control:<br>• **Add Folder (see page 652)** - allows a user to add a sub folder.<br>• **Move** - allows a user to move content into a folder. This item only appears when there is a folder. |
| | DisplayXslt (see page 232) | Select the type of review to display, or enter the path to a custom XSLT. (optional). |
| | DynamicParameter (see page 232) | Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. Leave blank to always display the list summary. |
| | FavoritesTitle (see page 232) | The title to show at the top of the control. |
| | FavoritesUriTitle (see page 233) | The title to show at the top of the control. |
| | Link (see page 233) | Enter a link to the content Web page. This allows a user to click a link in the Favorites (see page 229) control and be taken to the content. There are three variables used within the link that represent the content's ID, the language and the content's title.<br>• **{0}** - represents the content's ID.<br>• **{1}** - represents the content's language.<br>• **{2}** - represents the content's title.<br>You need to have all three variables in the link. The Web form can be relative or absolute. Below is an example.<br>contentpage.aspx?cid={0}&lang={1}&title={2} |
| | LinkTarget (see page 233) | Determines the type of window that appears on this Web form when a user clicks an item in the Favorites (see page 229) control. The default is _self. |
| | MarkupLanguage (see page 234) | Enter name of a markup file that manages the display of this server control. |
| | MaxResults (see page 234) | Maximum number of results to be returned. If set to zero, there is no limit. |
| | Stylesheet (see page 234) | Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | XmlDoc (see page 235) | Returns the Xslt object for the collection. |

### Favorites.CacheInterval Property

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

**C#**

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

**Visual Basic**

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

### Favorites.DefaultUserID Property

The ID for the community group that the control will use.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID for the user that the control will use.")]
public long DefaultUserID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID for the user that the control will use.")>
Public Property DefaultUserID() As long
```

### Favorites.DisplayMode Property

Select the way this control displays community group information. Choices are:

**List** - shows a list of Favorites (see page 229) sorted in alphabetical order.

**Directory (see page 189)** - this selection allows users to group their favorites by folders. When this selection is chosen, the following menu items are added to the control:

- **Add Folder (see page 652)** - allows a user to add a sub folder.
- **Move** - allows a user to move content into a folder. This item only appears when there is a folder.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Directory), Description("This
property enables the user to show the favorites either in the List or Directory display mode")]
public eDisplayMode DisplayMode;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Directory), Description("This
property enables the user to show the favorites either in the List or Directory display mode")>
Public Property DisplayMode() As eDisplayMode
```

### Favorites.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

**C#**

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

**Visual Basic**

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

### Favorites.DynamicParameter Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. Leave blank to always display the list summary.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. Leave blank to always use the default ID.")]
[Personalizable()]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. Leave blank to always use the default ID.")>
<Personalizable()>
Public Property DynamicParameter() As string
```

### Favorites.FavoritesTitle Property

The title to show at the top of the control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("Favorites"), Description("The title to show
for the Content at the top of the control.")]
public string FavoritesTitle;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("Favorites"), Description("The title to show
for the Content at the top of the control.")>
Public Property FavoritesTitle() As string
```

### Favorites.FavoritesUriTitle Property

The title to show at the top of the control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("Favorites"), Description("The title to show
for the Links at the top of the control.")]
public string FavoritesUriTitle;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("Favorites"), Description("The title to show
for the Links at the top of the control.")>
Public Property FavoritesUriTitle() As string
```

### Favorites.Link Property

Enter a link to the content Web page. This allows a user to click a link in the Favorites (see page 229) control and be taken to the content. There are three variables used within the link that represent the content's ID, the language and the content's title.

- **{0}** - represents the content's ID.
- **{1}** - represents the content's language.
- **{2}** - represents the content's title.

You need to have all three variables in the link. The Web form can be relative or absolute. Below is an example.

contentpage.aspx?cid={0}&lang={1}&title={2}

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"), Description("The favorite
link. {0} is the favorite ID, {1} is the favorite language, and {2} is the favorite title.") ]
public string Link;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"), Description("The favorite
link. {0} is the favorite ID, {1} is the favorite language, and {2} is the favorite title.") >
Public Property Link() As string
```

### Favorites.LinkTarget Property

Determines the type of window that appears on this Web form when a user clicks an item in the Favorites (see page 229) control. The default is _self.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self),
Description("Determines the type of window that appears on this Web form when a user clicks an
item in the Favorites control. The default is _self.") ]
public ItemLinkTargets LinkTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self),
Description("Determines the type of window that appears on this Web form when a user clicks an
item in the Favorites control. The default is _self.") >
Public Property LinkTarget() As ItemLinkTargets
```

### Favorites.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

### Favorites.MaxResults Property

Maximum number of results to be returned. If set to zero, there is no limit.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

**Visual Basic**

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

### Favorites.Stylesheet Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Specify the path to a style sheet for use with the sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Specify the path to a style sheet for use with the sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<Personalizable()>
Public Property Stylesheet() As string
```

### Favorites.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

## FlexMenu Class

The FlexMenu server control displays a menu on a Web form. Although it looks and acts like a Smart Menu (see page 358), its architecture is substantially different. Specifically, a FlexMenu creates XML. So, you can modify its behavior using an XSLT file, and change its appearance using a cascading style sheet (.css) file.

**Class Hierarchy**

**C#**

```
public class FlexMenu : Ektron.Cms.Controls.Menu;
```

**Visual Basic**

```
Public Class FlexMenu
Inherits Ektron.Cms.Controls.Menu
```

**File**

FlexMenu.cs

### FlexMenu Methods

The methods of the FlexMenu class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 235) | Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters. |
| | SetDirty (see page 235) | Marks the Control to call the database since output-dependent parameters have been changed. |

### FlexMenu.Fill Method

Calls up the database and updates the contents of the Menu if the data is out of sync with the specified parameters.

**C#**

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

**Visual Basic**

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

### FlexMenu.SetDirty Method

Marks the Control to call the database since output-dependent parameters have been changed.

**C#**

```
public new void SetDirty();
```

**Visual Basic**

```
Public new Function SetDirty() As void
```

## FlexMenu Properties

The properties of the FlexMenu class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AutoCollapseBranches (see page 236) | Set to True to automatically close other menu branches when a new one is opened. |
| | CacheInterval (see page 237) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DisplayXslt (see page 237) | The Xslt used to render the Menu (see page 358). |
| | EnableAjax (see page 237) | Set to true to enable Ajax, which only downloads sub-menus as needed. |
| | EnableMouseOverPopUp (see page 238) | Set to True and the submenus appear as soon as the cursor moves over them. |
| | EnableSmartOpen (see page 238) | When true, menu will auto-open to show matching IDs or folder associations. |
| | EnableSubmenuLinkPathMatching (see page 238) | Set to enable auto-submenu opening based on link match (if all else fails). |
| | EnableSubmenuTemplateMatching (see page 239) | Set to enable auto-submenu opening based on a template filename match (if all else fails). |
| | MasterControlId (see page 239) | If this control is a slave Flex menu, enter the ID of the master Flex menu control. Otherwise, leave blank. |
| | MenuDepth (see page 239) | To let site visitors browse through all menu levels, enter zero (0). To restrict site visitors to a menu level, enter the number of the lowest level. |
| | StartCollapsed (see page 240) | Set to True to initially render menu with submenus closed. |
| | StartLevel (see page 240) | Enter a number to indicate the level at which you want this menu to display when it first appears. To begin the menu display at the root level, enter zero (0). |
| | Stylesheet (see page 240) | The Style-Sheet to use for rendering the Menu (see page 358) (optional). |
| | UseAliasRealId (see page 241) | If system id using manual aliasing, and the aliased menu item does not get selected due to external link that generates a URL which contains the real path and ID, setting this true will force first matching menu item ID to be selected. |
| | XmlDoc (see page 241) | XmlDocument used in this control. |
| | XsltDoc (see page 241) | Xslt used in this control. |

## FlexMenu.AutoCollapseBranches Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to automatically close other menu branches when a new one is
opened.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool AutoCollapseBranches;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to automatically close other menu branches when a new one is
opened.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property AutoCollapseBranches() As bool
```

### FlexMenu.CacheInterval Property

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

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(600.0)]
public override double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(600.0)>
Public override Property CacheInterval() As double
```

### FlexMenu.DisplayXslt Property

The Xslt used to render the Menu (see page 358).

**C#**

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

**Visual Basic**

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

### FlexMenu.EnableAjax Property

Set to true to enable Ajax, which only downloads sub-menus as needed.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to true to enable Ajax, which only downloads sub-menus as needed.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool EnableAjax;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to true to enable Ajax, which only downloads sub-menus as needed.")>
```

```
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property EnableAjax() As bool
```

### FlexMenu.EnableMouseOverPopUp Property

Set to True and the submenus appear as soon as the cursor moves over them.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True and the submenus appear as soon as the cursor moves over them.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool EnableMouseOverPopUp;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True and the submenus appear as soon as the cursor moves over them.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property EnableMouseOverPopUp() As bool
```

### FlexMenu.EnableSmartOpen Property

When true, menu will auto-open to show matching IDs or folder associations.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("When true, menu will auto-open to show matching IDs or folder associations.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool EnableSmartOpen;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("When true, menu will auto-open to show matching IDs or folder associations.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property EnableSmartOpen() As bool
```

### FlexMenu.EnableSubmenuLinkPathMatching Property

Set to enable auto-submenu opening based on link match (if all else fails).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to enable auto-submenu opening based on link match (if all else fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuLinkPathMatching;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to enable auto-submenu opening based on link match (if all else fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
```

```
Public Property EnableSubmenuLinkPathMatching() As bool
```

### FlexMenu.EnableSubmenuTemplateMatching Property

Set to enable auto-submenu opening based on a template filename match (if all else fails).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to enable auto-submenu opening based on a template filename match (if all
else fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuTemplateMatching;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to enable auto-submenu opening based on a template filename match (if all
else fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property EnableSubmenuTemplateMatching() As bool
```

### FlexMenu.MasterControlId Property

If this control is a slave Flex menu, enter the ID of the master Flex menu control. Otherwise, leave blank.

**C#**

```
[Description("If this control is a slave Flex menu, enter the ID of the master Flex menu
control. Otherwise, leave blank.")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
[DefaultValue("")]
public string MasterControlId;
```

**Visual Basic**

```
<Description("If this control is a slave Flex menu, enter the ID of the master Flex menu
control. Otherwise, leave blank.")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
<DefaultValue("")>
Public Property MasterControlId() As string
```

### FlexMenu.MenuDepth Property

To let site visitors browse through all menu levels, enter zero (0). To restrict site visitors to a menu level, enter the number of the lowest level.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("To let site visitors browse through all menu levels, enter zero (0). To restrict
site visitors to a menu level, enter the number of the lowest level.")]
[TypeConverter(typeof(Int32Converter))]
[DefaultValue(0)]
public int MenuDepth;
```

**Visual Basic**

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

```
<Description("To let site visitors browse through all menu levels, enter zero (0). To restrict
site visitors to a menu level, enter the number of the lowest level.")>
<TypeConverter(typeof(Int32Converter))>
<DefaultValue(0)>
Public Property MenuDepth() As Integer
```

### FlexMenu.StartCollapsed Property

Set to True to initially render menu with submenus closed.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to initially render menu with submenus closed.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool StartCollapsed;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to initially render menu with submenus closed.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property StartCollapsed() As bool
```

### FlexMenu.StartLevel Property

Enter a number to indicate the level at which you want this menu to display when it first appears. To begin the menu display at the root level, enter zero (0).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Enter a number to indicate the level at which you want this menu to display when
it first appears. To begin the menu display at the root level, enter zero (0).")]
[TypeConverter(typeof(Int32Converter))]
[DefaultValue(1)]
public int StartLevel;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Enter a number to indicate the level at which you want this menu to display when
it first appears. To begin the menu display at the root level, enter zero (0).")>
<TypeConverter(typeof(Int32Converter))>
<DefaultValue(1)>
Public Property StartLevel() As Integer
```

### FlexMenu.Stylesheet Property

The Style-Sheet to use for rendering the Menu (see page 358) (optional).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Style-Sheet to use for rendering the Menu.")]
[TypeConverter(typeof(StringConverter))]
[DefaultValue("")]
public override string Stylesheet;
```

**Visual Basic**

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

```
<Category("Ektron")>
<Description("The Style-Sheet to use for rendering the Menu.")>
<TypeConverter(typeof(StringConverter))>
<DefaultValue("")>
Public override Property Stylesheet() As string
```

### FlexMenu.UseAliasRealId Property

If system id using manual aliasing, and the aliased menu item does not get selected due to external link that generates a URL which contains the real path and ID, setting this true will force first matching menu item ID to be selected.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("Not normally needed (default: False), but available for special cases when both
alias and non-alias references to the same menu item must be allowed.")]
[DefaultValue("false")]
public bool UseAliasRealId;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("Not normally needed (default: False), but available for special cases when both
alias and non-alias references to the same menu item must be allowed.")>
<DefaultValue("false")>
Public Property UseAliasRealId() As bool
```

### FlexMenu.XmlDoc Property

XmlDocument used in this control.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

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

### FlexMenu.XsltDoc Property

Xslt used in this control.

**C#**

```
public override XmlDocument XsltDoc;
```

**Visual Basic**

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

## FolderBreadcrumb Class

The FolderBreadcrumb server control displays sitemap breadcrumbs. This type of trail is typically made up of the folder path for the content. It is important to note that the FolderBreadcrumb server control does not simply read your folder structure and display its path. It reads breadcrumb information set in the Workarea's folder properties for each folder. Administrators define a folder's breadcrumbs on the Breadcrumb tab located in the Workarea folder properties. *See Also:* The Administrator Manual > "Managing Content (see page 788)" > "Managing Content (see page 788) Folders" > "Breadcrumbs"

For additional information on this server control, see the Developer Manual section "Introduction to Ektron CMS400.NET Server Controls (see page 7)" > "FolderBreadcrumb Server Control"

### Class Hierarchy

### C#

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

### Visual Basic

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

### File

FolderBreadcrumb.cs

## FolderBreadcrumb Methods

The methods of the FolderBreadcrumb class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 242) | Method that can be overridden and used to grab data from the back end for each server control. |

## FolderBreadcrumb.Fill Method

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

### C#

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

### Visual Basic

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

## FolderBreadcrumb Properties

The properties of the FolderBreadcrumb class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DefaultContentID (see page 243) | Gets or sets a Content (see page 788) ID and displays the breadcrumb trail for the folder in which the content resides. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (see page 652)] > Folder (see page 652) properties > Breadcrumb tab. |
| | DefaultFolderID (see page 243) | The folder ID for which you want the breadcrumb trail to display. If a DefaultContentID (see page 243) is given then it overrides this property. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (see page 652)] > Folder (see page 652) properties > Breadcrumb tab. |
| | DisplayStyle (see page 244) | Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal. |
| | DisplayXslt (see page 244) | Select the type of review to display, or enter the path to a custom XSLT. (optional). |
| | DynamicParameter (see page 244) | Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.) |

| | | |
|---|---|---|
| | LinkTarget (see page 245) | Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.<br>• **_Self** - opens link in the same window.<br>• **_Top** - opens link in a parent window.<br>• **_Blank** - opens link in a new window.<br>• **_Parent** - opens link in the parent frame. |
| | Mode (see page 245) | Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal.<br>• **Normal** = breadcrumb trail is hyperlinked<br>• **DisplayOnly** = breadcrumb trail is plain text |
| | separator (see page 245) | Enter one or more characters to separate the items in a breadcrumb trail. |
| | XmlDoc (see page 245) | Returns the Xslt object for the collection. |

### FolderBreadcrumb.DefaultContentID Property

Gets or sets a Content (see page 788) ID and displays the breadcrumb trail for the folder in which the content resides. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (see page 652)] > Folder (see page 652) properties > Breadcrumb tab.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or sets the Content ID and displays the breadcrumb trail for the folder in
which the content resides.")]
public long DefaultContentID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or sets the Content ID and displays the breadcrumb trail for the folder in
which the content resides.")>
Public Property DefaultContentID() As long
```

### FolderBreadcrumb.DefaultFolderID Property

The folder ID for which you want the breadcrumb trail to display. If a DefaultContentID (see page 243) is given then it overrides this property. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (see page 652)] > Folder (see page 652) properties > Breadcrumb tab.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The folder ID for which you want the breadcrumb trail to display. If a
DefaultContentID is given then it overrides this property.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long DefaultFolderID;
```

**Visual Basic**

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

```
<Description("The folder ID for which you want the breadcrumb trail to display. If a
DefaultContentID is given then it overrides this property.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property DefaultFolderID() As long
```

### FolderBreadcrumb.DisplayStyle Property

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

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(DisplayStyles.Horizontal),
Description("Indicate how to display the breadcrumb trail: horizontally or vertically. The
default is Horizontal."), ]
public new DisplayStyles DisplayStyle;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(DisplayStyles.Horizontal),
Description("Indicate how to display the breadcrumb trail: horizontally or vertically. The
default is Horizontal."), >
Public new Property DisplayStyle() As DisplayStyles
```

### FolderBreadcrumb.DisplayXslt Property

Select the type of review to display, or enter the path to a custom XSLT. (optional).

**C#**

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

**Visual Basic**

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

### FolderBreadcrumb.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 788) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)

**C#**

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

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
```

```
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

### FolderBreadcrumb.LinkTarget Property

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

- **_Self** - opens link in the same window.
- **_Top** - opens link in a parent window.
- **_Blank** - opens link in a new window.
- **_Parent** - opens link in the parent frame.

**C#**

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

**Visual Basic**

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

### FolderBreadcrumb.Mode Property

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

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

**C#**

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

**Visual Basic**

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

### FolderBreadcrumb.separator Property

Enter one or more characters to separate the items in a breadcrumb trail.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(" > "), Description("Enter one or more
characters to separate the items in a breadcrumb trail.") ]
public string separator;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(" > "), Description("Enter one or more
characters to separate the items in a breadcrumb trail.") >
Public Property separator() As string
```

### FolderBreadcrumb.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

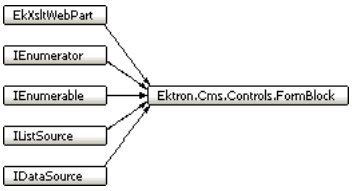
**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## FormBlock Class

The FormBlock server control displays a content block associated with a form.

**Class Hierarchy**




**C#**

```
[DefaultProperty("DefaultFormID"), ToolboxData("<{0}:FormBlock runat=server></{0}:FormBlock>")]
public class FormBlock : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource;
```

**Visual Basic**

```
<DefaultProperty("DefaultFormID"), ToolboxData("<{0}:FormBlock runat=server></{0}:FormBlock>")>
Public Class FormBlock
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
```

**File**

FormBlock.cs

### FormBlock Methods

The methods of the FormBlock class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | CopyTo (see page 247) | 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 247) | Overrideable method that can be used to grab data from the back end for each server control. |
| | GetEnumerator (see page 247) | Gets a System.Collections.IEnumerator for the collection. |
| | GetView (see page 247) | Gets the view data using a dataset. |
| | GetViewNames (see page 247) | Gets the view names. |
| | MoveNext (see page 248) | Moves up the index of the Collection (see page 88). Returns false if the end of the collection has been surpassed. |
| | Reset (see page 248) | Brings the index of the collection back to zero. Does not clear the collection. |
| | this (see page 248) | Index operator into a strongly typed dataset holding some common properties for a Collection (see page 88) item. |
| | ToString (see page 248) | 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 248) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### FormBlock.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

**C#**

```
public void CopyTo(Array array, int i);
```

**Visual Basic**

```
Public Function CopyTo(array As Array, i As Integer) As void
```

### FormBlock.Fill Method

Overrideable method that can be used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

### FormBlock.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

**C#**

```
public IEnumerator GetEnumerator();
```

**Visual Basic**

```
Public Function GetEnumerator() As IEnumerator
```

### FormBlock.GetView Method

Gets the view data using a dataset.

**C#**

```
public DataSourceView GetView(string viewName);
```

**Visual Basic**

```
Public Function GetView(viewName As string) As DataSourceView
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 88). 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 88) 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 249) | Set this value to True to add the validation to your main runat=server form. |
| | CacheInterval (see page 250) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | Count (see page 250) | Indicates the number of elements in your Collection (see page 88). |
| | Current (see page 250) | Gets the current object in the collection. |
| | DefaultFormID (see page 250) | Gets or Sets the Default FormID for this Control. |
| | DynamicParameter (see page 251) | Gets or sets the QueryString parameter to read a Form (see page 836) ID dynamically - Set to String.Empty if you wish to always display the default FormID (static.) |
| | EkItem (see page 251) | Gets the Ektron.Cms.Common.ContentBase for the current Form (see page 836) Block. |
| | Fields (see page 251) | Returns a collection of fields defined in the selected form. |
| | FormName (see page 252) | Set this value to the name of your main runat=server form. |
| | IncludeTags (see page 252) | Set to True to have this Ektron Form (see page 836) element to wrap its own Form (see page 836) tags around itself. |
| | IsSynchronized (see page 252) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | StartingFolderID (see page 253) | WEB PARTS ONLY: Allows a developer to set the starting folder for the personalization/collaboration selection dialog. |
| | SyncRoot (see page 253) | Gets an object that can be used to synchronize the collection. |

## FormBlock.AddValidation Property

Set this value to True to add the validation to your main runat=server form.

### C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set this value to True to add the validation to your main runat=server form.")]
[Personalizable()]
[Obsolete("Please remove form validation using the editor.")]
public bool AddValidation;
```

### Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set this value to True to add the validation to your main runat=server form.")>
<Personalizable()>
<Obsolete("Please remove form validation using the editor.")>
Public Property AddValidation() As bool
```

### FormBlock.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

### FormBlock.Count Property

Indicates the number of elements in your Collection (see page 88).

**C#**

```
[Browsable(false)]
public int Count;
```

**Visual Basic**

```
<Browsable(false)>
Public Property Count() As Integer
```

### FormBlock.Current Property

Gets the current object in the collection.

**C#**

```
[Browsable(false)]
public object Current;
```

**Visual Basic**

```
<Browsable(false)>
Public Property Current() As object
```

### FormBlock.DefaultFormID Property

Gets or Sets the Default FormID for this Control.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
```

```
[ParenthesizePropertyName(true)]
[Editor(typeof(FormUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("The default ID of the Form Block to display.")]
[Personalizable()]
public long DefaultFormID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<ParenthesizePropertyName(true)>
<Editor(typeof(FormUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("The default ID of the Form Block to display.")>
<Personalizable()>
Public Property DefaultFormID() As long
```

### FormBlock.DynamicParameter Property

Gets or sets the QueryString parameter to read a Form (see page 836) ID dynamically - Set to String.Empty if you wish to always display the default FormID (static.)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[TypeConverter(typeof(DynamicFormIDTypeConverter))]
[Description("Gets or sets the QueryString parameter to read a Form ID dynamically - Set to
String.Empty if you wish to always display the default FormID (static.)")]
[Personalizable()]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<TypeConverter(typeof(DynamicFormIDTypeConverter))>
<Description("Gets or sets the QueryString parameter to read a Form ID dynamically - Set to
String.Empty if you wish to always display the default FormID (static.)")>
<Personalizable()>
Public Property DynamicParameter() As string
```

### FormBlock.EkItem Property

Gets the Ektron.Cms.Common.ContentBase for the current Form (see page 836) Block.

**C#**

```
[Browsable(false)]
public ContentBase EkItem;
```

**Visual Basic**

```
<Browsable(false)>
Public Property EkItem() As ContentBase
```

### FormBlock.Fields Property

Returns a collection of fields defined in the selected form.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("List of fields defined for this form.")]
public FormFieldCollection Fields;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("List of fields defined for this form.")>
Public Property Fields() As FormFieldCollection
```

### FormBlock.FormName Property

Set this value to the name of your main runat=server form.

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Set this value to the name of your main runat=server form.")]
[Personalizable()]
public string FormName;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Set this value to the name of your main runat=server form.")>
<Personalizable()>
Public Property FormName() As string
```

### FormBlock.IncludeTags Property

Set to True to have this Ektron Form (see page 836) element to wrap its own Form (see page 836) tags around itself.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to have this Ektron Form element to wrap its own Form tags around
itself.")]
[Personalizable()]
public bool IncludeTags;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to have this Ektron Form element to wrap its own Form tags around
itself.")>
<Personalizable()>
Public Property IncludeTags() As bool
```

### FormBlock.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

```
[Browsable(false)]
public bool IsSynchronized;
```

**Visual Basic**

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

### FormBlock.StartingFolderID Property

WEB PARTS ONLY: Allows a developer to set the starting folder for the personalization/collaboration selection dialog.

**C#**

```
[Bindable(true)]
[Category("Ektron WebParts")]
[Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")]
[DefaultValue(0)]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable()]
public long StartingFolderID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron WebParts")>
<Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")>
<DefaultValue(0)>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable()>
Public Property StartingFolderID() As long
```

### FormBlock.SyncRoot Property

Gets an object that can be used to synchronize the collection.

**C#**

```
[Browsable(false)]
public object SyncRoot;
```

**Visual Basic**

```
<Browsable(false)>
Public Property SyncRoot() As object
```

## FormFieldItem Class

Defines a field within the form.

**Class Hierarchy**

**C#**

```
[DefaultProperty("FieldName")]
[ValidationProperty("Value")]
[TypeConverter(typeof(FormFieldItemConverter))]
public class FormFieldItem;
```

**Visual Basic**

```
<DefaultProperty("FieldName")>
<ValidationProperty("Value")>
<TypeConverter(typeof(FormFieldItemConverter))>
Public Class FormFieldItem
```

**File**

FormField.cs

## FormFieldItem Properties

The properties of the FormFieldItem class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Basetype (see page 254) | Gets the basic datatype of the field. |
| | ContentType (see page 254) | Gets the type of content allowed within the field. This is loosely associated with the datatype. |
| | DatalistItems (see page 254) | Gets the list of options for this field's value, if applicable, otherwise null/Nothing. |
| | Datatype (see page 255) | Gets the datatype of the field. |
| | DisplayName (see page 255) | Gets the name of the field to be displayed. |
| | FieldName (see page 255) | Gets the name of the field. |
| | Language (see page 256) | Gets the language code of the datalist. For example, en or en-US. |
| | Value (see page 256) | Gets or set the value of this field. |

## FormFieldItem.Basetype Property

Gets the basic datatype of the field.

**C#**

```
[Category("Data")]
[Description("Basic type of the field's data value, for example, 'text'.")]
public string Basetype;
```

**Visual Basic**

```
<Category("Data")>
<Description("Basic type of the field's data value, for example, 'text'.")>
Public Property Basetype() As string
```

**Remarks**

The possible values of the basic datatype vary, including, text, textbox, calendar, and checkbox.

## FormFieldItem.ContentType Property

Gets the type of content allowed within the field. This is loosely associated with the datatype.

**C#**

```
[Category("Data")]
[Description("Type of content allowed within the field")]
public string ContentType;
```

**Visual Basic**

```
<Category("Data")>
<Description("Type of content allowed within the field")>
Public Property ContentType() As string
```

**Remarks**

The possible values for content vary.

## FormFieldItem.DatalistItems Property

Gets the list of options for this field's value, if applicable, otherwise null/Nothing.

**C#**

```
[Browsable(false)]
[Category("Data")]
[Description("List of options for this field's value, if applicable, otherwise null/Nothing.")]
public List<FormFieldItemDatalistItem> DatalistItems;
```

**Visual Basic**

```
<Browsable(false)>
<Category("Data")>
<Description("List of options for this field's value, if applicable, otherwise null/Nothing.")>
Public Property DatalistItems() As List<FormFieldItemDatalistItem>
```

### FormFieldItem.Datatype Property

Gets the datatype of the field.

**C#**

```
[Category("Data")]
[Description("Type of the field's data value, generally conforming to W3C XML Schema types.")]
public string Datatype;
```

**Visual Basic**

```
<Category("Data")>
<Description("Type of the field's data value, generally conforming to W3C XML Schema types.")>
Public Property Datatype() As string
```

**Remarks**

The possible values of the datatype vary. Generally they conform to W3C XML Schema types such as 'string' and 'date'. Other values include 'selection' and 'choice'.

### FormFieldItem.DisplayName Property

Gets the name of the field to be displayed.

**C#**

```
[Category("Appearance")]
[Description("Name of the field suitable for display.")]
public string DisplayName;
```

**Visual Basic**

```
<Category("Appearance")>
<Description("Name of the field suitable for display.")>
Public Property DisplayName() As string
```

**Remarks**

The name is suitable for human reading.

### FormFieldItem.FieldName Property

Gets the name of the field.

**C#**

```
[Category("Data")]
[Description("Name of the field suitable for XML or programmatic processing.")]
public string FieldName;
```

**Visual Basic**

```
<Category("Data")>
<Description("Name of the field suitable for XML or programmatic processing.")>
Public Property FieldName() As string
```

**Remarks**

The name is suitable for machine processing, such as with an XML parser.

### FormFieldItem.Language Property

Gets the language code of the datalist. For example, en or en-US.

**C#**

```
[Category("Data")]
[Description("Language code of the datalist. For example, en or en-US.")]
public string Language;
```

**Visual Basic**

```
<Category("Data")>
<Description("Language code of the datalist. For example, en or en-US.")>
Public Property Language() As string
```

**Remarks**

The first two letters of the language code conform to ISO 639. The last two letters are an optional country code that conforms to ISO 3166.

### FormFieldItem.Value Property

Gets or set the value of this field.

**C#**

```
[Browsable(false)]
[Category("Data")]
[Description("The value of this field.")]
public string Value;
```

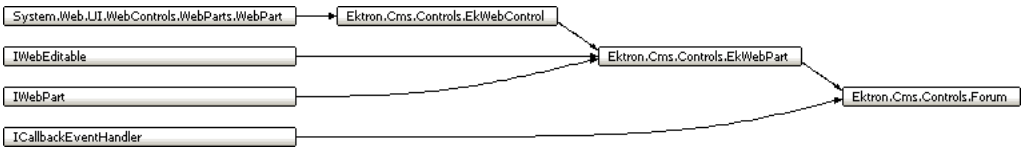
**Visual Basic**

```
<Browsable(false)>
<Category("Data")>
<Description("The value of this field.")>
Public Property Value() As string
```

## Forum Class

The Forum server control displays a discussion board on a Web page.

**Class Hierarchy**




**C#**

```
[DefaultProperty("BoardID"), ToolboxData("<{0}:Forum runat=server></{0}:Forum>")]
public class Forum : EkWebPart, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("BoardID"), ToolboxData("<{0}:Forum runat=server></{0}:Forum>")>
Public Class Forum
Inherits EkWebPart
Inherits ICallbackEventHandler
```

**File**

Forum.cs

## Forum Methods

The methods of the Forum class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 257) | Method that can be overridden and used to grab data from the back end for each server control. |
| | GetCallbackResult (see page 257) | Returns search results to the client callback function |
| | RaiseCallbackEvent (see page 257) | Processes the callback event by performing the search. |
| | ToString (see page 257) | 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 258) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## Forum.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## Forum.GetCallbackResult Method

Returns search results to the client callback function

**C#**

```
public string GetCallbackResult();
```

**Visual Basic**

```
Public Function GetCallbackResult() As string
```

**Returns**

String containing search results.

## Forum.RaiseCallbackEvent Method

Processes the callback event by performing the search.

**C#**

```
public void RaiseCallbackEvent(string eventArgs);
```

**Visual Basic**

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| eventArgs | Name-value pairs representing form fields passed by client callback function |

## Forum.ToString Method

Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

### Forum.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar |

**Returns**

Html string wrapped with a toolbar

### Forum Properties

The properties of the Forum class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AllowAnonymousPost (see page 260) | If set to false, will not allow anonymous post on the forum. Defaults to false. |
| | BoardID (see page 260) | The ID of the Discussion Board to display on this page if one is not defined in a query string parameter. |
| | CacheInterval (see page 260) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | CustomOrderBy (see page 261) | Order search results by a custom property. Set this property to the friendly name of the property defined in the Indexing Service. If an invalid property is specified, no search results are returned. If this is left empty, the OrderBy (see page 267) property is used to order search results. |
| | DynamicForumParameter (see page 261) | Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically. |
| | DynamicGroupParameter (see page 261) | Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically. |
| | DynamicParameter (see page 262) | Gets or Sets the QueryString parameter to read a Blog (see page 24) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 24) (static.) |
| | DynamicThreadParameter (see page 262) | Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically. |
| | EditorCSS (see page 262) | Set the stylesheet on the editor. |
| | EditorToolbar (see page 263) | Decide which buttons are available in the Editor. For additional information on the Toolbar Editor, See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Adding and Removing Toolbar Buttons From the Editor |
| | EnableForumQuickSelect (see page 263) | If set to true, will display a drop down list of all forums in the current board. Defaults to false. |
| | EnableThreadRating (see page 263) | If set to true, will display a 5 star rating system for users to rate a specific thread in a forum. |
| | ExcludeList (see page 264) | Enter a comma-separated list of custom user properties to exclude from the Forum (see page 256)'s profile page. For example, to suppress the Subscriptions field value, enter **subscriptions**. |
| | FilterXslt (see page 264) | Create your own custom xslt to filter forum content like HTML Attributes,Tags (see page 771) and unwanted words. See the CMS400.NET Developer Manual section |

| | | |
|---|---|---|
| | HideUserProfile (see page 265) | Set to True to suppress user profiles. The default is True.<br>• **True** - hides user profiles from non-administrator users<br>• **False** - displays user profiles from non-administrator users. However, an administrator can view a profile regardless of how this property is set.<br>On the other hand, if the Private Profile checkbox is checked for any user, his profile information is not visible, regardless of this setting.<br>When the profile is visible, only properties and their values not listed in the `ExcludeList (see page 264)` property appear. |
| | HighlightedUserRole (see page 265) | Set to True to suppress user profiles. The default is True.<br>• **True** - hides user profiles from non-administrator users<br>• **False** - displays user profiles from non-administrator users. However, an administrator can view a profile regardless of how this property is set.<br>On the other hand, if the Private Profile checkbox is checked for any user, his profile information is not visible, regardless of this setting.<br>When the profile is visible, only properties and their values not listed in the `ExcludeList (see page 264)` property appear. |
| | JavascriptEditorHeight (see page 265) | Set the height in pixels for the editor. The default is 400. |
| | JavascriptEditorHTMLMode (see page 266) | Set to True to allow users to edit their posts and replies in HTML. The default is False. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Allowing HTML Editing in the Javascript Editor" |
| | JavascriptEditorToolbar (see page 266) | JavaScriptEditor has been deprecated. Instead use EditorToolbar (see page 263) property to Add/Remove buttons in CMS400WebEditor." |
| | JavascriptEditorWidth (see page 267) | Set the width in pixels for the editor. The default is 625. |
| | OrderBy (see page 267) | The order of forum search results. Choices are:<br>• **Title (see page 223)** - The content title (alphabetical)<br>• **ID** - The content ID number<br>• **Date Created** - The date the content was created<br>• **Date Modified** - The date the content was most recently modified<br>• **Editor** - The user who last edited the content (alphabetical)<br>• **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 694)" > "Display of Search (see page 455) Results" > "Search (see page 455) Result Ranking" |
| | OrderDirection (see page 268) | Specify the order direction of forum search results. Choices are **Ascending** or **Descending**. |
| | ProfileLink (see page 268) | Enter a link to the user profile page. This allows a user to click a link in the control and be taken to the user's profile page. There is a variables used within the link that represent the user's ID.<br>**{0}** - represents the user's ID.<br>The Web form can be relative or absolute. Below is an example.<br>`userprofilepage.aspx?uid={0}` |
| | ProfileLinkTarget (see page 268) | Gets or Sets the target attribute for the user profile links. |
| | ShowCategories (see page 268) | If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False.<br>**True** = Search (see page 455) across taxonomy categories<br>**False** = Do not search across taxonomy categories |
| | TaxCategories (see page 269) | Is server control in workarea |
| | TaxonomyCols (see page 269) | Enter the number of columns in which this taxonomy/category will appear on the page. |

| | | |
|---|---|---|
| | TaxonomyMarkupLanguage (see page 269) | Enter name of a markup file that manages the display of the directory server control. |
| | Theme (see page 269) | Enter the custom theme's folder name. The theme can be relative to the site root or located in the current folder. If you do not specify a theme, the property uses the location defined by the Discussion Board's CSS theme property in the Workarea. |

### Forum.AllowAnonymousPost Property

If set to false, will not allow anonymous post on the forum. Defaults to false.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("If set to false, will not allow anonymous post on the forum. Defaults to true.")]
[DefaultValue(true)]
[Personalizable(), WebBrowsable()]
public bool AllowAnonymousPost;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("If set to false, will not allow anonymous post on the forum. Defaults to true.")>
<DefaultValue(true)>
<Personalizable(), WebBrowsable()>
Public Property AllowAnonymousPost() As bool
```

### Forum.BoardID Property

The ID of the Discussion Board to display on this page if one is not defined in a query string parameter.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the Discussion Board to display on this page if one is not defined in
a query string parameter.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long BoardID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the Discussion Board to display on this page if one is not defined in
a query string parameter.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property BoardID() As long
```

### Forum.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
```

```
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

### Forum.CustomOrderBy Property

Order search results by a custom property. Set this property to the friendly name of the property defined in the Indexing Service. If an invalid property is specified, no search results are returned. If this is left empty, the OrderBy (see page 267) property is used to order search results.

**C#**

```
[Category("Ektron"), DefaultValue(""), Description("Order search results by a custom property.
Set this property to the friendly name of the property.")]
public string CustomOrderBy;
```

**Visual Basic**

```
<Category("Ektron"), DefaultValue(""), Description("Order search results by a custom property.
Set this property to the friendly name of the property.")>
Public Property CustomOrderBy() As string
```

### Forum.DynamicForumParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically.")]
[DefaultValue("f")]
public string DynamicForumParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically.")>
<DefaultValue("f")>
Public Property DynamicForumParameter() As string
```

### Forum.DynamicGroupParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Group ID dynamically.")]
[DefaultValue("g")]
public string DynamicGroupParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
```

```
<Description("Gets or Sets the QueryString parameter to read a Group ID dynamically.")>
<DefaultValue("g")>
Public Property DynamicGroupParameter() As string
```

### Forum.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 24) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 24) (static.)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically. Set to
\"Use Default\" if you wish to always display the default Forum (static.)")]
[TypeConverter(typeof(DynamicBoardIDTypeConverter))]
[DefaultValue("boardid")]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Forum ID dynamically. Set to
\"Use Default\" if you wish to always display the default Forum (static.)")>
<TypeConverter(typeof(DynamicBoardIDTypeConverter))>
<DefaultValue("boardid")>
Public Property DynamicParameter() As string
```

### Forum.DynamicThreadParameter Property

Gets or Sets the QueryString parameter to read a Forum (see page 256) ID dynamically.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Thread ID dynamically.")]
[DefaultValue("t")]
public string DynamicThreadParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Thread ID dynamically.")>
<DefaultValue("t")>
Public Property DynamicThreadParameter() As string
```

### Forum.EditorCSS Property

Set the stylesheet on the editor.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Set stylesheet to the editor")]
public string EditorCSS;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Set stylesheet to the editor")>
Public Property EditorCSS() As string
```

### Forum.EditorToolbar Property

Decide which buttons are available in the Editor. For additional information on the Toolbar Editor, See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Adding and Removing Toolbar Buttons From the Editor

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("StyleMenu, FontMenu, ParagraphMenu, TextFormatMenu, LinkMenu, ClipBoardMenu,
SymbolsMenu, EmoticonSelect, WMV, Table")]
[Description("Decide which buttons are available in the Forum Editor. StyleMenu|Styles;
FontMenu|FontSize,FontName,FontColor;ParagraphMenu|JustifyLeft,JustifyCenter,JustifyRight,Justi
fyFull,BulletedList,NumberedList,Indent,Outdent;
TextFormatMenu|Bold,Italic,Underline,Strikethrough;Superscript,Subscript;
LinkMenu|CreateLink,Unlink;ClipBoardMenu|Cut,Copy,Paste;Undo,Redo,Print; SymbolsMenu;
EmoticonSelect; WMV; Table")]
[Personalizable(), WebBrowsable()]
public string EditorToolbar;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("StyleMenu, FontMenu, ParagraphMenu, TextFormatMenu, LinkMenu, ClipBoardMenu,
SymbolsMenu, EmoticonSelect, WMV, Table")>
<Description("Decide which buttons are available in the Forum Editor. StyleMenu|Styles;
FontMenu|FontSize,FontName,FontColor;ParagraphMenu|JustifyLeft,JustifyCenter,JustifyRight,Justi
fyFull,BulletedList,NumberedList,Indent,Outdent;
TextFormatMenu|Bold,Italic,Underline,Strikethrough;Superscript,Subscript;
LinkMenu|CreateLink,Unlink;ClipBoardMenu|Cut,Copy,Paste;Undo,Redo,Print; SymbolsMenu;
EmoticonSelect; WMV; Table")>
<Personalizable(), WebBrowsable()>
Public Property EditorToolbar() As string
```

### Forum.EnableForumQuickSelect Property

If set to true, will display a drop down list of all forums in the current board. Defaults to false.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("If set to true, will display a drop down list of all forums in the current
board. Defaults to false.")]
[DefaultValue(false)]
[Personalizable(), WebBrowsable()]
public bool EnableForumQuickSelect;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("If set to true, will display a drop down list of all forums in the current
board. Defaults to false.")>
<DefaultValue(false)>
<Personalizable(), WebBrowsable()>
Public Property EnableForumQuickSelect() As bool
```

### Forum.EnableThreadRating Property

If set to true, will display a 5 star rating system for users to rate a specific thread in a forum.

**C#**

```
[Bindable(true)]
```

```
[Category("Ektron")]
[Description("If set to true, will display a 5 star rating system for users to rate a specific
thread in a forum.")]
[DefaultValue(false)]
[Personalizable(), WebBrowsable()]
public bool EnableThreadRating;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("If set to true, will display a 5 star rating system for users to rate a specific
thread in a forum.")>
<DefaultValue(false)>
<Personalizable(), WebBrowsable()>
Public Property EnableThreadRating() As bool
```

### Forum.ExcludeList Property

Enter a comma-separated list of custom user properties to exclude from the Forum (see page 256)'s profile page. For example, to suppress the Subscriptions field value, enter **subscriptions**.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Enter a comma-separated list of custom user properties to exclude from the
Forum?s profile page. For example, to suppress the Subscriptions field value, enter
\"subscriptions\".")]
[DefaultValue("")]
public string ExcludeList;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Enter a comma-separated list of custom user properties to exclude from the
Forum?s profile page. For example, to suppress the Subscriptions field value, enter
\"subscriptions\".")>
<DefaultValue("")>
Public Property ExcludeList() As string
```

### Forum.FilterXslt Property

Create your own custom xslt to filter forum content like HTML Attributes,Tags (see page 771) and unwanted words. See the CMS400.NET Developer Manual section

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("/workarea/xslt/ForumFilter.xslt")]
[Description("")]
[Personalizable(), WebBrowsable()]
public string FilterXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("/workarea/xslt/ForumFilter.xslt")>
<Description("")>
<Personalizable(), WebBrowsable()>
Public Property FilterXslt() As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Path to Custom Xslt | Specify the path to the xslt file.It can be relative |

### Forum.HideUserProfile Property

Set to True to suppress user profiles. The default is True.

- **True** - hides user profiles from non-administrator users
- **False** - displays user profiles from non-administrator users. However, an administrator can view a profile regardless of how this property is set.

On the other hand, if the Private Profile checkbox is checked for any user, his profile information is not visible, regardless of this setting.

When the profile is visible, only properties and their values not listed in the `ExcludeList (see page 264)` property appear.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Hide users' profile")]
public bool HideUserProfile;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Hide users' profile")>
Public Property HideUserProfile() As bool
```

### Forum.HighlightedUserRole Property

Set to True to suppress user profiles. The default is True.

- **True** - hides user profiles from non-administrator users
- **False** - displays user profiles from non-administrator users. However, an administrator can view a profile regardless of how this property is set.

On the other hand, if the Private Profile checkbox is checked for any user, his profile information is not visible, regardless of this setting.

When the profile is visible, only properties and their values not listed in the `ExcludeList (see page 264)` property appear.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Set which user roles are highlighted.")]
public int HighlightedUserRole;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Set which user roles are highlighted.")>
Public Property HighlightedUserRole() As Integer
```

### Forum.JavascriptEditorHeight Property

Set the height in pixels for the editor. The default is 400.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
```

```
[DefaultValue(400)]
[Description("The pixel height for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
[Personalizable(), WebBrowsable()]
public int JavascriptEditorHeight;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(400)>
<Description("The pixel height for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
<Personalizable(), WebBrowsable()>
Public Property JavascriptEditorHeight() As Integer
```

### Forum.JavascriptEditorHTMLMode Property

Set to True to allow users to edit their posts and replies in HTML. The default is False. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Allowing HTML Editing in the Javascript Editor"

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to True to allow users to edit their posts and replies in HTML. The default
is False. See the CMS400.NET Developer Manual section \"Discussion Board Server Controls\" >
\"Allowing HTML Editing in the Javascript Editor\"")]
[Obsolete("JavaScript Editor has been deprecated. Instead control is using CMS400WebEditor.")]
[Personalizable(), WebBrowsable()]
public bool JavascriptEditorHTMLMode;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to True to allow users to edit their posts and replies in HTML. The default
is False. See the CMS400.NET Developer Manual section \"Discussion Board Server Controls\" >
\"Allowing HTML Editing in the Javascript Editor\"")>
<Obsolete("JavaScript Editor has been deprecated. Instead control is using CMS400WebEditor.")>
<Personalizable(), WebBrowsable()>
Public Property JavascriptEditorHTMLMode() As bool
```

### Forum.JavascriptEditorToolbar Property

JavaScriptEditor has been deprecated. Instead use EditorToolbar (see page 263) property to Add/Remove buttons in CMS400WebEditor."

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("ParagraphMenu,FontFacesMenu,FontSizesMenu,FontForeColorsMenu|Bold,Italic,Underli
ne,Strikethrough;Superscript,Subscript,RemoveFormat|JustifyLeft,JustifyCenter,JustifyRight,Just
ifyFull;BulletedList,NumberedList,Indent,Outdent;CreateLink,Unlink,InsertRule|Cut,Copy,Paste;Un
do,Redo,Print")]
[Description("JavaScriptEditor has been deprecated. Instead use EditorToolbar property to
Add/Remove buttons in CMS400WebEditor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead use EditorToolbar property to
Add/Remove buttons in CMS400WebEditor.")]
[Personalizable(), WebBrowsable()]
public string JavascriptEditorToolbar;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
```

```
<DefaultValue("ParagraphMenu,FontFacesMenu,FontSizesMenu,FontForeColorsMenu|Bold,Italic,Underli
ne,Strikethrough;Superscript,Subscript,RemoveFormat|JustifyLeft,JustifyCenter,JustifyRight,Just
ifyFull;BulletedList,NumberedList,Indent,Outdent;CreateLink,Unlink,InsertRule|Cut,Copy,Paste;Un
do,Redo,Print")>
<Description("JavaScriptEditor has been deprecated. Instead use EditorToolbar property to
Add/Remove buttons in CMS400WebEditor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead use EditorToolbar property to
Add/Remove buttons in CMS400WebEditor.")>
<Personalizable(), WebBrowsable()>
Public Property JavascriptEditorToolbar() As string
```

### Forum.JavascriptEditorWidth Property

Set the width in pixels for the editor. The default is 625.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(625)]
[Description("The pixel width for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
[Personalizable(), WebBrowsable()]
public int JavascriptEditorWidth;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(625)>
<Description("The pixel width for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
<Personalizable(), WebBrowsable()>
Public Property JavascriptEditorWidth() As Integer
```

### Forum.OrderBy Property

The order of forum search results. Choices are:

- **Title (see page 223)** - The content title (alphabetical)
- **ID** - The content ID number
- **Date Created** - The date the content was created
- **Date Modified** - The date the content was most recently modified
- **Editor** - The user who last edited the content (alphabetical)
- **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 694)" > "Display of Search (see page 455) Results" > "Search (see page 455) Result Ranking"

**C#**

```
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(5)]
[Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified. Default: Rank")]
[Personalizable(), WebBrowsable()]
public WebSearchOrder OrderBy;
```

**Visual Basic**

```
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(5)>
<Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified. Default: Rank")>
<Personalizable(), WebBrowsable()>
```

```
Public Property OrderBy() As WebSearchOrder
```

### Forum.OrderDirection Property

Specify the order direction of forum search results. Choices are **Ascending** or **Descending**.

**C#**

```
[Browsable(true)]
[Category("Ektron")]
[TypeConverter(typeof(SortOrderTypeConverter))]
[Description("Specify the order direction of the results.")]
[Personalizable(), WebBrowsable()]
public string OrderDirection;
```

**Visual Basic**

```
<Browsable(true)>
<Category("Ektron")>
<TypeConverter(typeof(SortOrderTypeConverter))>
<Description("Specify the order direction of the results.")>
<Personalizable(), WebBrowsable()>
Public Property OrderDirection() As string
```

### Forum.ProfileLink Property

Enter a link to the user profile page. This allows a user to click a link in the control and be taken to the user's profile page. There is a variables used within the link that represent the user's ID.

**{0}** - represents the user's ID.

The Web form can be relative or absolute. Below is an example.

```
userprofilepage.aspx?uid={0}
```

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue("Use Default"),
TypeConverter(typeof(ForumLinkConverter)), Description("The user profile link. {0} is the user
ID.") ]
public string ProfileLink;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue("Use Default"),
TypeConverter(typeof(ForumLinkConverter)), Description("The user profile link. {0} is the user
ID.") >
Public Property ProfileLink() As string
```

### Forum.ProfileLinkTarget Property

Gets or Sets the target attribute for the user profile links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the user profile links.") ]
public ItemLinkTargets ProfileLinkTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the user profile links.") >
Public Property ProfileLinkTarget() As ItemLinkTargets
```

### Forum.ShowCategories Property

If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False.

**True** = Search (see page 455) across taxonomy categories

**False** = Do not search across taxonomy categories

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")]
public bool ShowCategories;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")>
Public Property ShowCategories() As bool
```

### Forum.TaxCategories Property

Is server control in workarea

**C#**

```
[Browsable(false)]
public string TaxCategories;
```

**Visual Basic**

```
<Browsable(false)>
Public Property TaxCategories() As string
```

### Forum.TaxonomyCols Property

Enter the number of columns in which this taxonomy/category will appear on the page.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)]
[Description("This aligns the taxonomy categories in number of columns.")]
public int TaxonomyCols;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("This aligns the taxonomy categories in number of columns.")>
Public Property TaxonomyCols() As Integer
```

### Forum.TaxonomyMarkupLanguage Property

Enter name of a markup file that manages the display of the directory server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of the directory server control.")]
public string TaxonomyMarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of the directory server control.")>
Public Property TaxonomyMarkupLanguage() As string
```

### Forum.Theme Property

Enter the custom theme's folder name. The theme can be relative to the site root or located in the current folder. If you do not

specify a theme, the property uses the location defined by the Discussion Board's CSS theme property in the Workarea.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Enter the custom theme?s folder name. The theme can be relative to the site root
or located in the current folder. If you do not specify a theme, the property uses the
location defined by the Discussion Board?s CSS theme property in the Workarea.")]
[Personalizable(), WebBrowsable()]
public string Theme;
```

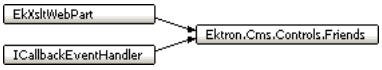
**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Enter the custom theme?s folder name. The theme can be relative to the site root
or located in the current folder. If you do not specify a theme, the property uses the
location defined by the Discussion Board?s CSS theme property in the Workarea.")>
<Personalizable(), WebBrowsable()>
Public Property Theme() As string
```

## Friends Class

Summary description for Friends (Colleague Server Control API (see page 639))

**Class Hierarchy**




**C#**

```
[DefaultProperty("DefaultUserID"), ToolboxData("<{0}:Friends runat=server></{0}:Friends>")]
public class Friends : EkXsltWebPart, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultUserID"), ToolboxData("<{0}:Friends runat=server></{0}:Friends>")>
Public Class Friends
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

**File**

Friends.cs

### Friends Methods

The methods of the Friends class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 270) | Populates the data for the Friends (see page 270). |
| | ToString (see page 271) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 271) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### Friends.Fill Method

Populates the data for the Friends (see page 270).

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

### Friends.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

### Friends.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

**Returns**

## Friends Properties

The properties of the Friends class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AvatarHeight (see page 272) | Gets or Sets the avatar height. 0 is unlimited. |
| | AvatarWidth (see page 272) | Gets or Sets the avatar width. 0 is unlimited. |
| | CacheInterval (see page 273) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultUserID (see page 273) | The default user ID for this control to use when there is no matching dynamic parameter value passed. |

| | | |
|---|---|---|
| | DisplayMode (see page 273) | Select the way this control displays colleagues. Choices are:<br>**Directory (see page 189)** - this selection allows users to group their colleagues by folders. When this selection is chosen, the following menu items are added to the control:<br>• **Add Folder (see page 652)** - allows a user to add a sub folder.<br>• **Move** - allows a user to move users into a folder. This item only appears when there is a folder.<br>**List** - shows a list colleagues in alphabetical order.<br>**Pending** - shows a list pending colleagues. This list is made up of users who have sent you colleagues requests.<br>**SentInvites** - shows a list... more (see page 273) |
| | DisplayXslt (see page 274) | Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. (optional). |
| | DynamicParameter (see page 274) | Gets or sets the Querystring parameter to read a user ID dynamically. To use the default user ID, leave blank. |
| | FriendLabel (see page 275) | Gets or sets the text for a colleague. |
| | FriendsLabel (see page 275) | Gets or Sets the text for colleagues. Similar to FriendLabel (see page 275), but is plural. |
| | Link (see page 275) | Enter a link to the user profile page. This allows a user to click a link in the Colleagues control and be taken to the user's profile page. There are two variables used within the link that represent the user's ID and the user's display name.<br>**{0}** - represents the user's ID.<br>**{1}** - represents the user's display name.<br>You need to have both variables in the link. The Web form can be relative or absolute. Below is an example.<br>`userprofilepage.aspx?uid={0}&dn={1}` |
| | LinkTarget (see page 275) | Gets or Sets the target attribute for the user links. |
| | MarkupLanguage (see page 276) | Enter name of a markup file that manages the display of this server control. |
| | MaxResults (see page 276) | Maximum number of results to be returned. |
| | ShowSortingOptions (see page 276) | Select wether the sorting options should be shown on the control. The following sorting options are available when this property is set to True:<br>• **Colleagues** - a list of current colleagues.<br>• **Pending Colleagues** - a list of colleagues who have sent you colleagues requests, but you have yet to accept.<br>• **Invited** - a list of colleagues to whom you have sent colleague requests. |
| | Stylesheet (see page 277) | Specifies the Css to which this control will provide a link. |
| | XmlDoc (see page 277) | Returns the Xslt object for the collection. |

### Friends.AvatarHeight Property

Gets or Sets the avatar height. 0 is unlimited.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") ]
public int AvatarHeight;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
height. 0 is unlimited.") >
Public Property AvatarHeight() As Integer
```

### Friends.AvatarWidth Property

Gets or Sets the avatar width. 0 is unlimited.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
```

```
width. 0 is unlimited.") ]
public int AvatarWidth;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(48), Description("Gets or Sets the avatar
width. 0 is unlimited.") >
Public Property AvatarWidth() As Integer
```

### Friends.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

### Friends.DefaultUserID Property

The default user ID for this control to use when there is no matching dynamic parameter value passed.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The default user ID for this control to use when there is no matching dynamic
parameter value passed.")]
public long DefaultUserID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The default user ID for this control to use when there is no matching dynamic
parameter value passed.")>
Public Property DefaultUserID() As long
```

### Friends.DisplayMode Property

Select the way this control displays colleagues. Choices are:

**Directory (see page 189)** - this selection allows users to group their colleagues by folders. When this selection is chosen, the following menu items are added to the control:

- **Add Folder (see page 652)** - allows a user to add a sub folder.
- **Move** - allows a user to move users into a folder. This item only appears when there is a folder.

**List** - shows a list colleagues in alphabetical order.

**Pending** - shows a list pending colleagues. This list is made up of users who have sent you colleagues requests.

**SentInvites** - shows a list of users to whom you've sent colleague requests.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Directory), Description("This
property enables the user to show their colleagues either in the List, Pending, SentInvite or
Directory display mode.")]
public eDisplayMode DisplayMode;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(eDisplayMode.Directory), Description("This
property enables the user to show their colleagues either in the List, Pending, SentInvite or
Directory display mode.")>
Public Property DisplayMode() As eDisplayMode
```

### Friends.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. (optional).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control. (optional)")]
[TypeConverter(typeof(XsltFriendTypeConverter))]
[DefaultValue("(Path To Custom Xslt)")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control. (optional)")>
<TypeConverter(typeof(XsltFriendTypeConverter))>
<DefaultValue("(Path To Custom Xslt)")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### Friends.DynamicParameter Property

Gets or sets the Querystring parameter to read a user ID dynamically. To use the default user ID, leave blank.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Gets or sets the Querystring parameter to read a user ID dynamically. To use the
default user ID, leave blank.")]
[Personalizable()]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Gets or sets the Querystring parameter to read a user ID dynamically. To use the
default user ID, leave blank.")>
```

```
<Personalizable()>
Public Property DynamicParameter() As string
```

### Friends.FriendLabel Property

Gets or sets the text for a colleague.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue("Colleague"), Description("Gets or Sets the
text for a colleague.") ]
public string FriendLabel;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue("Colleague"), Description("Gets or Sets the
text for a colleague.") >
Public Property FriendLabel() As string
```

### Friends.FriendsLabel Property

Gets or Sets the text for colleagues. Similar to FriendLabel (see page 275), but is plural.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue("Colleagues"), Description("Gets or Sets
the text for colleagues. Similar to FriendLabel, but is plural.") ]
public string FriendsLabel;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue("Colleagues"), Description("Gets or Sets
the text for colleagues. Similar to FriendLabel, but is plural.") >
Public Property FriendsLabel() As string
```

### Friends.Link Property

Enter a link to the user profile page. This allows a user to click a link in the Colleagues control and be taken to the user's profile page. There are two variables used within the link that represent the user's ID and the user's display name.

**{0}** - represents the user's ID.

**{1}** - represents the user's display name.

You need to have both variables in the link. The Web form can be relative or absolute. Below is an example.

```
userprofilepage.aspx?uid={0}&dn={1}
```

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The user link. {0} is the user ID
and {1} is the user display name.") ]
public string Link;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue("Use QuickLink"),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The user link. {0} is the user ID
and {1} is the user display name.") >
Public Property Link() As string
```

### Friends.LinkTarget Property

Gets or Sets the target attribute for the user links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the user links.") ]
```

```
public ItemLinkTargets LinkTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the user links.") >
Public Property LinkTarget() As ItemLinkTargets
```

### Friends.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

### Friends.MaxResults Property

Maximum number of results to be returned.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")]
[Personalizable()]
public int MaxResults;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of users to fetch per page. If set to zero, there is no
limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

### Friends.ShowSortingOptions Property

Select wether the sorting options should be shown on the control. The following sorting options are available when this property is set to True:

- **Colleagues** - a list of current colleagues.
- **Pending Colleagues** - a list of colleagues who have sent you colleagues requests, but you have yet to accept.
- **Invited** - a list of colleagues to whom you have sent colleague requests.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Description("Select wether the
sorting options should be shown on the control. The following sorting options are available
when this property is set to True: 'Colleagues' - a list of current colleagues. 'Pending
Colleagues' - a list of colleagues who have sent you colleague requests, but you have yet to
accept. 'Invited' - a list of colleagues to whom you have sent colleague requests.")]
public bool ShowSortingOptions;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Description("Select wether the
```

```
sorting options should be shown on the control. The following sorting options are available
when this property is set to True: 'Colleagues' - a list of current colleagues. 'Pending
Colleagues' - a list of colleagues who have sent you colleague requests, but you have yet to
accept. 'Invited' - a list of colleagues to whom you have sent colleague requests.")>
Public Property ShowSortingOptions() As bool
```

### Friends.Stylesheet Property

Specifies the Css to which this control will provide a link.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

### Friends.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## HtmlEditDisplay Class

### Class Hierarchy

**C#**

```
public class HtmlEditDisplay;
```

**Visual Basic**

```
Public Class HtmlEditDisplay
```

**File**

HtmlEditDisplay.cs

**Description**

This is class Ektron.Cms.Controls.HtmlEditDisplay.

### HtmlEditDisplay Methods

The methods of the HtmlEditDisplay class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | BaseJavaScript (see page 278) | Create the Javascript that handles the onready event and other CMS specific setup functionality. |
| | BeginJavaScript (see page 278) | 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 279) | This will end the surrounding of the JavaScript created for the CMS. |
| S | MakeFieldNameFromId (see page 279) | 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 279) | Creates the fields used when the content is posted. This creates fields which are posted to server with these IDs:<br>• [ControlName]_Action = the action to take - save, check in, undo checkout<br>• [ControlName]_ContentTitle = the content title<br>• [ControlName]_ContentId = The ID of the content<br>• [ControlName]_Language = The language of the content<br>The content title caption is surrounded with a span that has this ID:<br>• [ControlName]_TitleLabel<br>All of the information for all the content information fields (except the content itself) is surrounded with a div tag with this ID:<br>• [ControlName]_ContentInfo<br>The valid values for the [ControlName]_Action value are:<br>• Publish - Submit... more (see page 279) |
| | SelectedJavaScript (see page 280) | Renders any JavaScript based on options set by the developer through the properties. |

## HtmlEditDisplay.BaseJavaScript Method

Create the Javascript that handles the onready event and other CMS specific setup functionality.

**C#**

```
public string BaseJavaScript(string strTranslationFile, bool bShowCMSToolbar);
```

**Visual Basic**

```
Public Function BaseJavaScript(strTranslationFile As string, bShowCMSToolbar As bool) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| 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(long iContentId, int iLanguageID);
```

**Visual Basic**

```
Public Function BeginJavaScript(iContentId As long, 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 is posted. This creates fields which are posted to server with these IDs:

- [ControlName]_Action = the action to take - save, check in, undo checkout
- [ControlName]_ContentTitle = the content title
- [ControlName]_ContentId = The ID of the content
- [ControlName]_Language = The language of the content

The content title caption is surrounded with a span that has this ID:

- [ControlName]_TitleLabel

All of the information for all the content information fields (except the content itself) is surrounded with a div tag with this ID:

- [ControlName]_ContentInfo

The valid values for the [ControlName]_Action value are:

- Publish - Submit the posted content for publication
- CheckIn - Check in the content
- Save - Save the content and redisplay in the editor
- UndoCheckout - Undo the checkout of the content

**C#**

```
public string RenderCMSFields(string strContentTitle, string strTitleLabel, bool bShowTitle);
```

**Visual Basic**

```
Public Function RenderCMSFields(strContentTitle As string, strTitleLabel As string, bShowTitle
As bool) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| 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. The eWebEditPro editor can easily be placed onto a page for content editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor.

The control, at run time, produces the fields listed below. They are posted with the content and can be processed in the receiving page.

- [ControlName]_Action = the action to take - save, check in, undo checkout
- [ControlName]_ContentTitle = the content title
- [ControlName]_ContentId = the ID of the content
- [ControlName]_Language = the language of the content

The content title caption is surrounded with a span that has this ID: [ControlName]_TitleLabel.

All of the information for all the content information fields (except the content itself) is surrounded with a div tag with this ID: [ControlName]_ContentInfo.

The field values that are tied to the particular server control can be read through the respective properties:

- [ControlName]_Action read with .ContentAction (see page 285)
- [ControlName]_ContentTitle read with .ContentTitle (see page 286)
- [ControlName]_ContentId read with .ContentId
- [ControlName]_Language read with .ContentLanguage
- [ControlName]_SafeEncode read with .SafeEncode (see page 291)

The content sent or posted is retrieved with the simple: .Text (see page 293)

The valid values for the [ControlName]_Action value are:

- Publish - Submit the posted content for publication

- CheckIn - Check in the content
- Save - Save the content and redisplay in the editor
- UndoCheckout - Undo the checkout of the content

**Class Hierarchy**

**C#**

```
[ DefaultProperty("ContentID"), ToolboxData("<{0}:HtmlEditor runat=server></{0}:HtmlEditor>"),
ToolboxBitmap("HtmlEditor.bmp"), Designer(typeof(HtmlEditor.HtmlEditDesigner)) ]
public class HtmlEditor : eWebEditProEditor;
```

**Visual Basic**

```
< DefaultProperty("ContentID"), ToolboxData("<{0}:HtmlEditor runat=server></{0}:HtmlEditor>"),
ToolboxBitmap("HtmlEditor.bmp"), Designer(typeof(HtmlEditor.HtmlEditDesigner)) >
Public Class HtmlEditor
Inherits eWebEditProEditor
```

**File**

HtmlEditor.cs

## HtmlEditor.HtmlEditor Constructor

Constructor of the content editor. There are no parameters. The control depends on the settings given at design time or at run time.

**C#**

```
public HtmlEditor();
```

**Visual Basic**

```
Public Sub New()
```

## HtmlEditor Methods

The methods of the HtmlEditor class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | ExecuteContentAction (see page 281) | This uses the ContentAction (see page 285) value and performs the specified action on the content contained within the server control's posted values. The content and the other property values are placed into the CMS400 according to the Action value.<br>The possible action values are:<br>• "Submit"<br>• "CheckIn"<br>• "Save"<br>• "UndoCheckout" (cancel or undo) |
| | ToString (see page 282) | Returns the string which would be produced when the server control is rendered in the browser. |
| | WorkareaMode (see page 282) | Controls (see page 7) the operation of the internals of the server control. If this is not called, or given a value of 0, then this fills in all the necessary functionality to edit CMS content. |

## HtmlEditor.ExecuteContentAction Method

This uses the ContentAction (see page 285) value and performs the specified action on the content contained within the server control's posted values. The content and the other property values are placed into the CMS400 according to the Action value.

The possible action values are:

- "Submit"
- "CheckIn"
- "Save"
- "UndoCheckout" (cancel or undo)

**C#**

```
public string ExecuteContentAction();
```

**Visual Basic**

```
Public Function ExecuteContentAction() As string
```

**Returns**

### HtmlEditor.ToString Method

Returns the string which would be produced when the server control is rendered in the browser.

**C#**

```
public new string ToString();
```

**Visual Basic**

```
Public new Function ToString() As string
```

**Returns**

The complete HTML produced by the control when rendered.

### HtmlEditor.WorkareaMode Method

Controls (see page 7) the operation of the internals of the server control. If this is not called, or given a value of 0, then this fills in all the necessary functionality to edit CMS content.

**C#**

```
public void WorkareaMode(int iMode);
```

**Visual Basic**

```
Public Function WorkareaMode(iMode As Integer) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 284) | This signifies that the selected content is archived. |
| | Authenticated (see page 284) | 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 285) | DESIGN TIME ONLY: Used to find the Web service required to access the CMS in Design Time. |
| | ContentAction (see page 285) | The action to perform on the posted content. The CMS buttons will set this value when selected. This is mostly useful on the receipt of the post as this will inform the code behind as to what action to take. |

| | | |
|---|---|---|
| | ContentID (see page 286) | Specifies the content ID to load. This value must match a value that exists in the CMS database. The selection of content is similar to that in the ContentBlock (see page 140) control. The content can only be accessed if the user is logged in. This value can be set only on editor load, before the editor and contents are rendered. After editor load, this value is ignored.<br>If this value is 0, then the code behind must handle making the calls that check out content and setting the content at edit time by placing it into the .Text (see page 293) property for the... more (see page 286) |
| | ContentTitle (see page 286) | The title for the selected Content (see page 788). This is set when new content is read. It can not be changed by the code behind. A text field is created, with the name EktContentTitle, that contains the description. The user can change it there. |
| | ContentType (see page 287) | This value describes the type of content to which the selected ID refers. If this value is blank, then the content is a custom type, to be defined type, or unsupported type. These are the possible types:<br>• HTML<br>• Form (see page 836) |
| | EditorReadyCall (see page 287) | 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.<br>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 288) | If 'true', the user can upload locally selected images and linked files that they have put into their content. If enabled, the automatic upload mechanism will allow uploading of the local files automatically with minimal user intervention. If 'false', the site developer must provide an upload page, such as the CMS library, where the upload can occur. |
| | IncludeContentInfo (see page 288) | If true, includes the information about the content in fields posted with a form post. (These fields are defined in the summary of this control.)<br>Set the value to 'false' if the page maintains these values or the page is just using the editor to edit general formatted text.<br>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 292) property to false. |
| | IsLoggedIn (see page 289) | 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 289) | Specifies the ID of the language to use for the content to load. This value must match a value that exists in the CMS database. |
| | loggedInUserID (see page 290) | 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 290) | DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in. |
| | Path (see page 290) | The path to the site that contains this control. This is not available in the properties window. It is meant for code behind which wants to override the path used by the CMS.<br>The path is automatically determined by the control at run time, if it is not set by the code behind.<br>This is only available from Code Behind. |
| | PreferredType (see page 291) | The display to use for viewing the content. This can be used to simplify the display or to debug the operation of editing content on the site.<br>The options selectable are:<br>• "activex" = Use the ActiveX control<br>• "textarea" = Show HTML in text area<br>The default is "activex". This is only available from Code Behind. |
| | SafeEncode (see page 291) | If true, then the posted HTML is encoded using URL encoding. This makes the transfer secure. If the content is encoded then it must be decoded on the receiving side.<br>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 292) | Controls (see page 7) whether the CMS toolbar is displayed in the editor's toolbar. If this value is true, then commands containing CMS functionality are displayed in the editor's toolbar. The commands are for check-in, publish, and edit cancellation. |
| | ShowTitle (see page 292) | If 'true', the title of the selected content is displayed. The user can then edit the title. The submission of content posts up the title, even if it is not displayed to the user, unless the IncludeContentInfo (see page 288) property is set to a false value. |
| | StyleSheet (see page 293) | Assigns a single stylesheet to use with the edited content. |
| | Text (see page 293) | The HTML text being edited. |

| | | |
|---|---|---|
| | ToolbarLevel (see page 294) | 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.<br>These are the value values:<br>• "None"<br>• "Reduced"<br>• "Default"<br>• "DataDesigner"<br>• "DataEntry"<br>• "Calendar (see page 69)"<br>• "FormDesign"<br>• "XSLTDesign"<br>• "Full"<br>• "Custom" |
| | ToolbarResetCall (see page 294) | This specifies which JavaScript function to run when the toolbar resets. The toolbar reset offers the opportunity for the script to affect the operation of the toolbar with actions such as adding or removing commands, disabling commands, or creating entirely new toolbars.<br>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 295) | The translation file (or stream) that contains the XML elements for the translation of strings used in the editor. If this is blank, the client system's windows setting is used to select form the series of default translation files. |

### HtmlEditor.Archived Property

This signifies that the selected content is archived.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("Is the content selected archived.")]
public bool Archived;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("Is the content selected archived.")>
Public Property Archived() As bool
```

### HtmlEditor.Authenticated Property

DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in VisualStudio.NET.

**C#**

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
```

```
[Category("Ektron Authentication")]
[Editor(typeof(LoginUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(false)]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Description("DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and
permits them to login through a login dialog in VisualStudio.NET.")]
public string Authenticated;
```

**Visual Basic**

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<Editor(typeof(LoginUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(false)>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Description("DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and
permits them to login through a login dialog in VisualStudio.NET.")>
Public Property Authenticated() As string
```

### HtmlEditor.CmsSitePath Property

DESIGN TIME ONLY: Used to find the Web service required to access the CMS in Design Time.

**C#**

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
[Category("Ektron Authentication")]
[DefaultValue("")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")]
public string CmsSitePath;
```

**Visual Basic**

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<DefaultValue("")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Editor(typeof(SelectWSUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("DESIGN TIME ONLY: Used to find the web service required to access the CMS in
Design Time.")>
Public Property CmsSitePath() As string
```

### HtmlEditor.ContentAction Property

The action to perform on the posted content. The CMS buttons will set this value when selected. This is mostly useful on the receipt of the post as this will inform the code behind as to what action to take.

**C#**

```
[Bindable(true)]
[Browsable(false)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The action to perform with the content.")]
public string ContentAction;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The action to perform with the content.")>
Public Property ContentAction() As string
```

### HtmlEditor.ContentID Property

Specifies the content ID to load. This value must match a value that exists in the CMS database. The selection of content is similar to that in the ContentBlock (see page 140) control. The content can only be accessed if the user is logged in. This value can be set only on editor load, before the editor and contents are rendered. After editor load, this value is ignored.

If this value is 0, then the code behind must handle making the calls that check out content and setting the content at edit time by placing it into the .Text (see page 293) property for the editor.

A hidden field, named EktContentID_[ControlName], is created to hold this value. It is posted up with the rest of the content.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("The CMS ID of the content to load.")]
public long ContentID;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("The CMS ID of the content to load.")>
Public Property ContentID() As long
```

### HtmlEditor.ContentTitle Property

The title for the selected Content (see page 788). This is set when new content is read. It can not be changed by the code behind. A text field is created, with the name EktContentTitle, that contains the description. The user can change it there.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The title of the content.")]
public string ContentTitle;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The title of the content.")>
Public Property ContentTitle() As string
```

### HtmlEditor.ContentType Property

This value describes the type of content to which the selected ID refers. If this value is blank, then the content is a custom type, to be defined type, or unsupported type. These are the possible types:

- HTML
- Form (see page 836)

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The type of content selected.")]
public string ContentType;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The type of content selected.")>
Public Property ContentType() As string
```

### HtmlEditor.EditorReadyCall Property

This specifies which JavaScript function to run when the HTML editor is ready to start processing commands. This happens after the initialization of the editor and the loading of the initial content.

This is the name, not the full function definition, of a JavaScript function. The function must follow the format defined in the eWebEditPro documentation for this type of method. Below is an example:

```
function initEditorIsReady(sEditor, strURL, strAutoURL)
{ ... }
```

where the "initEditorIsReady" string is given to this property.

If this string is empty, the default CMS method is called.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
```

```
[DefaultValue("")]
[Description("Specifies the JavaScript function to run when the editor is ready to start
processing commands.")]
public string EditorReadyCall;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("Specifies the JavaScript function to run when the editor is ready to start
processing commands.")>
Public Property EditorReadyCall() As string
```

### HtmlEditor.EnableUpload Property

If 'true', the user can upload locally selected images and linked files that they have put into their content. If enabled, the automatic upload mechanism will allow uploading of the local files automatically with minimal user intervention. If 'false', the site developer must provide an upload page, such as the CMS library, where the upload can occur.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("Allows the user to upload images and linked files.")]
public bool EnableUpload;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Allows the user to upload images and linked files.")>
Public Property EnableUpload() As bool
```

### HtmlEditor.IncludeContentInfo Property

If true, includes the information about the content in fields posted with a form post. (These fields are defined in the summary of this control.)

Set the value to 'false' if the page maintains these values or the page is just using the editor to edit general formatted text.

If the information must be posted, but not shown to the end user, then define the class [ControlName]_TitleLabel as having no display, or set the ShowTitle (see page 292) property to false.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
```

```
[Description("Include the content information read from the database.")]
public bool IncludeContentInfo;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Include the content information read from the database.")>
Public Property IncludeContentInfo() As bool
```

### HtmlEditor.IsLoggedIn Property

DESIGN TIME ONLY: Read only property that lets a developer know if the user is currently log into the site as a user.

**C#**

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
[Category("Ektron Authentication")]
[DefaultValue(false)]
[Description("DESIGN TIME ONLY: Read only property that displays whether the user is currently
logged into the CMS site.")]
public bool IsLoggedIn;
```

**Visual Basic**

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<DefaultValue(false)>
<Description("DESIGN TIME ONLY: Read only property that displays whether the user is currently
logged into the CMS site.")>
Public Property IsLoggedIn() As bool
```

### HtmlEditor.Language Property

Specifies the ID of the language to use for the content to load. This value must match a value that exists in the CMS database.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[DefaultValue(1033)]
[Description("The Lanugage of the content to load.")]
[Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int Language;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<DefaultValue(1033)>
<Description("The Lanugage of the content to load.")>
<Editor(typeof(SelectLanguageIDTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
```

```
Public Property Language() As Integer
```

### HtmlEditor.loggedInUserID Property

DESIGN TIME ONLY: Read only property that lets a developer know the user ID of the user who is currently logged in.

**C#**

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
[Category("Ektron Authentication")]
[DefaultValue(0)]
[Description("DESIGN TIME ONLY: Read only property which displays the validated user ID for
the CMS site.")]
public long loggedInUserID;
```

**Visual Basic**

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<DefaultValue(0)>
<Description("DESIGN TIME ONLY: Read only property which displays the validated user ID for
the CMS site.")>
Public Property loggedInUserID() As long
```

### HtmlEditor.loggedInUserName Property

DESIGN TIME ONLY: Read only property that lets a developer know the user name of the user who is currently logged in.

**C#**

```
[Browsable(true)]
[DesignOnly(true)]
[Bindable(true)]
[Category("Ektron Authentication")]
[DefaultValue("")]
[Description("DESIGN TIME ONLY: Read only property which displays the validated login name for
the CMS site.")]
public string loggedInUserName;
```

**Visual Basic**

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<DefaultValue("")>
<Description("DESIGN TIME ONLY: Read only property which displays the validated login name for
the CMS site.")>
Public Property loggedInUserName() As string
```

### HtmlEditor.Path Property

The path to the site that contains this control. This is not available in the properties window. It is meant for code behind which wants to override the path used by the CMS.

The path is automatically determined by the control at run time, if it is not set by the code behind.

This is only available from Code Behind.

**C#**

```
[Bindable(true)]
[Browsable(false)]
[DesignOnly(false)]
[Category("Ektron")]
```

```
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[DefaultValue("")]
[Description("The path to the site that contains this control.")]
public new string Path;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<DefaultValue("")>
<Description("The path to the site that contains this control.")>
Public new Property Path() As string
```

### HtmlEditor.PreferredType Property

The display to use for viewing the content. This can be used to simplify the display or to debug the operation of editing content on the site.

The options selectable are:

- "activex" = Use the ActiveX control
- "textarea" = Show HTML in text area

The default is "activex". This is only available from Code Behind.

**C#**

```
[Bindable(true)]
[Browsable(false)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[DefaultValue("")]
[Description("The path to the site that contains this control.")]
public new string PreferredType;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<DefaultValue("")>
<Description("The path to the site that contains this control.")>
Public new Property PreferredType() As string
```

### HtmlEditor.SafeEncode Property

If true, then the posted HTML is encoded using URL encoding. This makes the transfer secure. If the content is encoded then it must be decoded on the receiving side.

If the decode process is not wanted, then set this value to false. This will require the validateRequest attribute in the page definition be set to false.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
```

```
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("Set to True to encode the HTML content.")]
public bool SafeEncode;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Set to True to encode the HTML content.")>
Public Property SafeEncode() As bool
```

**Example 1**

validateRequest="false"

**Example 2**

@Page Language (see page 289)="VB" validateRequest="false" AutoEventWireup="false" CodeFile="EditAnyHtmlSample.aspx.vb" Inherits="EditAnyHtmlSample"

### HtmlEditor.ShowCMSToolbar Property

Controls (see page 7) whether the CMS toolbar is displayed in the editor's toolbar. If this value is true, then commands containing CMS functionality are displayed in the editor's toolbar. The commands are for check-in, publish, and edit cancellation.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("Show CMS commands in the toolbar.")]
public bool ShowCMSToolbar;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Show CMS commands in the toolbar.")>
Public Property ShowCMSToolbar() As bool
```

### HtmlEditor.ShowTitle Property

If 'true', the title of the selected content is displayed. The user can then edit the title. The submission of content posts up the title, even if it is not displayed to the user, unless the IncludeContentInfo (see page 288) property is set to a false value.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
```

```
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("Set to True to show the content title for edit.")]
public bool ShowTitle;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Set to True to show the content title for edit.")>
Public Property ShowTitle() As bool
```

### HtmlEditor.StyleSheet Property

Assigns a single stylesheet to use with the edited content.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The stylesheet to use for the edited content.")]
public new string StyleSheet;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The stylesheet to use for the edited content.")>
Public new Property StyleSheet() As string
```

### HtmlEditor.Text Property

The HTML text being edited.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The HTML text being edited.")]
public new string Text;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
```

```
<DefaultValue("")>
<Description("The HTML text being edited.")>
Public new Property Text() As string
```

### HtmlEditor.ToolbarLevel Property

This specifies the level of the toolbar feature set offered to the user. It may be that they need a minimal set of functionality to keep the editing simple. It may be they need full functionality for a major edit. This defines the amount of functionality offered to the user.

These are the value values:

- "None"
- "Reduced"
- "Default"
- "DataDesigner"
- "DataEntry"
- "Calendar (see page 69)"
- "FormDesign"
- "XSLTDesign"
- "Full"
- "Custom"

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[TypeConverter(typeof(HtmlEditToolbarLvlTypeConverter))]
[DefaultValue("Default")]
[Description("Specifies the level of toolbar feature set offered to the user.")]
public string ToolbarLevel;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<TypeConverter(typeof(HtmlEditToolbarLvlTypeConverter))>
<DefaultValue("Default")>
<Description("Specifies the level of toolbar feature set offered to the user.")>
Public Property ToolbarLevel() As string
```

### HtmlEditor.ToolbarResetCall Property

This specifies which JavaScript function to run when the toolbar resets. The toolbar reset offers the opportunity for the script to affect the operation of the toolbar with actions such as adding or removing commands, disabling commands, or creating entirely new toolbars.

This is the name, not the full function definition, of a JavaScript function. The function must follow the format defined in the eWebEditPro documentation for this type of method. Below is an example:

```
function initToolbarCommands(sEditor, strURL, strAutoURL)
{ ... }
```

Where the "initToolbarCommands" string is given to this property.

A toolbar reset is only called when the toolbar actually resets. This often happens when the editor starts, however, if the user has been given the opportunity to save their own toolbar setup then this will not be called on startup. It is always called when the user selects to reset the toolbar.

If this string is empty, the default CMS method is called.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("Specifies the JavaScript function to run when the toolbar is reset.")]
public string ToolbarResetCall;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("Specifies the JavaScript function to run when the toolbar is reset.")>
Public Property ToolbarResetCall() As string
```

### HtmlEditor.TranslationFile Property

The translation file (or stream) that contains the XML elements for the translation of strings used in the editor. If this is blank, the client system's windows setting is used to select form the series of default translation files.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("What file to use for translations.")]
public string TranslationFile;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("What file to use for translations.")>
Public Property TranslationFile() As string
```

## ImageControl Class

The ImageControl server control, when viewed on a Web form, displays an image stored within the CMS400.NET document management feature. When a user is logged in and has permission to edit the image, he can right click the image and select edit. This creates a drag and drop box that the user can use to update the file.

### Class Hierarchy

**C#**

```
public class ImageControl : EkWebPart;
```

**Visual Basic**

```
Public Class ImageControl
Inherits EkWebPart
```

**File**

ImageControl.cs

## ImageControl Methods

The methods of the ImageControl class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 296) | Method that can be overridden and used to grab data from the back end for each server control. |
| | Wrap (see page 296) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## ImageControl.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## ImageControl.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar |

### Returns

Html string wrapped with a toolbar

## ImageControl Properties

The properties of the ImageControl class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DefaultImageID (see page 297) | Gets or Sets the Default ImageID for this Control. |
| | DynamicParameter (see page 297) | Gets or Sets the QueryString parameter to read a Image ID dynamically. Set to String.Empty if you wish to always display the default Image ID (static.) |
| | FolderID (see page 297) | The ID of the folder you wish to select data from. Root is 0. |
| | TaxonomyID (see page 298) | The ID of the taxonomy where assets are added. |
| | Title (see page 298) | Gets or sets the Image Alternative/Title text. By default it is image file name. |

## ImageControl.DefaultImageID Property

Gets or Sets the Default ImageID for this Control.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default Image ID for this Control.")]
[Personalizable(), WebBrowsable()]
public long DefaultImageID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default Image ID for this Control.")>
<Personalizable(), WebBrowsable()>
Public Property DefaultImageID() As long
```

## ImageControl.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Image ID dynamically. Set to String.Empty if you wish to always display the default Image ID (static.)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Image ID dynamically. Set to
\"Use Default\" if you wish to always display the default Image ID (static.)")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
[Personalizable(), WebBrowsable()]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Image ID dynamically. Set to
\"Use Default\" if you wish to always display the default Image ID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
<Personalizable(), WebBrowsable()>
Public Property DynamicParameter() As string
```

## ImageControl.FolderID Property

The ID of the folder you wish to select data from. Root is 0.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder you wish to select data from. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long FolderID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property FolderID() As long
```

### ImageControl.TaxonomyID Property

The ID of the taxonomy where assets are added.

**C#**

```
[Category("Ektron")]
[DefaultValue("0")]
[Description("The ID of the taxonomy where assets are added.")]
[Personalizable(), WebBrowsable()]
public long TaxonomyID;
```

**Visual Basic**

```
<Category("Ektron")>
<DefaultValue("0")>
<Description("The ID of the taxonomy where assets are added.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyID() As long
```

### ImageControl.Title Property

Gets or sets the Image Alternative/Title text. By default it is image file name.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the Image Alternative/Title text. By defaut it is image file
name.")]
[DefaultValue("")]
public new string Title;
```

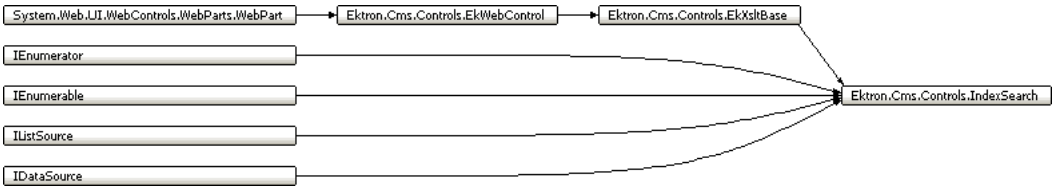
**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the Image Alternative/Title text. By defaut it is image file
name.")>
<DefaultValue("")>
Public new Property Title() As string
```

## IndexSearch Class

Use the IndexSearch server control to display a Smart Form (see page 836)'s search screen on any Web form. This lets a user search your site based on XML parameters.

### Class Hierarchy




**C#**

```
[DefaultProperty("XmlConfigId"), ToolboxData("<{0}:IndexSearch
runat=server></{0}:IndexSearch>"), PersistChildren(true)]
public class IndexSearch : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource;
```

**Visual Basic**

```
<DefaultProperty("XmlConfigId"), ToolboxData("<{0}:IndexSearch
runat=server></{0}:IndexSearch>"), PersistChildren(true)>
Public Class IndexSearch
Inherits EkXsltBase
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
```

**File**

IndexSearch.cs

## IndexSearch.IndexSearch Constructor

Constructor for IndexSearch class.

**C#**

```
public IndexSearch();
```

**Visual Basic**

```
Public Sub New()
```

## IndexSearch Structures

The structures of the IndexSearch class are listed here.

### Structures

| | Name | Description |
|---|---|---|
| | SearchPram (see page 299) | parameter structure used for search. |

## Ektron.Cms.Controls.IndexSearch.SearchPram Structure

parameter structure used for search.

**C#**

```
public struct SearchPram {
}
```

**Visual Basic**

```
Public Structure SearchPram
End Structure
```

**File**

IndexSearch.cs

## IndexSearch Methods

The methods of the IndexSearch class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddParm (see page 300) | This is the overview for the AddParm method overload. |
| | CopyTo (see page 301) | 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 301) | 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 301) | Gets a System.Collections.IEnumerator for the collection. |
| | GetSearchParmDataType (see page 301) | Returns the search parameter data type. |
| | GetSearchParmSearchType (see page 302) | Returns the search parameter search type. |
| | GetXPathValues (see page 302) | Returns array of strings using Xpaths. |
| | MoveNext (see page 302) | Moves up the index of the Collection (see page 88). Returns false if the end of the collection has been surpassed. |
| | Reset (see page 302) | Brings the index of the collection back to zero. Does not clear the collection. |
| | Search (see page 303) | This is the overview for the Search method overload. |
| | this (see page 303) | Index operator into a strongly typed dataset holding some common properties for a Collection (see page 88) item. |
| | Wrap (see page 303) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some HTML text. |

## AddParm Method

This is the overview for the AddParm method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | IndexSearch.AddParm () (see page 300) | Adds a parameter to the search. |
| | IndexSearch.AddParm (SearchPram) (see page 300) | 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 299) |

**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 88). Returns false if the end of the collection has been surpassed.

**C#**

```
public bool MoveNext();
```

**Visual Basic**

```
Public Function MoveNext() As bool
```

**Returns**

Boolean indicating if the move was within the range of the array.

### IndexSearch.Reset Method

Brings the index of the collection back to zero. Does not clear the collection.

**C#**

```
public void Reset();
```

**Visual Basic**

```
Public Function Reset() As void
```

## Search Method

This is the overview for the Search method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | IndexSearch.Search () (see page 303) | Performs the search action. |
| | IndexSearch.Search (System.Web.HttpRequest) (see page 303) | 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 88) 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 305) | If you want to display an image on the submit button, enter the server path to it. (Optional) |
| | ButtonText (see page 305) | The text that appears on the Search (see page 303) button. |
| | ContentParameter (see page 306) | Checks the QueryString for this value and replaces the search results with that content block when specified. Leave blank to always display the search results. |
| | Count (see page 306) | Indicates the number of elements in your Collection (see page 88). |
| | Current (see page 306) | Gets the current object in the collection. |
| | CustomOrderBy (see page 306) | Order search results by an indexed field in the smart form. Set this property to the XPATH of that indexed field. If invalid XPATH is specified, search result ordering defaults to Rank. |
| | EkItems (see page 307) | Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 88). Holds more information than the exposed strongly typed this[i] index operator. |
| | EmptyResultMsg (see page 307) | The message that is displayed when the search returns no hits. (Optional) |
| | EncodeContentHtml (see page 307) | Set the property to true if you wish to html encode the Content (see page 788) Html in returned result." |
| | FolderId (see page 307) | The ID of the folder where to start the search. Root starts at "0". |
| | FormName (see page 308) | The form name. |
| | IsSynchronized (see page 308) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | Labelafter (see page 308) | Specify a label for the date field where search criteria is "After". |
| | Labelbefore (see page 308) | Specify a label for the date field where search criteria is "Before". |
| | LabelBetween (see page 309) | Specify label for numeric field where search criteria is based on range of values. |
| | LabelContains (see page 309) | Specify the label for a string field where search criteria "Contains". |
| | LabeldateBetween (see page 309) | Specify a label for the date field where search criteria based on a range of dates. |
| | LabelEqual (see page 310) | Specify label for numeric field where search criteria is "Equal". |
| | LabelExactPhrase (see page 310) | Specify a label for the string field where search criteria "Exact Phrase". |
| | LabelGreaterThan (see page 310) | Specify label for numeric field where search criteria is "Greater Than". |
| | Labelinputdate (see page 311) | Specify a label for the message that appears when date entered are blank/incorrect. |
| | Labelinputnumber (see page 311) | Specify a label for the message that appears when values entered are blank/incorrect. |
| | LabelLessThan (see page 311) | Specify label for numeric field where search criteria "Less Than". |
| | Labelon (see page 312) | Specify a label for the Date field where search criteria is "On". |
| | LabelSelect (see page 312) | Specify labels for first item on selection dropdown. |
| | LinkTarget (see page 312) | Gets or sets the target window for the result to be displayed. |
| | MaxResults (see page 313) | The total number of results to display. If set to zero, unlimited results will be set. |

| | | |
|---|---|---|
| | OrderBy (see page 313) | Specify the order of the results. Choices are:<br>• **Title** - the title of the content block<br>• **ID** - the XML content block ID number<br>• **Date Created** - the date the content block was created<br>• **Date Modified** - the date the content block was last modified<br>• **AuthorLName** - the last author's last name<br>• **AuthorFname** - the last author's first name<br>• **StartDate** - the GoLive date |
| | OrderByDirection (see page 313) | The direction the results will be sorted in. This property works with the OrderBy (see page 313) property only. Choices are:<br>• Ascending<br>• Descending |
| | Recursive (see page 314) | Select True if you wish to search the subfolders of the selected FolderPath. |
| | SearchParmXML (see page 314) | Predefine a set of search criteria. |
| | ShowSearchBoxAlways (see page 314) | If this property is set to False, the search box will not be visible on PostBack. |
| | SyncRoot (see page 315) | Gets an object that can be used to synchronize the collection. |
| | Weighted (see page 315) | Select True if you wish perform a weighted search. |
| | XmlConfigId (see page 315) | ID of the Smart Form (see page 836) whose indexed fields will appear on the search screen. |
| | XmlDoc (see page 315) | Returns the XSLT object for the collection. |

### IndexSearch.ButtonImgSrc Property

If you want to display an image on the submit button, enter the server path to it. (Optional)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: The path to the image you wish to use for the Search button.")]
public string ButtonImgSrc;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: The path to the image you wish to use for the Search button.")>
Public Property ButtonImgSrc() As string
```

### IndexSearch.ButtonText Property

The text that appears on the Search (see page 303) button.

**C#**

```
[Category("Ektron")]
[Description("The text of the Search button.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_BTN)]
public string ButtonText;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("The text of the Search button.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
```

```
<DefaultValue(EkWebControl.RES_ISRCH_BTN)>
Public Property ButtonText() As string
```

### IndexSearch.ContentParameter Property

Checks the QueryString for this value and replaces the search results with that content block when specified. Leave blank to always display the search results.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and replaces the search results with a
content block when specified. Leave blank to always display the search results.")]
public string ContentParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and replaces the search results with a
content block when specified. Leave blank to always display the search results.")>
Public Property ContentParameter() As string
```

### IndexSearch.Count Property

Indicates the number of elements in your Collection (see page 88).

**C#**

```
[Category("Data")]
[Description("Indicates the number of elements in your Collection.")]
public int Count;
```

**Visual Basic**

```
<Category("Data")>
<Description("Indicates the number of elements in your Collection.")>
Public Property Count() As Integer
```

### IndexSearch.Current Property

Gets the current object in the collection.

**C#**

```
[Browsable(false)]
public object Current;
```

**Visual Basic**

```
<Browsable(false)>
Public Property Current() As object
```

### IndexSearch.CustomOrderBy Property

Order search results by an indexed field in the smart form. Set this property to the XPATH of that indexed field. If invalid XPATH is specified, search result ordering defaults to Rank.

**C#**

```
[Category("Ektron"), DefaultValue(""), Description("Order search results by an indexed field
in the smart form. Set this property to the XPATH of that indexed field.")]
public string CustomOrderBy;
```

**Visual Basic**

```
<Category("Ektron"), DefaultValue(""), Description("Order search results by an indexed field
in the smart form. Set this property to the XPATH of that indexed field.")>
Public Property CustomOrderBy() As string
```

### IndexSearch.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 88). Holds more information than the exposed strongly typed this[i] index operator.

**C#**

```
public Ektron.Cms.Common.IndexSearchItems EkItems;
```

**Visual Basic**

```
Public Property EkItems() As Ektron.Cms.Common.IndexSearchItems
```

### IndexSearch.EmptyResultMsg Property

The message that is displayed when the search returns no hits. (Optional)

**C#**

```
[Category("Ektron")]
[Description("Optional: The return message if the search returns no hits.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_NOHITS)]
public string EmptyResultMsg;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Optional: The return message if the search returns no hits.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_NOHITS)>
Public Property EmptyResultMsg() As string
```

### IndexSearch.EncodeContentHtml Property

Set the property to true if you wish to html encode the Content (see page 788) Html in returned result."

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to true, Content Html in returned result will be
encoded.")]
public bool EncodeContentHtml;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to true, Content Html in returned result will be
encoded.")>
Public Property EncodeContentHtml() As bool
```

### IndexSearch.FolderId Property

The ID of the folder where to start the search. Root starts at "0".

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
```

```
[Description("The ID of the folder you wish to start the search. Root starts at \"0\".")]
[Editor(typeof(GetFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long FolderId;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The ID of the folder you wish to start the search. Root starts at \"0\".")>
<Editor(typeof(GetFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderId() As long
```

### IndexSearch.FormName Property

The form name.

**C#**

```
[Bindable(false), Browsable(false)]
[Description("")]
public string FormName;
```

**Visual Basic**

```
<Bindable(false), Browsable(false)>
<Description("")>
Public Property FormName() As string
```

### IndexSearch.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

```
[Browsable(false)]
public bool IsSynchronized;
```

**Visual Basic**

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

### IndexSearch.Labelafter Property

Specify a label for the date field where search criteria is "After".

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the date field where search criteria is \"After\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_AFTER)]
public string Labelafter;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the date field where search criteria is \"After\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_AFTER)>
Public Property Labelafter() As string
```

### IndexSearch.Labelbefore Property

Specify a label for the date field where search criteria is "Before".

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the date field where search criteria is \"Before\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_BFORE)]
public string Labelbefore;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the date field where search criteria is \"Before\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_BFORE)>
Public Property Labelbefore() As string
```

### IndexSearch.LabelBetween Property

Specify label for numeric field where search criteria is based on range of values.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for numeric field where search criteria is based on range of
values.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_BTWEEN)]
public string LabelBetween;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for numeric field where search criteria is based on range of
values.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_BTWEEN)>
Public Property LabelBetween() As string
```

### IndexSearch.LabelContains Property

Specify the label for a string field where search criteria "Contains".

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify the label for a string field where search criteria \"Contains\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_CONTS)]
public string LabelContains;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify the label for a string field where search criteria \"Contains\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_CONTS)>
Public Property LabelContains() As string
```

### IndexSearch.LabeldateBetween Property

Specify a label for the date field where search criteria based on a range of dates.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the date field where search criteria based on a range of
dates.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_DBTWEEN)]
public string LabeldateBetween;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the date field where search criteria based on a range of
dates.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_DBTWEEN)>
Public Property LabeldateBetween() As string
```

### IndexSearch.LabelEqual Property

Specify label for numeric field where search criteria is "Equal".

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for numeric field where search criteria is \"Equal\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_EQUAL)]
public string LabelEqual;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for numeric field where search criteria is \"Equal\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_EQUAL)>
Public Property LabelEqual() As string
```

### IndexSearch.LabelExactPhrase Property

Specify a label for the string field where search criteria "Exact Phrase".

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the string field where search criteria \"Exact Phrase\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_XACT)]
public string LabelExactPhrase;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the string field where search criteria \"Exact Phrase\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_XACT)>
Public Property LabelExactPhrase() As string
```

### IndexSearch.LabelGreaterThan Property

Specify label for numeric field where search criteria is "Greater Than".

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for numeric field where search criteria is \"Greater Than\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_GRTER)]
public string LabelGreaterThan;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for numeric field where search criteria is \"Greater Than\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_GRTER)>
Public Property LabelGreaterThan() As string
```

### IndexSearch.Labelinputdate Property

Specify a label for the message that appears when date entered are blank/incorrect.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for message that appears when date entered are blank/incorrect.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_IPUTDT)]
public string Labelinputdate;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for message that appears when date entered are blank/incorrect.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_IPUTDT)>
Public Property Labelinputdate() As string
```

### IndexSearch.Labelinputnumber Property

Specify a label for the message that appears when values entered are blank/incorrect.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify a label for the message that appears when values entered are
blank/incorrect.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_IPUTNUM)]
public string Labelinputnumber;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify a label for the message that appears when values entered are
blank/incorrect.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_IPUTNUM)>
Public Property Labelinputnumber() As string
```

### IndexSearch.LabelLessThan Property

Specify label for numeric field where search criteria "Less Than".

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for numeric field where search criteria is \"Less Than\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_LESS)]
public string LabelLessThan;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for numeric field where search criteria is \"Less Than\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_LESS)>
Public Property LabelLessThan() As string
```

### IndexSearch.Labelon Property

Specify a label for the Date field where search criteria is "On".

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for Date field where search criteria is \"On\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_ON)]
public string Labelon;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for Date field where search criteria is \"On\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_ON)>
Public Property Labelon() As string
```

### IndexSearch.LabelSelect Property

Specify labels for first item on selection dropdown.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify labels for first item on selection dropdown.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_SELECT)]
public string LabelSelect;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify labels for first item on selection dropdown.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_SELECT)>
Public Property LabelSelect() As string
```

### IndexSearch.LinkTarget Property

Gets or sets the target window for the result to be displayed.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
```

```
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

### IndexSearch.MaxResults Property

The total number of results to display. If set to zero, unlimited results will be set.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The total number of results to display. If set to zero, unlimited results will
be set.")]
public int MaxResults;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The total number of results to display. If set to zero, unlimited results will
be set.")>
Public Property MaxResults() As Integer
```

### IndexSearch.OrderBy Property

Specify the order of the results. Choices are:

- **Title** - the title of the content block
- **ID** - the XML content block ID number
- **Date Created** - the date the content block was created
- **Date Modified** - the date the content block was last modified
- **AuthorLName** - the last author's last name
- **AuthorFname** - the last author's first name
- **StartDate** - the GoLive date

**C#**

```
[Category("Ektron"), DefaultValue("Title"), Description("Specify the order of the results.
Choices are: Title, ID, Date Created, Date Modified, AuthorLName, AuthorFName or StartDate")]
public string OrderBy;
```

**Visual Basic**

```
<Category("Ektron"), DefaultValue("Title"), Description("Specify the order of the results.
Choices are: Title, ID, Date Created, Date Modified, AuthorLName, AuthorFName or StartDate")>
Public Property OrderBy() As string
```

### IndexSearch.OrderByDirection Property

The direction the results will be sorted in. This property works with the OrderBy (see page 313) property only. Choices are:

- Ascending
- Descending

**C#**

```
[Category("Ektron"), TypeConverter(typeof(SortOrderTypeConverter)), Description("Specify the
```

```
order direction of the results.")]
public string OrderByDirection;
```

**Visual Basic**

```
<Category("Ektron"), TypeConverter(typeof(SortOrderTypeConverter)), Description("Specify the
order direction of the results.")>
Public Property OrderByDirection() As string
```

### IndexSearch.Recursive Property

Select True if you wish to search the subfolders of the selected FolderPath.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Select True if you wish to search the subfolders of the selected FolderPath.")]
public bool Recursive;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Select True if you wish to search the subfolders of the selected FolderPath.")>
Public Property Recursive() As bool
```

### IndexSearch.SearchParmXML Property

Predefine a set of search criteria.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[Description("Predefine a search criteria.")]
[Editor(typeof(SearchParmColUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public string SearchParmXML;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("Predefine a search criteria.")>
<Editor(typeof(SearchParmColUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property SearchParmXML() As string
```

### IndexSearch.ShowSearchBoxAlways Property

If this property is set to False, the search box will not be visible on PostBack.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to False, the search box will not be visible on
PostBack.")]
public bool ShowSearchBoxAlways;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to False, the search box will not be visible on
PostBack.")>
Public Property ShowSearchBoxAlways() As bool
```

### IndexSearch.SyncRoot Property

Gets an object that can be used to synchronize the collection.

**C#**

```
[Browsable(false)]
public object SyncRoot;
```

**Visual Basic**

```
<Browsable(false)>
Public Property SyncRoot() As object
```

### IndexSearch.Weighted Property

Select True if you wish perform a weighted search.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Select True if you wish to do an OR search between indexed fields.")]
public bool Weighted;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Select True if you wish to do an OR search between indexed fields.")>
Public Property Weighted() As bool
```

### IndexSearch.XmlConfigId Property

ID of the Smart Form (see page 836) whose indexed fields will appear on the search screen.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("ID of the Smart Form whose indexed fields will appear on the search screen.")]
[Editor(typeof(XMLConfigIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long XmlConfigId;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("ID of the Smart Form whose indexed fields will appear on the search screen.")>
<Editor(typeof(XMLConfigIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property XmlConfigId() As long
```

### IndexSearch.XmlDoc Property

Returns the XSLT object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

# Invite Class

### Class Hierarchy

**C#**

```
public class Invite : EkXsltWebPart;
```

**Visual Basic**

```
Public Class Invite
Inherits EkXsltWebPart
```

### File

Invite.cs

## Invite Methods

The methods of the Invite class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 316) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 316) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 316) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## Invite.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## Invite.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

## Invite.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

### Returns

## Invite Properties

The properties of the Invite class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DisplayXslt (see page 317) | The Xslt to use to render the control. (optional) |
| | FriendMessageId (see page 317) | The ID of the message to send in the link section of the friend request email. |
| | InviteMessageId (see page 318) | The ID of the message to send in the link section of the invite email. |
| | OptionalMessageText (see page 318) | This is the default text that will be shown in the optional-message textarea, which can be edited by users. |
| | Stylesheet (see page 318) | This is the default stylesheet used to style the appearance of the control. |
| | XmlDoc (see page 318) | Returns the Xslt object for the collection. |

## Invite.DisplayXslt Property

The Xslt to use to render the control. (optional)

### C#

```
[Category("Ektron")]
[Description("The Xslt to use to render the control. (optional)")]
[TypeConverter(typeof(XsltInviteTypeConverter))]
[DefaultValue("Basic Invite")]
public override string DisplayXslt;
```

### Visual Basic

```
<Category("Ektron")>
<Description("The Xslt to use to render the control. (optional)")>
<TypeConverter(typeof(XsltInviteTypeConverter))>
<DefaultValue("Basic Invite")>
Public override Property DisplayXslt() As string
```

## Invite.FriendMessageId Property

The ID of the message to send in the link section of the friend request email.

### C#

```
[Category("Ektron")]
[Description("The ID of the message to send in the link section of the friend request email
(defined in the workarea), if zero then uses the default message.")]
[TypeConverter(typeof(int))]
[DefaultValue("0")]
public long FriendMessageId;
```

### Visual Basic

```
<Category("Ektron")>
<Description("The ID of the message to send in the link section of the friend request email
(defined in the workarea), if zero then uses the default message.")>
<TypeConverter(typeof(int))>
<DefaultValue("0")>
Public Property FriendMessageId() As long
```

### Invite.InviteMessageId Property

The ID of the message to send in the link section of the invite email.

**C#**

```
[Category("Ektron")]
[Description("The ID of the message to send in the link section of the invite email (defined
in the workarea), if zero then uses the default message.")]
[TypeConverter(typeof(int))]
[DefaultValue("0")]
public long InviteMessageId;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("The ID of the message to send in the link section of the invite email (defined
in the workarea), if zero then uses the default message.")>
<TypeConverter(typeof(int))>
<DefaultValue("0")>
Public Property InviteMessageId() As long
```

### Invite.OptionalMessageText Property

This is the default text that will be shown in the optional-message textarea, which can be edited by users.

**C#**

```
[Category("Ektron")]
[Description("This is the default text that will be shown in the optional-message textarea,
which can be edited by users.")]
[TypeConverter(typeof(string))]
[DefaultValue("Hello, I am sending you this email because I would like you to join me at this
website.")]
public string OptionalMessageText;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("This is the default text that will be shown in the optional-message textarea,
which can be edited by users.")>
<TypeConverter(typeof(string))>
<DefaultValue("Hello, I am sending you this email because I would like you to join me at this
website.")>
Public Property OptionalMessageText() As string
```

### Invite.Stylesheet Property

This is the default stylesheet used to style the appearance of the control.

**C#**

```
[Category("Ektron")]
[Description("This is the default stylesheet used to style the appearance of the control.")]
[TypeConverter(typeof(string))]
[DefaultValue("Community.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("This is the default stylesheet used to style the appearance of the control.")>
<TypeConverter(typeof(string))>
<DefaultValue("Community.css")>
Public Property Stylesheet() As string
```

### Invite.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## LanguageAPI Class

Control that can be used to override the default system language assignment behavior.

**Class Hierarchy**

**C#**

```
public class LanguageAPI : WebControl;
```

**Visual Basic**

```
Public Class LanguageAPI
Inherits WebControl
```

**File**

LanguageAPI.cs

### LanguageAPI.LanguageAPI Constructor

Upon initialization, SiteLanguage (see page 323) is set to DefaultLanguage (see page 321).

**C#**

```
public LanguageAPI();
```

**Visual Basic**

```
Public Sub New()
```

### LanguageAPI Methods

The methods of the LanguageAPI class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 319) | Use this function along with LanguageLogic to implement your custom language control logic. |
| | GetLanguage (see page 320) | Returns a Language string corresponding to a given LanguageID. |
| | GetLanguageID (see page 320) | Returns the LanguageID corresponding to a given Language string. |
| | IsValid (see page 320) | This is the overview for the IsValid method overload. |

### LanguageAPI.Fill Method

Use this function along with LanguageLogic to implement your custom language control logic.

**C#**

```
public virtual void Fill();
```

**Visual Basic**

```
Public virtual Function Fill() As void
```

### LanguageAPI.GetLanguage Method

Returns a Language string corresponding to a given LanguageID.

**C#**

```
public string GetLanguage(int LanguageID);
```

**Visual Basic**

```
Public Function GetLanguage(LanguageID As Integer) As string
```

### LanguageAPI.GetLanguageID Method

Returns the LanguageID corresponding to a given Language string.

**C#**

```
public int GetLanguageID(string Language);
```

**Visual Basic**

```
Public Function GetLanguageID(Language As string) As Integer
```

### IsValid Method

This is the overview for the IsValid method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | LanguageAPI.IsValid (int) (see page 320) | Returns true if the passed LanguageID is an active system language. |
| | LanguageAPI.IsValid (string) (see page 320) | Returns true if the passed Language is an active system language. |

### LanguageAPI.IsValid Method (int)

Returns true if the passed LanguageID is an active system language.

**C#**

```
public bool IsValid(int LanguageID);
```

**Visual Basic**

```
Public Function IsValid(LanguageID As Integer) As bool
```

### LanguageAPI.IsValid Method (string)

Returns true if the passed Language is an active system language.

**C#**

```
public bool IsValid(string Language);
```

**Visual Basic**

```
Public Function IsValid(Language As string) As bool
```

### LanguageAPI Properties

The properties of the LanguageAPI class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | CurrentLanguage (see page 321) | Returns the current language (from a cookie). |
| | CurrentLanguageID (see page 321) | Returns the current language ID (from a cookie). |
| | DefaultLanguage (see page 321) | If not overridden, the system string "Default" is used. |
| | DefaultLanguageID (see page 322) | If not overridden, the ID '0' is used. |
| | LangaugeIdList (see page 322) | Returns a list of all the language IDs that are activated in Ektron CMS400.NET |
| | LangaugeTitleList (see page 322) | Returns a list of all the languages that are activated in Ektron CMS400.NET. |
| | MultiLanguageEnabled (see page 322) | Tells if the site supports multi-language mode.<br>• **True** = Multi-language enabled<br>• **False** = Multi-language not enabled |
| | OverrideDefault (see page 322) | 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 323) is not equal to DefaultLanguage (see page 321). If they are equal, false is returned. |
| | SiteLanguage (see page 323) | Sets the site language of the site. Runs through the IsValid (see page 320) function to verify that the language is an active language in the system. |
| | SiteLanguageID (see page 323) | If not overridden, the ID '0' is used. |

## LanguageAPI.CurrentLanguage Property

Returns the current language (from a cookie).

**C#**

```
[Browsable(false)]
public string CurrentLanguage;
```

**Visual Basic**

```
<Browsable(false)>
Public Property CurrentLanguage() As string
```

## LanguageAPI.CurrentLanguageID Property

Returns the current language ID (from a cookie).

**C#**

```
[Browsable(false)]
public int CurrentLanguageID;
```

**Visual Basic**

```
<Browsable(false)>
Public Property CurrentLanguageID() As Integer
```

## LanguageAPI.DefaultLanguage Property

If not overridden, the system string "Default" is used.

**C#**

```
[Browsable(false)]
public string DefaultLanguage;
```

**Visual Basic**

```
<Browsable(false)>
```

```
Public Property DefaultLanguage() As string
```

### LanguageAPI.DefaultLanguageID Property

If not overridden, the ID '0' is used.

**C#**

```
[Browsable(false)]
public int DefaultLanguageID;
```

**Visual Basic**

```
<Browsable(false)>
Public Property DefaultLanguageID() As Integer
```

### LanguageAPI.LangaugeIdList Property

Returns a list of all the language IDs that are activated in Ektron CMS400.NET

**C#**

```
[Browsable(false)]
public int LangaugeIdList;
```

**Visual Basic**

```
<Browsable(false)>
Public Property LangaugeIdList() As Integer
```

### LanguageAPI.LangaugeTitleList Property

Returns a list of all the languages that are activated in Ektron CMS400.NET.

**C#**

```
[Browsable(false)]
public string LangaugeTitleList;
```

**Visual Basic**

```
<Browsable(false)>
Public Property LangaugeTitleList() As string
```

### LanguageAPI.MultiLanguageEnabled Property

Tells if the site supports multi-language mode.

- **True** = Multi-language enabled
- **False** = Multi-language not enabled

**C#**

```
[Browsable(false)]
public bool MultiLanguageEnabled;
```

**Visual Basic**

```
<Browsable(false)>
Public Property MultiLanguageEnabled() As bool
```

### LanguageAPI.OverrideDefault Property

If you wish to override the default behavior of CMS400.NET, set this to true and implement your logic in the 'LanguageLogic()' overridable method. Currently this property returns the set value only if SiteLanguage (see page 323) is not equal to DefaultLanguage (see page 321). 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 320) function to verify that the language is an active language in the system.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the site language of the site. Runs through the IsValid function to verify
that the language is an active language in the system.")]
[TypeConverter(typeof(LanguageTypeConverter))]
public string SiteLanguage;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the site language of the site. Runs through the IsValid function to verify
that the language is an active language in the system.")>
<TypeConverter(typeof(LanguageTypeConverter))>
Public Property SiteLanguage() As string
```

### LanguageAPI.SiteLanguageID Property

If not overridden, the ID '0' is used.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the site language of the site. Runs through the IsValid function to verify
that the language is an active language in the system.")]
public int SiteLanguageID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the site language of the site. Runs through the IsValid function to verify
that the language is an active language in the system.")>
Public Property SiteLanguageID() As Integer
```

## LanguageSelect Class

The LanguageSelect server control displays a language selection dropdown list on an Ektron CMS400.NET Web page. The dropdown box lists all languages selected on the Settings > Configuration > Language (see page 219) settings screen.

**Class Hierarchy**

**C#**

```
public class LanguageSelect : EkWebControl;
```

**Visual Basic**

```
Public Class LanguageSelect
Inherits EkWebControl
```

### File

LanguageSelect.cs

## LanguageSelect Methods

The methods of the LanguageSelect class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 324) | Method that can be overridden and used to grab data from the back end for each server control. |
| | Wrap (see page 324) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## LanguageSelect.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

### C#

```
public override void Fill();
```

### Visual Basic

```
Public override Function Fill() As void
```

## LanguageSelect.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

### C#

```
public override string Wrap(string Html);
```

### Visual Basic

```
Public override Function Wrap(Html As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| 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 324) | Gets or sets the label next to the language select dropdown box. |
| | Text (see page 325) | 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. |

## LanguageSelect.LabelName Property

Gets or sets the label next to the language select dropdown box.

### C#

```
[Bindable(true)]
[Category("Ektron")]
```

```
[Description("Label next to the language select. The default value is set to
\"res_lngsel_lbl\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_LANGSEL_LBL)]
public string LabelName;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Label next to the language select. The default value is set to
\"res_lngsel_lbl\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_LANGSEL_LBL)>
Public Property LabelName() As string
```

### LanguageSelect.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

**C#**

```
public override string Text;
```

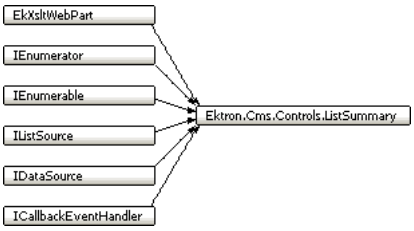
**Visual Basic**

```
Public override Property Text() As string
```

## ListSummary Class

Summary description for ListSummary.

**Class Hierarchy**




**C#**

```
[DefaultProperty("FolderID"), ToolboxData("<{0}:ListSummary runat=server></{0}:ListSummary>")]
public class ListSummary : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("FolderID"), ToolboxData("<{0}:ListSummary runat=server></{0}:ListSummary>")>
Public Class ListSummary
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

**File**

ListSummary.cs

### ListSummary Methods

The methods of the ListSummary class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | CopyTo (see page 326) | 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 326) | Populates the data for the ListSummary (see page 325). |
| | GetEnumerator (see page 326) | Gets a System.Collections.IEnumerator for the ListSummary (see page 325). |
| | GetRssFeed (see page 326) | This method returns the ListSummary (see page 325) RSS feed. |
| | GetView (see page 327) | Gets the view data using dataset. |
| | GetViewNames (see page 327) | Gets the view names. |
| | MoveNext (see page 327) | Move to next index. |
| | Reset (see page 327) | Resets the index. |
| | this (see page 328) | Returns the current ContentItem.ContentRow. |
| | ToString (see page 328) | Overrides the system ToString function. |
| | Wrap (see page 328) | Wrap the Html toolbar. |

## ListSummary.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

**C#**

```
public void CopyTo(Array array, int index);
```

**Visual Basic**

```
Public Function CopyTo(array As Array, index As Integer) As void
```

## ListSummary.Fill Method

Populates the data for the ListSummary (see page 325).

**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 325).

**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 325) 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 788) 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 |
|---|---|---|
| | AddText (see page 329) | Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 788)' could be changed to 'Add Report') |
| | CacheInterval (see page 330) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |

| | | |
|---|---|---|
| | ContentParameter (see page 330) | Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary. |
| | ContentType (see page 331) | Determines the type for content to return. For example, Forms. Choices are:<br>• **AllTypes** - displays all of the content types for the given folder.<br>• **Content (see page 788)** - displays a list of content blocks.<br>• **Forms** - forms appear in the list summary.<br>• **Archive_Content** - archived content blocks appear in the list summary.<br>• **Archive_Forms** - archived forms appear in the list summary.<br>• **Assets** - assets, such as offices documents, appear in the list summary.<br>• **Archive_Assets** - archived assets appear in the list summary.<br>• **LibraryItem** - library items appear in the list summary.<br>• **Multimedia** - multimedia items appear in the list summary.<br>• **Archive_Media** -... more (see page 331) |
| | Count (see page 331) | Indicates the number of elements in your ListSummary (see page 325). |
| | Current (see page 331) | Gets the current index. |
| | Direction (see page 332) | Direction for sorting results, "ascending" or "descending". |
| | EkItems (see page 332) | Returns the array of ContentBase. |
| | EnablePaging (see page 332) | Enables paging if the total records exceeds MaxResult value. |
| | FolderDescription (see page 333) | Returns a description of the given folder ID. |
| | FolderID (see page 333) | The ID of the folder from which to select data. Root is 0. |
| | FolderName (see page 333) | Returns name of the given folder ID. |
| | FolderPath (see page 333) | The path of the folder from which to select data. Root starts at \. |
| | GetHtml (see page 334) | If set to True, retrieves the content (Html body) of every content block in the list summary for display. |
| | IncludeIcons (see page 334) | Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser. |
| | IsSynchronized (see page 334) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | LinkTarget (see page 335) | Gets or Sets the target window for the result. |
| | MarkupLanguage (see page 335) | Enter the name of a markup file that manages the display of this server control. |
| | MaxResults (see page 335) | Maximum number of results to be returned. If set to 0 (zero), there is no limit. |
| | OrderByDirection (see page 335) | Direction (see page 332) for sorting results, "ascending" or "descending". |
| | OrderKey (see page 336) | The key by which to sort your results. |
| | Random (see page 336) | If set to true, grabs a single content block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the list summary. |
| | Recursive (see page 336) | Determines if subfolders of the selected folders should be included in the list. |
| | SuppressWrapperTags (see page 337) | Suppresses the output of span/div tags around the control. |
| | SyncRoot (see page 337) | Gets an object that can be used to synchronize the ListSummary (see page 325). |
| | XmlDoc (see page 337) | Returns the Xml object loaded for ListSummary (see page 325). |

### ListSummary.AddText Property

Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 788)' could be changed to 'Add Report')

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
```

```
[Description("Allows overriding the derived controls default Add-menu button text (i.e. 'Add
HTML Content' could be changed to 'Add Report')")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Visible)
]
[Browsable(true)]
public virtual string AddText;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows overriding the derived controls default Add-menu button text (i.e. 'Add
HTML Content' could be changed to 'Add Report')")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Visible)
>
<Browsable(true)>
Public virtual Property AddText() As string
```

### ListSummary.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

### ListSummary.ContentParameter Property

Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and replaces the list summary with that
content block when specified. Leave blank to always display the list summary.")]
[Personalizable(), WebBrowsable()]
public string ContentParameter;
```

**Visual Basic**

```
<Bindable(true)>
```

```
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and replaces the list summary with that
content block when specified. Leave blank to always display the list summary.")>
<Personalizable(), WebBrowsable()>
Public Property ContentParameter() As string
```

### ListSummary.ContentType Property

Determines the type for content to return. For example, Forms. Choices are:

- **AllTypes** - displays all of the content types for the given folder.
- **Content (see page 788)** - displays a list of content blocks.
- **Forms** - forms appear in the list summary.
- **Archive_Content** - archived content blocks appear in the list summary.
- **Archive_Forms** - archived forms appear in the list summary.
- **Assets** - assets, such as offices documents, appear in the list summary.
- **Archive_Assets** - archived assets appear in the list summary.
- **LibraryItem** - library items appear in the list summary.
- **Multimedia** - multimedia items appear in the list summary.
- **Archive_Media** - archived multimedia items appear in the list summary.
- **NonLibraryContent** - all types of content appear in the list summary except for library items.
- **ImageContent** - images appear in the list summary.
- **DiscussionTopic** - forum topics appear in the list summary.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Detemines the type for content to return. For example, forms.")]
[Personalizable(), WebBrowsable()]
public Ektron.Cms.Controls.CmsWebService.CMSContentType ContentType;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Detemines the type for content to return. For example, forms.")>
<Personalizable(), WebBrowsable()>
Public Property ContentType() As Ektron.Cms.Controls.CmsWebService.CMSContentType
```

### ListSummary.Count Property

Indicates the number of elements in your ListSummary (see page 325).

**C#**

```
[Browsable(false)]
public int Count;
```

**Visual Basic**

```
<Browsable(false)>
Public Property Count() As Integer
```

### ListSummary.Current Property

Gets the current index.

**C#**

```
[Browsable(false)]
public object Current;
```

**Visual Basic**

```
<Browsable(false)>
Public Property Current() As object
```

### ListSummary.Direction Property

Direction for sorting results, "ascending" or "descending".

**C#**

```
[Obsolete("Use OrderByDirection Property")]
[Browsable(false)]
[Bindable(true)]
[Category("Ektron")]
[Description("The direction (ascending or descending) that you wish your sorted results to be
displayed.")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")]
public new Ektron.Cms.Controls.CmsWebService.OrderByDirection Direction;
```

**Visual Basic**

```
<Obsolete("Use OrderByDirection Property")>
<Browsable(false)>
<Bindable(true)>
<Category("Ektron")>
<Description("The direction (ascending or descending) that you wish your sorted results to be
displayed.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Direction"),
WebDescription("Direction")>
Public new Property Direction() As Ektron.Cms.Controls.CmsWebService.OrderByDirection
```

### ListSummary.EkItems Property

Returns the array of ContentBase.

**C#**

```
[Browsable(false)]
public Ektron.Cms.Common.ContentBase EkItems;
```

**Visual Basic**

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

### ListSummary.EnablePaging Property

Enables paging if the total records exceeds MaxResult value.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enables paging if total records exceeds MaxResult value. Use this property with
non zero value of Maxresults.")]
[Personalizable(), WebBrowsable()]
public bool EnablePaging;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
```

```
<DefaultValue(0)>
<Description("Enables paging if total records exceeds MaxResult value. Use this property with
non zero value of Maxresults.")>
<Personalizable(), WebBrowsable()>
Public Property EnablePaging() As bool
```

### ListSummary.FolderDescription Property

Returns a description of the given folder ID.

**C#**

```
public string FolderDescription;
```

**Visual Basic**

```
Public Property FolderDescription() As string
```

### ListSummary.FolderID Property

The ID of the folder from which to select data. Root is 0.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder from which to select data. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long FolderID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder from which to select data. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property FolderID() As long
```

### ListSummary.FolderName Property

Returns name of the given folder ID.

**C#**

```
public string FolderName;
```

**Visual Basic**

```
Public Property FolderName() As string
```

### ListSummary.FolderPath Property

The path of the folder from which to select data. Root starts at \.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Browsable(false)]
[Description("The path of the folder from which to select data. Root starts at \"\\\".")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Obsolete()]
public string FolderPath;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Browsable(false)>
<Description("The path of the folder from which to select data. Root starts at \"\\\".")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Obsolete()>
Public Property FolderPath() As string
```

### ListSummary.GetHtml Property

If set to True, retrieves the content (Html body) of every content block in the list summary for display.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to True, retrieves the content (Html body) of every content block in the
list summary for display.")]
[Personalizable(), WebBrowsable()]
public bool GetHtml;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to True, retrieves the content (Html body) of every content block in the
list summary for display.")>
<Personalizable(), WebBrowsable()>
Public Property GetHtml() As bool
```

### ListSummary.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.

**C#**

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
[Personalizable(), WebBrowsable()]
public bool IncludeIcons;
```

**Visual Basic**

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
<Personalizable(), WebBrowsable()>
Public Property IncludeIcons() As bool
```

### ListSummary.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

```
[Browsable(false)]
public bool IsSynchronized;
```

**Visual Basic**

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

### ListSummary.LinkTarget Property

Gets or Sets the target window for the result.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

### ListSummary.MarkupLanguage Property

Enter the name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter the name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter the name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

### ListSummary.MaxResults Property

Maximum number of results to be returned. If set to 0 (zero), there is no limit.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
[Personalizable(), WebBrowsable()]
public int MaxResults;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable(), WebBrowsable()>
Public Property MaxResults() As Integer
```

### ListSummary.OrderByDirection Property

Direction (see page 332) for sorting results, "ascending" or "descending".

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(EkEnumeration.OrderByDirection.Ascending)]
[Description("Direction for sorting results, 'ascending' or 'descending'.")]
public Ektron.Cms.Common.EkEnumeration.OrderByDirection OrderByDirection;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(EkEnumeration.OrderByDirection.Ascending)>
<Description("Direction for sorting results, 'ascending' or 'descending'.")>
Public Property OrderByDirection() As Ektron.Cms.Common.EkEnumeration.OrderByDirection
```

### ListSummary.OrderKey Property

The key by which to sort your results.

**C#**

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Controls.CmsWebService.TeasersOrderBy.Title)]
[Description("The key by which to sort your results.")]
[Category("Ektron")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")]
public Ektron.Cms.Controls.CmsWebService.TeasersOrderBy OrderKey;
```

**Visual Basic**

```
<Bindable(true)>
<DefaultValue(Ektron.Cms.Controls.CmsWebService.TeasersOrderBy.Title)>
<Description("The key by which to sort your results.")>
<Category("Ektron")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")>
Public Property OrderKey() As Ektron.Cms.Controls.CmsWebService.TeasersOrderBy
```

### ListSummary.Random Property

If set to true, grabs a single content block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the list summary.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Set this property to True to grab a single, random piece of content from this
list summary.")]
[Personalizable(), WebBrowsable()]
public bool Random;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Set this property to True to grab a single, random piece of content from this
list summary.")>
<Personalizable(), WebBrowsable()>
Public Property Random() As bool
```

### ListSummary.Recursive Property

Determines if subfolders of the selected folders should be included in the list.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set recursive to True if you wish to display results in subfolders of your
FolderPath.")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Recursive"), WebDescription("Recursive:
")]
public bool Recursive;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set recursive to True if you wish to display results in subfolders of your
FolderPath.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Recursive"), WebDescription("Recursive:
")>
Public Property Recursive() As bool
```

### ListSummary.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

**C#**

```
public override bool SuppressWrapperTags;
```

**Visual Basic**

```
Public override Property SuppressWrapperTags() As bool
```

### ListSummary.SyncRoot Property

Gets an object that can be used to synchronize the ListSummary (see page 325).

**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 325).

**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

## Login Methods

The methods of the Login class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 338) | Method that can be overridden and used to grab data from the back end for each server control. |

## Login.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## Login Properties

The properties of the Login class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AutoAddType (see page 338) | Set this property to define auto-added Active Directory (see page 189) users as Members or Authors. |
| | AutoLogin (see page 339) | Set this property to True if you want to use Auto Login (see page 337) with Active Directory (see page 189). |
| | OnlyAllowMemberLogin (see page 339) | Set this property to True if you only allow membership users. |
| | Password (see page 339) | Write Only value for passing in the password. |
| | PromptLogout (see page 339) | Set this property to False if you do not want to prompt users to verify logout. |
| | SuppressHelpButton (see page 339) | Set this property to True if you don't want to show the Help button. |
| | Text (see page 340) | 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 340) | Write Only value for passing in the UserName. |

## Login.AutoAddType Property

Set this property to define auto-added Active Directory (see page 189) users as Members or Authors.

**C#**

```
[Bindable(false), Category("Ektron"), Description("Set this property to define auto-added
Active Directory users as Members or Authors."),
DefaultValue(Common.EkEnumeration.AutoAddUserTypes.Author)]
public Common.EkEnumeration.AutoAddUserTypes AutoAddType;
```

**Visual Basic**

```
<Bindable(false), Category("Ektron"), Description("Set this property to define auto-added
Active Directory users as Members or Authors."),
DefaultValue(Common.EkEnumeration.AutoAddUserTypes.Author)>
Public Property AutoAddType() As Common.EkEnumeration.AutoAddUserTypes
```

### Login.AutoLogin Property

Set this property to True if you want to use Auto Login (see page 337) with Active Directory (see page 189).

**C#**

```
[Bindable(false), Category("Ektron"), Description("Set this property to True if you want to
use Auto Login with Active Directory."), DefaultValue(false)]
public bool AutoLogin;
```

**Visual Basic**

```
<Bindable(false), Category("Ektron"), Description("Set this property to True if you want to
use Auto Login with Active Directory."), DefaultValue(false)>
Public Property AutoLogin() As bool
```

### Login.OnlyAllowMemberLogin Property

Set this property to True if you only allow membership users.

**C#**

```
[Bindable(false), Category("Ektron"), Description("Set this property to True if you only allow
membership users."), DefaultValue(false)]
public bool OnlyAllowMemberLogin;
```

**Visual Basic**

```
<Bindable(false), Category("Ektron"), Description("Set this property to True if you only allow
membership users."), DefaultValue(false)>
Public Property OnlyAllowMemberLogin() As bool
```

### Login.Password Property

Write Only value for passing in the password.

**C#**

```
public string Password;
```

**Visual Basic**

```
Public Property Password() As string
```

### Login.PromptLogout Property

Set this property to False if you do not want to prompt users to verify logout.

**C#**

```
[Bindable(false), Category("Ektron"), Description("Set this property to False if you do not
want to prompt users to verify logout."), DefaultValue(true)]
public bool PromptLogout;
```

**Visual Basic**

```
<Bindable(false), Category("Ektron"), Description("Set this property to False if you do not
want to prompt users to verify logout."), DefaultValue(true)>
Public Property PromptLogout() As bool
```

### Login.SuppressHelpButton Property

Set this property to True if you don't want to show the Help button.

**C#**

```
[Bindable(false), Category("Ektron"), Description("Set this property to True if you don't want
to show the Help button."), DefaultValue(false)]
public bool SuppressHelpButton;
```

**Visual Basic**

```
<Bindable(false), Category("Ektron"), Description("Set this property to True if you don't want
to show the Help button."), DefaultValue(false)>
Public Property SuppressHelpButton() As bool
```

### Login.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

**C#**

```
public override string Text;
```

**Visual Basic**

```
Public override Property Text() As string
```

### Login.UserName Property

Write Only value for passing in the UserName.

**C#**

```
public string UserName;
```

**Visual Basic**

```
Public Property UserName() As string
```

## Map Class

Map control

**Class Hierarchy**

**C#**

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:Map runat=server></{0}:Map>")]
public class Map : EkWebControl, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:Map runat=server></{0}:Map>")>
Public Class Map
Inherits EkWebControl
Inherits ICallbackEventHandler
```

**File**

Map.cs

### Map Methods

The methods of the Map class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 341) | |
| | GetCallbackResult (see page 341) | |
| | RaiseCallbackEvent (see page 341) | |

## Map.Fill Method

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## Map.GetCallbackResult Method

**C#**

```
public string GetCallbackResult();
```

**Visual Basic**

```
Public Function GetCallbackResult() As string
```

## Map.RaiseCallbackEvent Method

**C#**

```
public void RaiseCallbackEvent(string eventArgs);
```

**Visual Basic**

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

### Parameters

| Parameters | Description |
|---|---|
| eventArgs | |

## Map Properties

The properties of the Map class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Address (see page 342) | A default address while loading control. If this value is set, the Latitude (see page 345) and Longitude (see page 346) properties are ignored. |
| | ContentId (see page 343) | Use this property in conjunction with the EnableSearchResult (see page 344) property to limit the map to a single content item. Here, you identify the content item to be mapped. |
| | CustomIcon (see page 343) | Custom icon for the category. |
| | DateSearch (see page 343) | Enable or disable DateTime selector for map data. |
| | DisplayTextResult (see page 343) | Enable or disable text output for the search results. |
| | DistanceUnit (see page 344) | Distance measurement units in Miles or Kilometers. |

| | | |
|---|---|---|
| | EnableSearchResult (see page 344) | Determine if the Map (see page 340) server control accesses the CMS search to return results.<br>• **True** = Map (see page 340) server control accesses the CMS search to return results.<br>• **False** = Display a map with only one content item, that is, a single location. Specify this item at the ContentID property. In this case, the map does not find other CMS content whose longitude and latitude are within the map's boundaries.<br>True is the default.<br>Site (see page 694) visitors viewing the map can still use the Search (see page 455) and Directions tabs to get directions to the location.<br>For example, your site features homes for sale, and you... more (see page 344) |
| | FolderId (see page 344) | The ID of the folder from which to search map data. |
| | ForceCategory (see page 345) | When set to true, this property causes the map to only show content associated with the taxonomy category defined in the MapCategory (see page 346) property. When set to false, the map shows all content within the maps boundaries.<br>For example, if you have five content blocks that appear on a map and three of them are associated with a taxonomy, set this property to true and the MapCategory (see page 346) property to the ID of the taxonomy. When a user views the map, it will only display the three content items associated with the taxonomy. |
| | GeoControl (see page 345) | Enable or disable GeoControl which allows a user to perform a search and get directions. |
| | Latitude (see page 345) | Startup latitude value while loading control. |
| | Longitude (see page 346) | Startup longitude value while loading control. |
| | MapCategory (see page 346) | Category to perform search. This category must be created through the taxonomy wizard. |
| | MapProvider (see page 346) | Provider to connect map server. The supported providers are VirtualEarth and Google. |
| | MapStyle (see page 346) | Map (see page 340) style accepted values are "Hybrid", "Road" and "Satellite" |
| | MarkupLanguage (see page 347) | Name of a markup file that manages the display of this server control. |
| | MinZoomLevel (see page 347) | Search (see page 455) will not be performed beyond this zoom level. |
| | PageSize (see page 347) | Page Size to populate search result. |
| | Recursive (see page 347) | Enable or disable recursive search from the specified FolderId (see page 344). |
| | SearchData (see page 348) | Sets source for the search result. |
| | SelectedCategoryLogicalType (see page 348) | Logical type operators to search through the selected map categories. |
| | StartZoomLevel (see page 348) | Startup zoom level. Accepted values between 1 and 19. |
| | Stylesheet (see page 348) | Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | TemplateParamName (see page 349) | Gets or sets a Querystring parameter for the ID of a user. |
| | TemplateUserProfile (see page 349) | Sets the link for user profile. |
| | TypeControl (see page 349) | Enables or disables the Google type control and the zoom, direction and type control for Virtual Earth. |
| | ZoomControl (see page 349) | Enable or Disable zoom control for Google map only. |

## Map.Address Property

A default address while loading control. If this value is set, the Latitude (see page 345) and Longitude (see page 346) properties are ignored.

**C#**

```
[Bindable(true), Category("Ektron")]
[Description("A default address while loading control. If this value is set, the Latitude and
Longitude properties are ignored.")]
[Personalizable(), WebBrowsable()]
public string Address;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron")>
<Description("A default address while loading control. If this value is set, the Latitude and
Longitude properties are ignored.")>
<Personalizable(), WebBrowsable()>
Public Property Address() As string
```

### Map.ContentId Property

Use this property in conjunction with the EnableSearchResult (see page 344) property to limit the map to a single content item. Here, you identify the content item to be mapped.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("Gets or Sets the Default ContentID for this Control.")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long ContentId;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("Gets or Sets the Default ContentID for this Control.")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property ContentId() As long
```

### Map.CustomIcon Property

Custom icon for the category.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(false)]
[Description("Custom icon for the category.")]
public bool CustomIcon;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(false)>
<Description("Custom icon for the category.")>
Public Property CustomIcon() As bool
```

### Map.DateSearch Property

Enable or disable DateTime selector for map data.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(false)]
[Description("Event time selector for map data.")]
public bool DateSearch;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(false)>
<Description("Event time selector for map data.")>
Public Property DateSearch() As bool
```

### Map.DisplayTextResult Property

Enable or disable text output for the search results.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enable or disable text output for the search results.")]
[Personalizable(), WebBrowsable()]
public bool DisplayTextResult;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or disable text output for the search results.")>
<Personalizable(), WebBrowsable()>
Public Property DisplayTextResult() As bool
```

### Map.DistanceUnit Property

Distance measurement units in Miles or Kilometers.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(Unit.Miles)]
[Description("Distance mesurement units in Miles or Kilometers.")]
public Unit DistanceUnit;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(Unit.Miles)>
<Description("Distance mesurement units in Miles or Kilometers.")>
Public Property DistanceUnit() As Unit
```

### Map.EnableSearchResult Property

Determine if the Map (see page 340) server control accesses the CMS search to return results.

- **True** = Map (see page 340) server control accesses the CMS search to return results.
- **False** = Display a map with only one content item, that is, a single location. Specify this item at the ContentID property. In this case, the map does not find other CMS content whose longitude and latitude are within the map's boundaries.

True is the default.

Site (see page 694) visitors viewing the map can still use the Search (see page 455) and Directions tabs to get directions to the location.

For example, your site features homes for sale, and you want a map to show a featured home of the week. To do so, set this property to False, and enter the content block that describes that home in the ContentID field.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enable or disable catalog search.")]
public bool EnableSearchResult;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or disable catalog search.")>
Public Property EnableSearchResult() As bool
```

### Map.FolderId Property

The ID of the folder from which to search map data.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder from which to search map data.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
```

```
public long FolderId;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder from which to search map data.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderId() As long
```

### Map.ForceCategory Property

When set to true, this property causes the map to only show content associated with the taxonomy category defined in the MapCategory (see page 346) property. When set to false, the map shows all content within the maps boundaries.

For example, if you have five content blocks that appear on a map and three of them are associated with a taxonomy, set this property to true and the MapCategory (see page 346) property to the ID of the taxonomy. When a user views the map, it will only display the three content items associated with the taxonomy.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(false)]
[Description("It will force the category filter to MapCategory being applied on startup and on
clearall.")]
public bool ForceCategory;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(false)>
<Description("It will force the category filter to MapCategory being applied on startup and on
clearall.")>
Public Property ForceCategory() As bool
```

### Map.GeoControl Property

Enable or disable GeoControl which allows a user to perform a search and get directions.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enable or disable GeoControl which allows a user to perform a search and get
directions.")]
public bool GeoControl;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or disable GeoControl which allows a user to perform a search and get
directions.")>
Public Property GeoControl() As bool
```

### Map.Latitude Property

Startup latitude value while loading control.

**C#**

```
[Bindable(true), Category("Ektron")]
[Description("Startup latitude value while loading control.")]
[Personalizable(), WebBrowsable()]
public string Latitude;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron")>
<Description("Startup latitude value while loading control.")>
<Personalizable(), WebBrowsable()>
Public Property Latitude() As string
```

### Map.Longitude Property

Startup longitude value while loading control.

**C#**

```
[Bindable(true), Category("Ektron")]
[Description("Startup longitude value while loading control.")]
[Personalizable(), WebBrowsable()]
public string Longitude;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron")>
<Description("Startup longitude value while loading control.")>
<Personalizable(), WebBrowsable()>
Public Property Longitude() As string
```

### Map.MapCategory Property

Category to perform search. This category must be created through the taxonomy wizard.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(0)]
[Description("Category to perform search. This category must be created through the taxonomy
wizard.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long MapCategory;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(0)>
<Description("Category to perform search. This category must be created through the taxonomy
wizard.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property MapCategory() As long
```

### Map.MapProvider Property

Provider to connect map server. The supported providers are VirtualEarth and Google.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(Provider.VirtualEarth)]
[Description("Provider to connect map server. The supported providers are VirtualEarth and
Google.")]
[Personalizable(), WebBrowsable()]
public Provider MapProvider;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(Provider.VirtualEarth)>
<Description("Provider to connect map server. The supported providers are VirtualEarth and
Google.")>
<Personalizable(), WebBrowsable()>
Public Property MapProvider() As Provider
```

### Map.MapStyle Property

Map (see page 340) style accepted values are "Hybrid", "Road" and "Satellite"

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(Style.Road)]
[Description("Map style accepted values are \"Hybrid\", \"Road\" and \"Satellite\"")]
public Style MapStyle;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(Style.Road)>
<Description("Map style accepted values are \"Hybrid\", \"Road\" and \"Satellite\"")>
Public Property MapStyle() As Style
```

### Map.MarkupLanguage Property

Name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("")]
[Description("Enter the name of a markup file that manages the display of this server
control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("")>
<Description("Enter the name of a markup file that manages the display of this server
control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

### Map.MinZoomLevel Property

Search (see page 455) will not be performed beyond this zoom level.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(1)]
[Description("Search will not be performed beyond this zoom level.")]
[Personalizable(), WebBrowsable()]
public int MinZoomLevel;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(1)>
<Description("Search will not be performed beyond this zoom level.")>
<Personalizable(), WebBrowsable()>
Public Property MinZoomLevel() As Integer
```

### Map.PageSize Property

Page Size to populate search result.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(10)]
[Description("Page Size to populate search result.")]
[Personalizable(), WebBrowsable()]
public int PageSize;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(10)>
<Description("Page Size to populate search result.")>
<Personalizable(), WebBrowsable()>
Public Property PageSize() As Integer
```

### Map.Recursive Property

Enable or disable recursive search from the specified FolderId (see page 344).

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enable or disable recursive search from the specified FolderId.")]
```

```
public bool Recursive;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or disable recursive search from the specified FolderId.")>
Public Property Recursive() As bool
```

### Map.SearchData Property

Sets source for the search result.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(MapData.Content)]
[Description("Sets source for the search result.")]
[Personalizable(), WebBrowsable()]
public MapData SearchData;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(MapData.Content)>
<Description("Sets source for the search result.")>
<Personalizable(), WebBrowsable()>
Public Property SearchData() As MapData
```

### Map.SelectedCategoryLogicalType Property

Logical type operators to search through the selected map categories.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(LogicalType.Or)]
[Description("Logical type operators to search through the selected map categories.")]
public LogicalType SelectedCategoryLogicalType;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(LogicalType.Or)>
<Description("Logical type operators to search through the selected map categories.")>
Public Property SelectedCategoryLogicalType() As LogicalType
```

### Map.StartZoomLevel Property

Startup zoom level. Accepted values between 1 and 19.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(13)]
[Description("Startup zoom level. Accepted values between 1 and 19.")]
[Personalizable(), WebBrowsable()]
public int StartZoomLevel;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(13)>
<Description("Startup zoom level. Accepted values between 1 and 19.")>
<Personalizable(), WebBrowsable()>
Public Property StartZoomLevel() As Integer
```

### Map.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

**C#**

```
[Category("Ektron")]
[Description("Allows selecting a different CSS file for the map server control.")]
[TypeConverter(typeof(string))]
[DefaultValue("csslib/map.css")]
```

```
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Allows selecting a different CSS file for the map server control.")>
<TypeConverter(typeof(string))>
<DefaultValue("csslib/map.css")>
Public Property Stylesheet() As string
```

### Map.TemplateParamName Property

Gets or sets a Querystring parameter for the ID of a user.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Gets or sets a Querystring parameter for the ID of a user.")]
[DefaultValue("id")]
public string TemplateParamName;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Gets or sets a Querystring parameter for the ID of a user.")>
<DefaultValue("id")>
Public Property TemplateParamName() As string
```

### Map.TemplateUserProfile Property

Sets the link for user profile.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("")]
[Description("Sets the link for user profile.")]
public string TemplateUserProfile;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("")>
<Description("Sets the link for user profile.")>
Public Property TemplateUserProfile() As string
```

### Map.TypeControl Property

Enables or disables the Google type control and the zoom, direction and type control for Virtual Earth.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enables or disables the Google type control and the zoom, direction and type
control for Virtual Earth.")]
public bool TypeControl;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enables or disables the Google type control and the zoom, direction and type
control for Virtual Earth.")>
Public Property TypeControl() As bool
```

### Map.ZoomControl Property

Enable or Disable zoom control for Google map only.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
```

```
[Description("Enable or Disable zoom control for Google map only.")]
public bool ZoomControl;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or Disable zoom control for Google map only.")>
Public Property ZoomControl() As bool
```

## Membership Class

The Membership server control creates a form on a Web page that allows a user to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user properties and any custom user properties that exist. It also provides a way to have new membership users read and accept the terms and conditions for using Discussion Boards (Forums) when needed.

**Class Hierarchy**

**C#**

```
[DefaultProperty("Text"), ToolboxData("<{0}:Membership runat=server></{0}:Membership>")]
public class Membership : EkWebControl, INamingContainer;
```

**Visual Basic**

```
<DefaultProperty("Text"), ToolboxData("<{0}:Membership runat=server></{0}:Membership>")>
Public Class Membership
Inherits EkWebControl
Inherits INamingContainer
```

**File**

membership.cs

### Membership Methods

The methods of the Membership class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 350) | Method that can be overridden and used to grab data from the back end for each server control. |

### Membership.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

### Membership Properties

The properties of the Membership class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AllowAvatarUpload (see page 351) | Type of Avatar Upload displayed. Choose from the following *__Active__- Allow users to upload avatar *__(Path To Custom Template)__- Create your own custom template with images and make users to select avatars *__Disable__- Disable avatar upload |
| | AutoLogin (see page 352) | Set this property to True to automatically log-in after the registration is complete. |
| | BoardID (see page 352) | Specify the Forum (see page 256) ID to use for the terms and conditions text. |
| | DisplayMode (see page 352) | Type of membership form displayed. Choose from the following:<br>• **UserRegistration** - lets a site visitor register as a membership user.<br>• **ResetPassword** - lets a membership user reset his password.<br>• **UnsubscribeSecured** - lets a membership user unsubscribe. He needs to enter his username and password.<br>• **UnsubscribeUnsecured** - lets a membership user unsubscribe by entering his membership username only.<br>• **AccountActivate** - let a site visitor activate his membership account by entering the ID number in the account verification email. |
| | DynamicParameter (see page 353) | To make this control dynamic, select id. When you do, this server control is attached to the user passed as a URL parameter. |
| | JavascriptEditorHeight (see page 353) | Set the height in pixels for the Signature editor. The default is 300. |
| | JavascriptEditorWidth (see page 353) | Set the width in pixels for the signature editor. The default is 360. |
| | RedirectFailedURL (see page 354) | Redirect the user if registering to the site fails. |
| | RedirectSuccessURL (see page 354) | Redirect the user after successfully registering to the site. |
| | RegisterButtonImg (see page 354) | Specify the image to use instead of the register button. The image path should be relative to the root. |
| | RegisterButtonText (see page 354) | Specify the form submit button text. |
| | ResetButtonImg (see page 355) | Specify the image to use instead of the reset button. The image path should be relative to the root. |
| | ResetButtonText (see page 355) | Specify the form reset button text. |
| | ResetPasswordMessageId (see page 355) | To make this control dynamic, select id. When you do, this server control is attached to the user passed as a URL parameter. |
| | SaveButtonImg (see page 355) | Specify the image to use instead of the save button. The image path should be relative to the root. |
| | SaveButtonText (see page 356) | Specify the form submit button text when logged in. |
| | ShowExtended (see page 356) | Include user's custom properties defined in the CMS. The default value is "True". |
| | ShowTerms (see page 356) | Display the Terms and Conditions text. The default value is "False". |
| | TaxonomyId (see page 356) | Enter the ID number of the taxonomy or category to appear in this server control. Root is 0. |
| | Text (see page 357) | 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 357) | User (see page 913) already exists error message. |
| | UserID (see page 357) | Specify the user ID for which to retrieve information. |
| | UserSuccessMessage (see page 357) | User (see page 913) successfully registered message. |
| | UserUpdateSuccessMessage (see page 358) | Message explaining the user successfully update his/her information. |

## Membership.AllowAvatarUpload Property

Type of Avatar Upload displayed. Choose from the following *__Active__- Allow users to upload avatar *__(Path To Custom__

**Template)**- Create your own custom template with images and make users to select avatars *__Disable__- Disable avatar upload

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The path to use to render the Avatar Template. (optional)")]
[TypeConverter(typeof(AllowAvatarUploadTypeConverter))]
[DefaultValue("Active")]
public string AllowAvatarUpload;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The path to use to render the Avatar Template. (optional)")>
<TypeConverter(typeof(AllowAvatarUploadTypeConverter))>
<DefaultValue("Active")>
Public Property AllowAvatarUpload() As string
```

### Membership.AutoLogin Property

Set this property to True to automatically log-in after the registration is complete.

**C#**

```
[Bindable(false), Category("Ektron"), Description("Set this property to True to automatically
log-in after the registration is complete."), DefaultValue(true)]
public bool AutoLogin;
```

**Visual Basic**

```
<Bindable(false), Category("Ektron"), Description("Set this property to True to automatically
log-in after the registration is complete."), DefaultValue(true)>
Public Property AutoLogin() As bool
```

### Membership.BoardID Property

Specify the Forum (see page 256) ID to use for the terms and conditions text.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the Forum for which to show terms and conditions.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BoardID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the Forum for which to show terms and conditions.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As long
```

### Membership.DisplayMode Property

Type of membership form displayed. Choose from the following:

- **UserRegistration** - lets a site visitor register as a membership user.
- **ResetPassword** - lets a membership user reset his password.
- **UnsubscribeSecured** - lets a membership user unsubscribe. He needs to enter his username and password.
- **UnsubscribeUnsecured** - lets a membership user unsubscribe by entering his membership username only.
- **AccountActivate** - let a site visitor activate his membership account by entering the ID number in the account verification

email.

**C#**

```
[Bindable(true), Category("Ektron"), Description("Choose the control's display mode. Each mode
has it own UI and behaves differently. Choices are: UserRegistration, ResetPassword,
UnsubscribeSecured, UnsubscribeUnsecured or AccountActivate."), DefaultValue(0)]
public Mode DisplayMode;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Choose the control's display mode. Each mode
has it own UI and behaves differently. Choices are: UserRegistration, ResetPassword,
UnsubscribeSecured, UnsubscribeUnsecured or AccountActivate."), DefaultValue(0)>
Public Property DisplayMode() As Mode
```

### Membership.DynamicParameter Property

To make this control dynamic, select id. When you do, this server control is attached to the user passed as a URL parameter.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("To make this control dynamic, select id. When you do, this server control is
attached to the user passed as a URL parameter.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("To make this control dynamic, select id. When you do, this server control is
attached to the user passed as a URL parameter.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

### Membership.JavascriptEditorHeight Property

Set the height in pixels for the Signature editor. The default is 300.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(300)]
[Description("The pixel height for the Signature Editor. The default is 300.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public int JavascriptEditorHeight;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(300)>
<Description("The pixel height for the Signature Editor. The default is 300.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorHeight() As Integer
```

### Membership.JavascriptEditorWidth Property

Set the width in pixels for the signature editor. The default is 360.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
```

```
[DefaultValue(360)]
[Description("The pixel width for the Signature editor. The default is 360.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public int JavascriptEditorWidth;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(360)>
<Description("The pixel width for the Signature editor. The default is 360.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorWidth() As Integer
```

### Membership.RedirectFailedURL Property

Redirect the user if registering to the site fails.

**C#**

```
[Bindable(true), Category("Ektron"), Description("Redirect the user if registering to the site
fails."), DefaultValue("")]
public string RedirectFailedURL;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Redirect the user if registering to the site
fails."), DefaultValue("")>
Public Property RedirectFailedURL() As string
```

### Membership.RedirectSuccessURL Property

Redirect the user after successfully registering to the site.

**C#**

```
[Bindable(true), Category("Ektron"), Description("Redirect the user after successfully
registering to the site."), DefaultValue("")]
public string RedirectSuccessURL;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Redirect the user after successfully
registering to the site."), DefaultValue("")>
Public Property RedirectSuccessURL() As string
```

### Membership.RegisterButtonImg Property

Specify the image to use instead of the register button. The image path should be relative to the root.

**C#**

```
[Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
register button. The image path should be relative to the root."), DefaultValue("")]
public string RegisterButtonImg;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
register button. The image path should be relative to the root."), DefaultValue("")>
Public Property RegisterButtonImg() As string
```

### Membership.RegisterButtonText Property

Specify the form submit button text.

**C#**

```
[Bindable(true), Category("Ektron"), Description("Specify the form submit button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_REGBTN)]
```

```
public string RegisterButtonText;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Specify the form submit button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_REGBTN)>
Public Property RegisterButtonText() As string
```

### Membership.ResetButtonImg Property

Specify the image to use instead of the reset button. The image path should be relative to the root.

**C#**

```
[Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
reset button. The image path should be relative to the root."), DefaultValue("")]
public string ResetButtonImg;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
reset button. The image path should be relative to the root."), DefaultValue("")>
Public Property ResetButtonImg() As string
```

### Membership.ResetButtonText Property

Specify the form reset button text.

**C#**

```
[Bindable(true), Category("Ektron"), Description("Specify the form reset button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_RESTBTN)]
public string ResetButtonText;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Specify the form reset button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_RESTBTN)>
Public Property ResetButtonText() As string
```

### Membership.ResetPasswordMessageId Property

To make this control dynamic, select id. When you do, this server control is attached to the user passed as a URL parameter.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The id of the message to send out for the reset password email.")]
[DefaultValue(0)]
public long ResetPasswordMessageId;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The id of the message to send out for the reset password email.")>
<DefaultValue(0)>
Public Property ResetPasswordMessageId() As long
```

### Membership.SaveButtonImg Property

Specify the image to use instead of the save button. The image path should be relative to the root.

**C#**

```
[Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the save
button. The image path should be relative to the root."), DefaultValue("")]
```

```
public string SaveButtonImg;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the save
button. The image path should be relative to the root."), DefaultValue("")>
Public Property SaveButtonImg() As string
```

### Membership.SaveButtonText Property

Specify the form submit button text when logged in.

**C#**

```
[Bindable(true), Category("Ektron"), Description("Specify the form submit button text when
logged in."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_SAVBTN)]
public string SaveButtonText;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Specify the form submit button text when
logged in."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_SAVBTN)>
Public Property SaveButtonText() As string
```

### Membership.ShowExtended Property

Include user's custom properties defined in the CMS. The default value is "True".

**C#**

```
[Bindable(true), Category("Ektron"), Description("Include user's custom properties defined in
the CMS. The default value is \"True\"."), DefaultValue(true)]
public bool ShowExtended;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Include user's custom properties defined in
the CMS. The default value is \"True\"."), DefaultValue(true)>
Public Property ShowExtended() As bool
```

### Membership.ShowTerms Property

Display the Terms and Conditions text. The default value is "False".

**C#**

```
[Bindable(true), Category("Ektron"), Description("Display the Terms and Conditions text. The
default value is \"False\"."), DefaultValue(false)]
public bool ShowTerms;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("Display the Terms and Conditions text. The
default value is \"False\"."), DefaultValue(false)>
Public Property ShowTerms() As bool
```

### Membership.TaxonomyId Property

Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)]
[Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long TaxonomyId;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)>
<Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property TaxonomyId() As long
```

### Membership.Text Property

Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

**C#**

```
public override string Text;
```

**Visual Basic**

```
Public override Property Text() As string
```

### Membership.UserExistsMessage Property

User (see page 913) already exists error message.

**C#**

```
[Bindable(true), Category("Ektron"), Description("User already exists error message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_EXISTS)]
public string UserExistsMessage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("User already exists error message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_EXISTS)>
Public Property UserExistsMessage() As string
```

### Membership.UserID Property

Specify the user ID for which to retrieve information.

**C#**

```
[Bindable(false), Browsable(false), Description("Specify the user ID for which to retrieve
information."), DefaultValue("0")]
public long UserID;
```

**Visual Basic**

```
<Bindable(false), Browsable(false), Description("Specify the user ID for which to retrieve
information."), DefaultValue("0")>
Public Property UserID() As long
```

### Membership.UserSuccessMessage Property

User (see page 913) successfully registered message.

**C#**

```
[Bindable(true), Category("Ektron"), Description("User successfully registered message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_SUCCEED)]
public string UserSuccessMessage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("User successfully registered message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
```

```
DefaultValue(EkWebControl.RES_MEMBR_SUCCEED)>
Public Property UserSuccessMessage() As string
```

### Membership.UserUpdateSuccessMessage Property

Message explaining the user successfully update his/her information.

**C#**

```
[Bindable(true), Category("Ektron"), Description("User successfully update his/her
information."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_UPDATE)]
public string UserUpdateSuccessMessage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), Description("User successfully update his/her
information."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_UPDATE)>
Public Property UserUpdateSuccessMessage() As string
```

## Menu Class

The Menu server control calls a menu to be displayed on a page. Using the Menu server control, you can manipulate a menu by using the DisplayXslt (see page 360) property.

**Class Hierarchy**

**C#**

```
[DefaultProperty("DefaultMenuID"), ToolboxData("<{0}:Menu runat=server></{0}:Menu>"),
PersistChildren(true)]
public class Menu : EkXsltBase;
```

**Visual Basic**

```
<DefaultProperty("DefaultMenuID"), ToolboxData("<{0}:Menu runat=server></{0}:Menu>"),
PersistChildren(true)>
Public Class Menu
Inherits EkXsltBase
```

**File**

Menu.cs

### Menu Methods

The methods of the Menu class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 358) | Calls up the database and updates the contents of the Menu (see page 358) if the data is out of sync with the specified parameters. |
| | Wrap (see page 359) | Method that can overridden and be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### Menu.Fill Method

Calls up the database and updates the contents of the Menu (see page 358) 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 359) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultMenuID (see page 360) | Gets or Sets the Default MenuID for this Control. |
| | DisplayXslt (see page 360) | The Xslt to use to render the Menu (see page 358) (optional) |
| | DynamicParameter (see page 360) | The parameter to query for a Dynamic Menu (see page 358). |
| | Stylesheet (see page 361) | Specify the style sheet relative to site root. Leave blank for default. |
| | SuppressAddEdit (see page 361) | Setting this property to True caused the menu to hide the Add and Edit buttons for the menu. It is defaulted to false. |

### Menu.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
```

```
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

### Menu.DefaultMenuID Property

Gets or Sets the Default MenuID for this Control.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Editor(typeof(MenuUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("Gets or Sets the Default MenuID for this Control.")]
public long DefaultMenuID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Editor(typeof(MenuUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("Gets or Sets the Default MenuID for this Control.")>
Public Property DefaultMenuID() As long
```

### Menu.DisplayXslt Property

The Xslt to use to render the Menu (see page 358) (optional)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Menu")]
[TypeConverter(typeof(XsltMenuTypeConverter))]
[DefaultValue("SampleMenu")]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Menu")>
<TypeConverter(typeof(XsltMenuTypeConverter))>
<DefaultValue("SampleMenu")>
Public override Property DisplayXslt() As string
```

### Menu.DynamicParameter Property

The parameter to query for a Dynamic Menu (see page 358).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("The parameter to query for a Dynamic Menu.")]
[TypeConverter(typeof(DynamicMenuTypeConverter))]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<Description("The parameter to query for a Dynamic Menu.")>
<TypeConverter(typeof(DynamicMenuTypeConverter))>
Public Property DynamicParameter() As string
```

### Menu.Stylesheet Property

Specify the style sheet relative to site root. Leave blank for default.

**C#**

```
[Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
public virtual string Stylesheet;
```

**Visual Basic**

```
<Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
Public virtual Property Stylesheet() As string
```

### Menu.SuppressAddEdit Property

Setting this property to True caused the menu to hide the Add and Edit buttons for the menu. It is defaulted to false.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Setting this property to true causes the menu to hide the Add and Edit buttons
for the menu. It is defaulted to false.")]
[DefaultValue(false)]
public virtual bool SuppressAddEdit;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Setting this property to true causes the menu to hide the Add and Edit buttons
for the menu. It is defaulted to false.")>
<DefaultValue(false)>
Public virtual Property SuppressAddEdit() As bool
```

## MessageBoard Class

Summary description for Message board

**Class Hierarchy**

**C#**

```
[DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:MessageBoard
runat=server></{0}:MessageBoard>")]
public class MessageBoard : EkXsltWebPart, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:MessageBoard
runat=server></{0}:MessageBoard>")>
Public Class MessageBoard
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

**File**

MessageBoard.cs

## MessageBoard Methods

The methods of the MessageBoard class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 362) | Populates the data for the MessageBoard (see page 361). |
| | RaiseCallbackEvent (see page 362) | |
| | ToString (see page 362) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 362) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## MessageBoard.Fill Method

Populates the data for the MessageBoard (see page 361).

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## MessageBoard.RaiseCallbackEvent Method

**C#**

```
public void RaiseCallbackEvent(string eventArgs);
```

**Visual Basic**

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| eventArgs | string |

## MessageBoard.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

## MessageBoard.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

**Returns**

## MessageBoard Properties

The properties of the MessageBoard class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 363) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultObjectID (see page 364) | The default object ID for this control to use when there is no matching dynamic parameter value passed. |
| | DisplayXslt (see page 364) | Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. |
| | DynamicObjectParameter (see page 364) | Gets or sets the Querystring parameter to read a object ID dynamically. To use the default object ID, leave blank. |
| | EnablePaging (see page 365) | Enables paging if the total records exceeds MaxResult value. This property, in conjunction with the MaxResults (see page 366) property, lets site visitors view an unlimited number of document items while controlling the amount of screen space. To accomplish this, the display is limited to the number set in the MaxResults (see page 366) property.If you set this property to true, and the number of items exceeds the MaxResults (see page 366) number, navigation aids appear below the last item. |
| | MarkupLanguage (see page 365) | Identify the template markup file that controls the display of the server control. For example, mymarkup.ekml. If the *.ekml file is located in the same folder as the Web form containing the server control, just enter its name. Otherwise, the path to the file can be relative or absolute. |
| | MaxResults (see page 366) | Maximum number of results to be returned. |
| | Moderate (see page 366) | This property indicates if the messages are to be moderated by the administrator of the site. |
| | ObjectType (see page 366) | The type of object to which the message board is assigned. Choices are:<br>• **Content (see page 788)** - the message board is assigned to a content item.<br>• **User (see page 913)** - the message board is assigned to an individual.<br>• **CommunityGroup (see page 719)** - the message board is assigned to a community group. |
| | ProfileParamName (see page 367) | The name of the parameter to pass to the profile template. |
| | ShowMaxCharacter (see page 367) | The maximum number of characters to display in the message. If the number of characters in the message surpasses the number set in this property, a "more..." link appears. Clicking this link displays the full text of the message. |
| | Stylesheet (see page 367) | Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | TemplateUserProfile (see page 368) | The URL to launch a new window with, when a member clicks the display-name shown in the messages' browse users screen (used to get more information on a potential message recipient). Optional. |
| | XmlDoc (see page 368) | Returns the Xslt object for the collection. |

## MessageBoard.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
```

```
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

### MessageBoard.DefaultObjectID Property

The default object ID for this control to use when there is no matching dynamic parameter value passed.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The default object ID for this control to use when there is no matching dynamic
parameter value passed.")]
public long DefaultObjectID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The default object ID for this control to use when there is no matching dynamic
parameter value passed.")>
Public Property DefaultObjectID() As long
```

### MessageBoard.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltMessageBoardDataTypeConverter))]
[DefaultValue("Builtin")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltMessageBoardDataTypeConverter))>
<DefaultValue("Builtin")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### MessageBoard.DynamicObjectParameter Property

Gets or sets the Querystring parameter to read a object ID dynamically. To use the default object ID, leave blank.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
```

```
[Description("Gets or sets the Querystring parameter to read a object ID dynamically. To use
the default object ID, leave blank.")]
[Personalizable()]
public string DynamicObjectParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Gets or sets the Querystring parameter to read a object ID dynamically. To use
the default object ID, leave blank.")>
<Personalizable()>
Public Property DynamicObjectParameter() As string
```

### MessageBoard.EnablePaging Property

Enables paging if the total records exceeds MaxResult value. This property, in conjunction with the MaxResults (see page 366) property, lets site visitors view an unlimited number of document items while controlling the amount of screen space. To accomplish this, the display is limited to the number set in the MaxResults (see page 366) property.If you set this property to true, and the number of items exceeds the MaxResults (see page 366) number, navigation aids appear below the last item.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Enables paging if total records exceeds MaxResult value. This property, in
conjunction with the MaxResults property, lets site visitors view an unlimited number of
document items while controlling the amount of screen space. To accomplish this, the display
is limited to the number set in the MaxResults property. If you set this property to true, and
the number of items exceeds the MaxResults number, navigation aids appear below the last
item.")]
[Personalizable()]
public bool EnablePaging;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Enables paging if total records exceeds MaxResult value. This property, in
conjunction with the MaxResults property, lets site visitors view an unlimited number of
document items while controlling the amount of screen space. To accomplish this, the display
is limited to the number set in the MaxResults property. If you set this property to true, and
the number of items exceeds the MaxResults number, navigation aids appear below the last
item.")>
<Personalizable()>
Public Property EnablePaging() As bool
```

### MessageBoard.MarkupLanguage Property

Identify the template markup file that controls the display of the server control. For example, mymarkup.ekml. If the *.ekml file is located in the same folder as the Web form containing the server control, just enter its name. Otherwise, the path to the file can be relative or absolute.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter the name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter the name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

### MessageBoard.MaxResults Property

Maximum number of results to be returned.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
[Personalizable()]
public int MaxResults;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable()>
Public Property MaxResults() As Integer
```

### MessageBoard.Moderate Property

This property indicates if the messages are to be moderated by the administrator of the site.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("This property indicates if the messages are to be moderated by the administrator
of the site.")]
[DefaultValue(false)]
public bool Moderate;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("This property indicates if the messages are to be moderated by the administrator
of the site.")>
<DefaultValue(false)>
Public Property Moderate() As bool
```

### MessageBoard.ObjectType Property

The type of object to which the message board is assigned. Choices are:

- **Content (see page 788)** - the message board is assigned to a content item.
- **User (see page 913)** - the message board is assigned to an individual.
- **CommunityGroup (see page 719)** - the message board is assigned to a community group.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select the type of object with which this control is associated. Choices are:
Content, User and CommunityGroup.")]
[DefaultValue(EkEnumeration.MessageBoardObjectType.Content)]
public EkEnumeration.MessageBoardObjectType ObjectType;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select the type of object with which this control is associated. Choices are:
Content, User and CommunityGroup.")>
<DefaultValue(EkEnumeration.MessageBoardObjectType.Content)>
```

```
Public Property ObjectType() As EkEnumeration.MessageBoardObjectType
```

### MessageBoard.ProfileParamName Property

The name of the parameter to pass to the profile template.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The parameter name to pass on the querystring to the profile and location pages,
which will be set to the ID of the user that was clicked on.")]
[DefaultValue("id")]
public string ProfileParamName;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The parameter name to pass on the querystring to the profile and location pages,
which will be set to the ID of the user that was clicked on.")>
<DefaultValue("id")>
Public Property ProfileParamName() As string
```

### MessageBoard.ShowMaxCharacter Property

The maximum number of characters to display in the message. If the number of characters in the message surpasses the number set in this property, a “more...” link appears. Clicking this link displays the full text of the message.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of characters to display in the message. If the number of
characters in the message surpasses the number set in this property, a ?more...? link appears.
Clicking this link displays the full text of the message.")]
[Personalizable()]
public int ShowMaxCharacter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The maximum number of characters to display in the message. If the number of
characters in the message surpasses the number set in this property, a ?more...? link appears.
Clicking this link displays the full text of the message.")>
<Personalizable()>
Public Property ShowMaxCharacter() As Integer
```

### MessageBoard.Stylesheet Property

Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file. The location can be relative or absolute.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control. Select either 'Use Default', 'None', or enter a
path to a custom Css file. The location can be relative or absolute.")>
<Personalizable()>
Public Property Stylesheet() As string
```

### MessageBoard.TemplateUserProfile Property

The URL to launch a new window with, when a member clicks the display-name shown in the messages' browse users screen (used to get more information on a potential message recipient). Optional.

**C#**

```
[Category("Ektron")]
[Description("The URL to launch a new window with, when a member clicks the display-name shown
in the messages' browse users screen (used to get more information on a potential message
recipient). Optional.")]
[TypeConverter(typeof(string))]
[DefaultValue("")]
public string TemplateUserProfile;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("The URL to launch a new window with, when a member clicks the display-name shown
in the messages' browse users screen (used to get more information on a potential message
recipient). Optional.")>
<TypeConverter(typeof(string))>
<DefaultValue("")>
Public Property TemplateUserProfile() As string
```

### MessageBoard.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## Messaging Class

**Class Hierarchy**

**C#**

```
[ToolboxData("<{0}:Messaging runat=server></{0}:Messaging>")]
public class Messaging : EkXsltWebPart;
```

**Visual Basic**

```
<ToolboxData("<{0}:Messaging runat=server></{0}:Messaging>")>
Public Class Messaging
Inherits EkXsltWebPart
```

**File**

Messaging.cs

## Messaging Methods

The methods of the Messaging class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 369) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 369) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 369) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## Messaging.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## Messaging.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

## Messaging.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

### Returns

## Messaging Properties

The properties of the Messaging class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DisplayXslt (see page 370) | Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. (optional) |
| | EnablePreSearch (see page 370) | Determines if the message recipient user search should render with a completed search for all when opened. |

| | | |
|---|---|---|
| | FriendsOnly (see page 371) | When this property is set to True, users can only send messages to people on their colleagues list. Default is True.<br>• **True** - send messages to colleagues only.<br>• **False** - send messages to all CMS400.NET users. |
| | GroupParamName (see page 371) | Enter the QueryString parameter used to pass the ID of a group. For example, if the QueryString is: `?gid=1`, then enter "gid" in this property. You can only pass one group. This property is typically used when you want to pre-populate the **To:** field of a message. |
| | JavascriptEditorHeight (see page 371) | Set the height in pixels for the editor. The default is 400. |
| | JavascriptEditorHTMLMode (see page 372) | Set to True to allow users to edit their posts and replies in HTML. The default is False. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Allowing HTML Editing in the Javascript Editor" |
| | JavascriptEditorToolbar (see page 372) | Decide which buttons are available in the Javascript Editor. For additional information on the Javascript Toolbar Editor, See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Adding and Removing Toolbar Buttons From the Javascript Editor" |
| | JavascriptEditorWidth (see page 373) | Set the width in pixels for the editor. The default is 625. |
| | ProfileParamName (see page 373) | This property identifies the QueryString parameter used to pass the ID of a user to a profile or location template when a user name is clicked in the Browse User (see page 913) screen. The default is ID. The parameter defined in this property is appended to the QueryString of the path defined in the ProfileUrl (see page 373) property. The default is ID. **Important:** This parameter needs to match the parameter set in the UserProfile (see page 499) server control's DynamicParameter property on the page defined in the ProfileUrl (see page 373) property. |
| | ProfileUrl (see page 373) | The URL of the profile page template. This page is opened in a new window when a user clicks a display name on the Browse Users screen. This allows a user to get additional profile information about the potential message recipient. (Optional) |
| | RecipientParamName (see page 374) | Enter the QueryString parameter used to pass the ID of a user. For example, if the QueryString is: `?uid=1`, then enter "uid" in this property. You can pass multiple users by repeating this. For example: `?uid=1&uid=20&uid=12&uid=18` This property is typically used when you want to pre-populate the **To:** field of a message. |
| | ShowSearchOptions (see page 374) | This property is obsolete and no longer has any affect. |
| | Stylesheet (see page 374) | Specify the path to a style sheet for use with the server control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | UserSearchDisplayXslt (see page 375) | The XSLT used to display the user search control inside the Messaging (see page 368) server control. |
| | XmlDoc (see page 375) | Returns the Xslt object for the collection. |

## Messaging.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute. (optional)

**C#**

```
[Category("Ektron")]
[Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. (optional)")]
[TypeConverter(typeof(XsltMessagingTypeConverter))]
[DefaultValue("Basic Messaging")]
public override string DisplayXslt;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. (optional)")>
<TypeConverter(typeof(XsltMessagingTypeConverter))>
<DefaultValue("Basic Messaging")>
Public override Property DisplayXslt() As string
```

## Messaging.EnablePreSearch Property

Determines if the message recipient user search should render with a completed search for all when opened.

**C#**

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set this to true to have all users (or colleagues) listed when the search is
opened, otherwise setting false will prevent this.")]
public bool EnablePreSearch;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set this to true to have all users (or colleagues) listed when the search is
opened, otherwise setting false will prevent this.")>
Public Property EnablePreSearch() As bool
```

### Messaging.FriendsOnly Property

When this property is set to True, users can only send messages to people on their colleagues list. Default is True.

- **True** - send messages to colleagues only.
- **False** - send messages to all CMS400.NET users.

**C#**

```
[Category("Ektron")]
[DefaultValue(true)]
[Description("When set to false, includes all CMS users when True only friends will be
available as message recipients. True is default.")]
public bool FriendsOnly;
```

**Visual Basic**

```
<Category("Ektron")>
<DefaultValue(true)>
<Description("When set to false, includes all CMS users when True only friends will be
available as message recipients. True is default.")>
Public Property FriendsOnly() As bool
```

### Messaging.GroupParamName Property

Enter the QueryString parameter used to pass the ID of a group. For example, if the QueryString is: `?gid=1`, then enter “gid” in this property. You can only pass one group. This property is typically used when you want to pre-populate the **To:** field of a message.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The parameter name to pass on the query string, that identifies the group for
the message (used to pre-populate the to=field)")]
[DefaultValue("gid")]
public string GroupParamName;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The parameter name to pass on the query string, that identifies the group for
the message (used to pre-populate the to=field)")>
<DefaultValue("gid")>
Public Property GroupParamName() As string
```

### Messaging.JavascriptEditorHeight Property

Set the height in pixels for the editor. The default is 400.

**C#**

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue(400)]
[Description("The pixel height for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public int JavascriptEditorHeight;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(400)>
<Description("The pixel height for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorHeight() As Integer
```

### Messaging.JavascriptEditorHTMLMode Property

Set to True to allow users to edit their posts and replies in HTML. The default is False. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Allowing HTML Editing in the Javascript Editor"

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Determines if HTML mode is allowed for the Javascript Editor.")]
[Obsolete("JavaScript Editor has been deprecated. Instead control is using CMS400WebEditor.")]
public bool JavascriptEditorHTMLMode;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Determines if HTML mode is allowed for the Javascript Editor.")>
<Obsolete("JavaScript Editor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorHTMLMode() As bool
```

### Messaging.JavascriptEditorToolbar Property

Decide which buttons are available in the Javascript Editor. For additional information on the Javascript Toolbar Editor, See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Adding and Removing Toolbar Buttons From the Javascript Editor"

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("ParagraphMenu,FontFacesMenu,FontSizesMenu,FontForeColorsMenu|Bold,Italic,Underli
ne,Strikethrough;Superscript,Subscript,RemoveFormat|JustifyLeft,JustifyCenter,JustifyRight,Just
ifyFull;BulletedList,NumberedList,Indent,Outdent;CreateLink,Unlink,InsertRule|Cut,Copy,Paste;Un
do,Redo,Print")]
[Description("The toolbar configuration for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public string JavascriptEditorToolbar;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("ParagraphMenu,FontFacesMenu,FontSizesMenu,FontForeColorsMenu|Bold,Italic,Underli
ne,Strikethrough;Superscript,Subscript,RemoveFormat|JustifyLeft,JustifyCenter,JustifyRight,Just
ifyFull;BulletedList,NumberedList,Indent,Outdent;CreateLink,Unlink,InsertRule|Cut,Copy,Paste;Un
do,Redo,Print")>
<Description("The toolbar configuration for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorToolbar() As string
```

### Messaging.JavascriptEditorWidth Property

Set the width in pixels for the editor. The default is 625.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(625)]
[Description("The pixel width for the Javascript Editor.")]
[Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")]
public int JavascriptEditorWidth;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(625)>
<Description("The pixel width for the Javascript Editor.")>
<Obsolete("JavaScriptEditor has been deprecated. Instead control is using CMS400WebEditor.")>
Public Property JavascriptEditorWidth() As Integer
```

### Messaging.ProfileParamName Property

This property identifies the QueryString parameter used to pass the ID of a user to a profile or location template when a user name is clicked in the Browse User (see page 913) screen. The default is ID. The parameter defined in this property is appended to the QueryString of the path defined in the ProfileUrl (see page 373) property. The default is ID. **Important:** This parameter needs to match the parameter set in the UserProfile (see page 499) server control's DynamicParameter property on the page defined in the ProfileUrl (see page 373) property.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("This property identifies the QueryString parameter used to pass the ID of a user
to a profile page or location template when a user name is clicked in the Browse User screen.
The default is ID. The parameter defined in this property is appended to the QueryString of
the path defined in the ProfileUrl property. The default is ID.")]
[DefaultValue("id")]
public string ProfileParamName;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("This property identifies the QueryString parameter used to pass the ID of a user
to a profile page or location template when a user name is clicked in the Browse User screen.
The default is ID. The parameter defined in this property is appended to the QueryString of
the path defined in the ProfileUrl property. The default is ID.")>
<DefaultValue("id")>
Public Property ProfileParamName() As string
```

### Messaging.ProfileUrl Property

The URL of the profile page template. This page is opened in a new window when a user clicks a display name on the Browse Users screen. This allows a user to get additional profile information about the potential message recipient. (Optional)

**C#**

```
[Category("Ektron")]
[Description("The URL of the profile page template. This page is opened when a user clicks a
display name in the Browse Users screen. This allows a user to get additional profile
information about the potential message recipient. (Optional)")]
[TypeConverter(typeof(string))]
[DefaultValue("")]
public string ProfileUrl;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("The URL of the profile page template. This page is opened when a user clicks a
display name in the Browse Users screen. This allows a user to get additional profile
information about the potential message recipient. (Optional)")>
<TypeConverter(typeof(string))>
<DefaultValue("")>
Public Property ProfileUrl() As string
```

### Messaging.RecipientParamName Property

Enter the QueryString parameter used to pass the ID of a user. For example, if the QueryString is: `?uid=1`, then enter “uid” in this property. You can pass multiple users by repeating this. For example: `?uid=1&uid=20&uid=12&uid=18` This property is typically used when you want to pre-populate the **To:** field of a message.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The parameter name to pass on the querystring, that identifies the recipient of
the message (used to prepopulate the to=field)")]
[DefaultValue("uid")]
public string RecipientParamName;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The parameter name to pass on the querystring, that identifies the recipient of
the message (used to prepopulate the to=field)")>
<DefaultValue("uid")>
Public Property RecipientParamName() As string
```

### Messaging.ShowSearchOptions Property

This property is obsolete and no longer has any affect.

**C#**

```
[Category("Ektron")]
[DefaultValue(false)]
[Description("This property is obsolete and no longer has any affect")]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool ShowSearchOptions;
```

**Visual Basic**

```
<Category("Ektron")>
<DefaultValue(false)>
<Description("This property is obsolete and no longer has any affect")>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property ShowSearchOptions() As bool
```

### Messaging.Stylesheet Property

Specify the path to a style sheet for use with the server control. The location can be relative or absolute. Leave blank to use the default style sheet.

**C#**

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with the server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(string))]
[DefaultValue("Community.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with the server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(string))>
<DefaultValue("Community.css")>
Public Property Stylesheet() As string
```

### Messaging.UserSearchDisplayXslt Property

The XSLT used to display the user search control inside the Messaging (see page 368) server control.

**C#**

```
[Category("Ektron")]
[Description("The XSLT used to display the user search control inside the Messaging server
control.")]
[TypeConverter(typeof(String))]
[DefaultValue("MsgUserSelect.xsl")]
public string UserSearchDisplayXslt;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("The XSLT used to display the user search control inside the Messaging server
control.")>
<TypeConverter(typeof(String))>
<DefaultValue("MsgUserSelect.xsl")>
Public Property UserSearchDisplayXslt() As string
```

### Messaging.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## MetaData Class

The Metadata server control lets you add the metadata from content blocks to a Web page. This lets developers add metadata quickly without having to type it in. You can add metadata from a single content block, multiple content blocks, or by dynamically passing a content ID from a URL.

**Class Hierarchy**

**C#**

```
public class MetaData : Ektron.Cms.Controls.EkWebControl;
```

**Visual Basic**

```
Public Class MetaData
Inherits Ektron.Cms.Controls.EkWebControl
```

**File**

MetaData.cs

## MetaData Methods

The methods of the MetaData class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 376) | Method that can be overridden and used to grab data from the back end for each server control. |

## MetaData.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## MetaData Properties

The properties of the MetaData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 376) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultContentID (see page 377) | The content block ID from which the server control gets the metadata. If you want to add metadata from several content blocks, set this property to 0 (zero) and use the DefaultItemList (see page 377) property to identify the content blocks. |
| | DefaultItemList (see page 377) | A comma delimited list of content block IDs from which to get metadata. DefaultContentID (see page 377) must be set to 0 (zero) so you can add IDs to the list. |
| | DynamicParameter (see page 377) | Gets or sets the QueryString parameter to read a content ID dynamically. Set to "Use Default" if you wish to always display the default ContentID (static.) |
| | GenerateDublinCore (see page 378) | When enabled, this property automatically creates seven of the Simple Dublin Core metadata fields from standard CMS400.NET system properties. The default is false.<br>• **True** = generate Simple Dublin Core metadata fields<br>• **False** = do not generate Simple Dublin Core metadata fields<br>The seven fields and how they are associated with the CMS400.NET properties is explained in the CMS400.NET Developer Manual section "MetaData (see page 375) Server Control" > "Using the Simple Dublin Core Metadata (see page 683) Standard" |

## MetaData.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
```

```
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

### MetaData.DefaultContentID Property

The content block ID from which the server control gets the metadata. If you want to add metadata from several content blocks, set this property to 0 (zero) and use the DefaultItemList (see page 377) property to identify the content blocks.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control.")]
public long DefaultContentID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control.")>
Public Property DefaultContentID() As long
```

### MetaData.DefaultItemList Property

A comma delimited list of content block IDs from which to get metadata. DefaultContentID (see page 377) must be set to 0 (zero) so you can add IDs to the list.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("A comma delimited list of content block IDs from which to get metadata.
DefaultContentID must be set to 0 (zero) so you can add IDs to the list.")]
public string DefaultItemList;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("A comma delimited list of content block IDs from which to get metadata.
DefaultContentID must be set to 0 (zero) so you can add IDs to the list.")>
Public Property DefaultItemList() As string
```

### MetaData.DynamicParameter Property

Gets or sets the QueryString parameter to read a content ID dynamically. Set to "Use Default" if you wish to always display the default ContentID (static.)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")]
```

```
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the QueryString parameter to read a Content ID dynamically - Set to
\"Use Default\" if you wish to always display the default ContentID (static.)")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

### MetaData.GenerateDublinCore Property

When enabled, this property automatically creates seven of the Simple Dublin Core metadata fields from standard CMS400.NET system properties. The default is false.

- **True** = generate Simple Dublin Core metadata fields
- **False** = do not generate Simple Dublin Core metadata fields

The seven fields and how they are associated with the CMS400.NET properties is explained in the CMS400.NET Developer Manual section "MetaData (see page 375) Server Control" > "Using the Simple Dublin Core Metadata (see page 683) Standard"

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Append Dublin Core Metadata")]
public bool GenerateDublinCore;
```

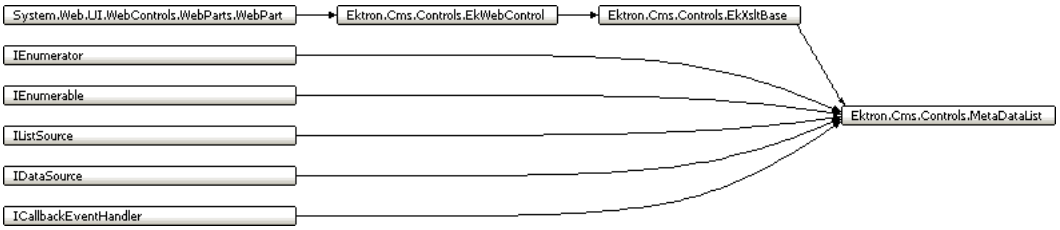
**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Append Dublin Core Metadata")>
Public Property GenerateDublinCore() As bool
```

## MetaDataList Class

Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.

**Class Hierarchy**




**C#**

```
[DefaultProperty("KeyWordName"), PersistChildren(true)]
public class MetaDataList : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("KeyWordName"), PersistChildren(true)>
Public Class MetaDataList
```

```
Inherits EkXsltBase
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

**File**

MetaDataList.cs

## MetaDataList Methods

The methods of the MetaDataList class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | CopyTo (see page 379) | 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 379) | 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 379) | Gets a System.Collections.IEnumerator for the collection. |
| | MoveNext (see page 380) | Moves up the index of the Collection (see page 88). Returns false if the end of the collection has been surpassed. |
| | Reset (see page 380) | Brings the index of the collection back to zero. Does not clear the collection. |
| | this (see page 380) | Index operator into a strongly typed dataset holding some common properties for a Collection (see page 88) item. |
| | Wrap (see page 380) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## MetaDataList.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

**C#**

```
public void CopyTo(Array array, int i);
```

**Visual Basic**

```
Public Function CopyTo(array As Array, i As Integer) As void
```

## MetaDataList.Fill Method

Calls up the database and updates the items in the control if the data is out of sync with the specified parameters.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## MetaDataList.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

**C#**

```
public IEnumerator GetEnumerator();
```

**Visual Basic**

```
Public Function GetEnumerator() As IEnumerator
```

### MetaDataList.MoveNext Method

Moves up the index of the Collection (see page 88). 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 88) 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 382) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | ContentType (see page 383) | Select a type of content for this control. Choices are:<br>• all types<br>• archived content<br>• archived forms<br>• content<br>• forms |
| | Count (see page 383) | Indicates the number of elements in your Collection (see page 88). |
| | Current (see page 383) | Gets the current object in the collection. |
| | EkItems (see page 383) | Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 88). Holds more information than the exposed strongly typed this[i] index operator. |
| | EnablePaging (see page 384) | This property, in conjunction with the `MaxNumber (see page 388)` property, lets site visitors view an unlimited number of content items while controlling the amount of screen space. To accomplish this, the content display is limited to the number set in the `MaxNumber (see page 388)` property. If you set this property to **True**, and the number of content items exceeds the `MaxNumber (see page 388)` number, navigation aids appear below the last item. |
| | ExactPhrase (see page 384) | Gets or Sets the boolean value of ExactPhrase. Determines whether the KeyWordValue (see page 386) needs to match the phrase exactly.<br>For example, if "site" is the KeyWordValue (see page 386) and the title of a content block is "Welcome (see page 1) to the site" and ExactPhrase is set to **True**, you would not see the content block in the metadata list. This is because "site" does not equal "Welcome (see page 1) to the site".<br>• **True** = match the exact phrase<br>• **False** = does not need to match exact phrase |
| | FolderID (see page 385) | The folder ID from which content is retrieved. At the `Recursive (see page 389)` property, you determine if content in this folder's subfolders is also retrieved. |
| | GetHtml (see page 385) | If set to true, retrieves the content (Html body) of every content block in the MetadataList for display. |
| | IncludeIcons (see page 385) | Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with "ecmSummary" and "ecmTeaser" or when the [$imageIcon] variable is used in an EkML file. |
| | IsSynchronized (see page 386) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | KeyWordName (see page 386) | KeyWordName represents a metadata definition, that is, the container for the KeyWordValues. Examples of a KeyWordName are **Keywords** and **Title**. |
| | KeyWordValue (see page 386) | Enter the values associated with the `KeyWordName (see page 386)`. Only content whose metadata (defined at the KeyWordName (see page 386) property) matches this value appears on the metadata list. |
| | KeyWordValueMatchAll (see page 387) | This property is only used if you enter more than one keyword value.<br>If you do, and only want content to appear on the metadata list if all values entered at the KeyWordValue (see page 386) field match its metadata values, enter true.<br>If metadata can appear on the list as long as *any value* defined at the KeyWordValue (see page 386) field matches the selected metadata value for a content item, enter **false**.<br>**Example:**<br>KeyWordValue (see page 386) for **Title** (assigned for this server control): home; page; company.<br>Metadata (see page 683) values for a content item's Title metadata definition field: software; ektron; company.<br>If `KeyWordValueMatchAll = true`, content... more (see page 387) |
| | KeyWordValueSeparator (see page 387) | Enter the character used to separate the list of keyword values. An example is a semicolon(**;**). |

| | | |
|---|---|---|
| | LinkTarget (see page 387) | Gets or Sets the target window for the result. Choices are:<br>• **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named.<br>• **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target.<br>• **_parent** - this target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.<br>• **_top** - this target makes the link load in the full... more (see page 387) |
| | MarkupLanguage (see page 388) | Enter name of a markup file that manages the display of this server control. |
| | MaxNumber (see page 388) | Determine the maximum number of content items to return. If set to zero then all items are returned. |
| | OrderBy (see page 388) | The order of the list to be returned. Order by:<br>• **Title** - The title of the content block<br>• **ID** - The content block ID number<br>• **Date Created** - The date the content block was created<br>• **Date Modified** - The date the content block was last modified<br>• **LastEditorLname** - The last editor's last name<br>• **LastEditorFname** - The last editor's first name |
| | Recursive (see page 389) | Whether to search sub-folders of the identified root folder. The starting folder is identified in the `FolderID` (see page 385) property. |
| | SortOrder (see page 389) | Gets or sets the sort order. Choose whether to sort by Ascending or Descending. |
| | SuppressWrapperTags (see page 390) | Suppresses the output of span/div tags around the control. |
| | SyncRoot (see page 390) | Gets an object that can be used to synchronize the collection. |
| | XmlDoc (see page 390) | Returns the XSLT object for the MetaDataList (see page 378) control. |

### MetaDataList.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable(), WebBrowsable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable(), WebBrowsable()>
Public virtual Property CacheInterval() As double
```

### MetaDataList.ContentType Property

Select a type of content for this control. Choices are:

- all types
- archived content
- archived forms
- content
- forms

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select a type of content for this control. Choices are: all types, archived
content, archived forms, content or forms.")]
[Personalizable(), WebBrowsable()]
public Ektron.Cms.Controls.CmsWebService.CMSContentType ContentType;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select a type of content for this control. Choices are: all types, archived
content, archived forms, content or forms.")>
<Personalizable(), WebBrowsable()>
Public Property ContentType() As Ektron.Cms.Controls.CmsWebService.CMSContentType
```

### MetaDataList.Count Property

Indicates the number of elements in your Collection (see page 88).

**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 88). Holds more information than the exposed strongly typed this[i] index operator.

**C#**

```
[Browsable(false)]
```

```
public Ektron.Cms.Common.ContentBase EkItems;
```

**Visual Basic**

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

### MetaDataList.EnablePaging Property

This property, in conjunction with the MaxNumber (see page 388) property, lets site visitors view an unlimited number of content items while controlling the amount of screen space. To accomplish this, the content display is limited to the number set in the MaxNumber (see page 388) property. If you set this property to **True**, and the number of content items exceeds the MaxNumber (see page 388) number, navigation aids appear below the last item.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enables paging if total records exceeds MaxResult value. Use this property with
a non zero value in the MaxNumbers properties.")]
public bool EnablePaging;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enables paging if total records exceeds MaxResult value. Use this property with
a non zero value in the MaxNumbers properties.")>
Public Property EnablePaging() As bool
```

### MetaDataList.ExactPhrase Property

Gets or Sets the boolean value of ExactPhrase. Determines whether the KeyWordValue (see page 386) needs to match the phrase exactly.

For example, if “site” is the KeyWordValue (see page 386) and the title of a content block is “Welcome (see page 1) to the site” and ExactPhrase is set to **True**, you would not see the content block in the metadata list. This is because “site” does not equal “Welcome (see page 1) to the site”.

- **True** = match the exact phrase
- **False** = does not need to match exact phrase

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("True")]
[Description("Gets or Sets the boolean value of ExactPhrase. Determines whether the
KeyWordValue needs to match the phrase exactly. For example, if \"site\" is the KeyWordValue
and the title of a content block is \"Welcome to the site\" and ExactPhrase is set to
\"True\", you would not see the content block in the metadata list. This is because \"site\"
does not equal \"Welcome to the site\".")]
public bool ExactPhrase;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("True")>
<Description("Gets or Sets the boolean value of ExactPhrase. Determines whether the
KeyWordValue needs to match the phrase exactly. For example, if \"site\" is the KeyWordValue
and the title of a content block is \"Welcome to the site\" and ExactPhrase is set to
\"True\", you would not see the content block in the metadata list. This is because \"site\"
does not equal \"Welcome to the site\".")>
Public Property ExactPhrase() As bool
```

### MetaDataList.FolderID Property

The folder ID from which content is retrieved. At the `Recursive` (see page 389) property, you determine if content in this folder's subfolders is also retrieved.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("The folder ID from which content is retrieved. At the \"Recursive property\",
you determine if content in this folder?s subfolders is also retrieved.")]
[Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long FolderID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<Description("The folder ID from which content is retrieved. At the \"Recursive property\",
you determine if content in this folder?s subfolders is also retrieved.")>
<Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property FolderID() As long
```

### MetaDataList.GetHtml Property

If set to true, retrieves the content (Html body) of every content block in the MetadataList for display.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, retrieves the content (Html body) of every content block in the
MetadataList for display.")]
public bool GetHtml;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, retrieves the content (Html body) of every content block in the
MetadataList for display.")>
Public Property GetHtml() As bool
```

### MetaDataList.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with "ecmSummary" and "ecmTeaser" or when the [$imageIcon] variable is used in an EkML file.

**C#**

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
public bool IncludeIcons;
```

**Visual Basic**

```
<Browsable(true)>
<Bindable(true)>
```

```
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
Public Property IncludeIcons() As bool
```

### MetaDataList.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

```
[Browsable(false)]
public bool IsSynchronized;
```

**Visual Basic**

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

### MetaDataList.KeyWordName Property

KeyWordName represents a metadata definition, that is, the container for the KeyWordValues. Examples of a KeyWordName are **Keywords** and **Title**.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(" -None -")]
[Description("KeyWordName represents a metadata definition, that is, the container for the
KeyWordValues. Examples of a KeyWordName are \"Keywords\" and \"Title\".")]
[Editor(typeof(MetaDataUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public string KeyWordName;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(" -None -")>
<Description("KeyWordName represents a metadata definition, that is, the container for the
KeyWordValues. Examples of a KeyWordName are \"Keywords\" and \"Title\".")>
<Editor(typeof(MetaDataUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property KeyWordName() As string
```

### MetaDataList.KeyWordValue Property

Enter the values associated with the KeyWordName (see page 386). Only content whose metadata (defined at the KeyWordName (see page 386) property) matches this value appears on the metadata list.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("None - Use Default")]
[Description("Enter the values associated with the \"KeyWordName\". Only content whose
metadata (defined at the KeyWordName property) matches this value appears on the metadata
list.")]
[Personalizable(), WebBrowsable()]
public string KeyWordValue;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("None - Use Default")>
<Description("Enter the values associated with the \"KeyWordName\". Only content whose
```

```
metadata (defined at the KeyWordName property) matches this value appears on the metadata
list.")>
<Personalizable(), WebBrowsable()>
Public Property KeyWordValue() As string
```

### MetaDataList.KeyWordValueMatchAll Property

This property is only used if you enter more than one keyword value.

If you do, and only want content to appear on the metadata list if all values entered at the KeyWordValue (see page 386) field match its metadata values, enter true.

If metadata can appear on the list as long as *any value* defined at the KeyWordValue (see page 386) field matches the selected metadata value for a content item, enter **false**.

**Example:**

KeyWordValue (see page 386) for **Title** (assigned for this server control): home; page; company.

Metadata (see page 683) values for a content item's Title metadata definition field: software; ektron; company.

If `KeyWordValueMatchAll = true`, content does not appear on metadata list because some items do not match.

If `KeyWordValueMatchAll = false`, content item appears on metadata list because one item (company) matches.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Match all values specified in KeyWordValue separator by KeyWordValueSeparator.")]
public bool KeyWordValueMatchAll;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Match all values specified in KeyWordValue separator by KeyWordValueSeparator.")>
Public Property KeyWordValueMatchAll() As bool
```

### MetaDataList.KeyWordValueSeparator Property

Enter the character used to separate the list of keyword values. An example is a semicolon(**;**).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(";")]
[Description("Enter the character used to separate the list of keyword values. An example is a
semicolon(;).")]
[Personalizable(), WebBrowsable()]
public string KeyWordValueSeparator;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(";")>
<Description("Enter the character used to separate the list of keyword values. An example is a
semicolon(;).")>
<Personalizable(), WebBrowsable()>
Public Property KeyWordValueSeparator() As string
```

### MetaDataList.LinkTarget Property

Gets or Sets the target window for the result. Choices are:

- **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named.
- **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target.
- **_parent** - this target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- **_top** - this target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

### MetaDataList.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

### MetaDataList.MaxNumber Property

Determine the maximum number of content items to return. If set to zero then all items are returned.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Determine the maximum number of content items to return. If set to zero then all
items are returned.")]
public int MaxNumber;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Determine the maximum number of content items to return. If set to zero then all
items are returned.")>
Public Property MaxNumber() As Integer
```

### MetaDataList.OrderBy Property

The order of the list to be returned. Order by:

- **Title** - The title of the content block
- **ID** - The content block ID number

- **Date Created** - The date the content block was created
- **Date Modified** - The date the content block was last modified
- **LastEditorLname** - The last editor's last name
- **LastEditorFname** - The last editor's first name

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("This value determines how to order the return list. Choices are: Title, ID. Date
Created, Date Modified, LastEditorLname or LastEditorFname.")]
[Personalizable(), WebBrowsable()]
public Ektron.Cms.Controls.CmsWebService.ContentOrderBy OrderBy;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("This value determines how to order the return list. Choices are: Title, ID. Date
Created, Date Modified, LastEditorLname or LastEditorFname.")>
<Personalizable(), WebBrowsable()>
Public Property OrderBy() As Ektron.Cms.Controls.CmsWebService.ContentOrderBy
```

### MetaDataList.Recursive Property

Whether to search sub-folders of the identified root folder. The starting folder is identified in the `FolderID` (see page 385) property.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Whether to search sub-folders of the identified root folder. The starting folder
is identified in the \"FolderID\" property.")]
public bool Recursive;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Whether to search sub-folders of the identified root folder. The starting folder
is identified in the \"FolderID\" property.")>
Public Property Recursive() As bool
```

### MetaDataList.SortOrder Property

Gets or sets the sort order. Choose whether to sort by Ascending or Descending.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("This value determines how to order the return list, Ascending or Descending.")]
[Personalizable(), WebBrowsable()]
public Ektron.Cms.Controls.CmsWebService.OrderByDirection SortOrder;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("This value determines how to order the return list, Ascending or Descending.")>
<Personalizable(), WebBrowsable()>
Public Property SortOrder() As Ektron.Cms.Controls.CmsWebService.OrderByDirection
```

### MetaDataList.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

**C#**

```
public override bool SuppressWrapperTags;
```

**Visual Basic**

```
Public override Property SuppressWrapperTags() As bool
```

### MetaDataList.SyncRoot Property

Gets an object that can be used to synchronize the collection.

**C#**

```
[Browsable(false)]
public object SyncRoot;
```

**Visual Basic**

```
<Browsable(false)>
Public Property SyncRoot() As object
```

### MetaDataList.XmlDoc Property

Returns the XSLT object for the MetaDataList (see page 378) control.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## MyAccount Class

**Class Hierarchy**

**C#**

```
[DefaultProperty("DefaultMyAccountID"), ToolboxData("<{0}:MyAccount
runat=server></{0}:MyAccount>")]
public class MyAccount : ECommerceControl, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultMyAccountID"), ToolboxData("<{0}:MyAccount
runat=server></{0}:MyAccount>")>
Public Class MyAccount
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

**File**

MyAccount.cs

**Description**

This is class Ektron.Cms.Controls.MyAccount.

## MyAccount Methods

The methods of the MyAccount class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 391) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 391) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 391) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## MyAccount.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## MyAccount.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

## MyAccount.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

### Returns

## MyAccount Properties

The properties of the MyAccount class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | CustomPropertyID (see page 392) | An ID or comma separated list of IDs for a Custom User (see page 913) Property (or properties) to expose in the UI. These properties are used to define information about a user beyond the standard Ektron CMS400.NET user properties; such as Username, First Name, Password and email Address. |

| | | |
|---|---|---|
| | DefaultCountryId (see page 392) | The ID of the default country that appears in the Billing Address and Shipping Address. Set this ID to the country from which the majority of your customers will be making purchases. If the site visitor is from a different country, he can it change when editing the Billing or Shipping Address. **Note:** If a site visitor enters information in the Billing Address area, his Billing Country seeds the Shipping Country when adding a new Shipping Address. |
| | DisplayXslt (see page 393) | Specify an external XSLT file to render the control. By default, the control uses MyAccount.xsl. This file is located in |
| | EkItem (see page 393) | Provides access to the CustomerData (see page 1255) object. |
| | LoadingImage (see page 393) | Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**. |
| | RedirectUrl (see page 394) | Enter the path of a template that contains the Login (see page 337) server control. When a site visitor arrives at this control and is not logged in, the control checks this property for a path to template containing a Login (see page 337) server control. If one exists, the site visitor is sent to that template. Once he's logged in, he's returned to the template containing the MyAccount (see page 390) server control. If no path exists, the control checks the following key the web.config file for a path. |
| | Stylesheet (see page 394) | Specify the path to a style sheet for use with this server control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | UseMsAjax (see page 394) | When true, uses Microsoft Ajax. |

## MyAccount.CustomPropertyID Property

An ID or comma separated list of IDs for a Custom User (see page 913) Property (or properties) to expose in the UI. These properties are used to define information about a user beyond the standard Ektron CMS400.NET user properties; such as Username, First Name, Password and email Address.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("An ID or comma separated list of IDs for a Custom User Property (or properties)
to expose in the UI. These properties are used to define information about a user beyond the
standard Ektron CMS400.NET user properties; such as Username, First Name, Password and email
Address.")]
[DefaultValue("")]
public string CustomPropertyID;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("An ID or comma separated list of IDs for a Custom User Property (or properties)
to expose in the UI. These properties are used to define information about a user beyond the
standard Ektron CMS400.NET user properties; such as Username, First Name, Password and email
Address.")>
<DefaultValue("")>
Public Property CustomPropertyID() As string
```

## MyAccount.DefaultCountryId Property

The ID of the default country that appears in the Billing Address and Shipping Address. Set this ID to the country from which the majority of your customers will be making purchases. If the site visitor is from a different country, he can it change when editing the Billing or Shipping Address. **Note:** If a site visitor enters information in the Billing Address area, his Billing Country seeds the Shipping Country when adding a new Shipping Address.

**C#**

```
[Category("Ektron")]
[Description("The numeric ID of the country to use by default, which is then used to obtain
regions/states when country is unknown (for USA, DefaultCountryId=438).")]
public int DefaultCountryId;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("The numeric ID of the country to use by default, which is then used to obtain
regions/states when country is unknown (for USA, DefaultCountryId=438).")>
Public Property DefaultCountryId() As Integer
```

### MyAccount.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses MyAccount.xsl. This file is located in

```
[Site Root]\Workarea\Xslt\Commerce
```

**Important:** If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses MyAccount.xsl. This file is located in [Site Root]\\Workarea\\Xslt\\Commerce. Important:
If you want to edit this file, it is strongly recommended that you make a copy, change its
name and move it outside of the Workarea folder. If you make changes to this file and do not
move it out of the Workarea, this file and the changes will be lost when you upgrade.")]
[TypeConverter(typeof(XsltMyAccountTypeConverter))]
[DefaultValue("MyAccount.xsl")]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses MyAccount.xsl. This file is located in [Site Root]\\Workarea\\Xslt\\Commerce. Important:
If you want to edit this file, it is strongly recommended that you make a copy, change its
name and move it outside of the Workarea folder. If you make changes to this file and do not
move it out of the Workarea, this file and the changes will be lost when you upgrade.")>
<TypeConverter(typeof(XsltMyAccountTypeConverter))>
<DefaultValue("MyAccount.xsl")>
Public override Property DisplayXslt() As string
```

### MyAccount.EkItem Property

Provides access to the CustomerData (see page 1255) object.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Browsable(false)]
[Description("Returns the current CustomerData object")]
[Personalizable()]
public Ektron.Cms.Commerce.CustomerData EkItem;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Browsable(false)>
<Description("Returns the current CustomerData object")>
<Personalizable()>
Public Property EkItem() As Ektron.Cms.Commerce.CustomerData
```

### MyAccount.LoadingImage Property

Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
```

```
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

### MyAccount.RedirectUrl Property

Enter the path of a template that contains the Login (see page 337) server control. When a site visitor arrives at this control and is not logged in, the control checks this property for a path to template containing a Login (see page 337) server control. If one exists, the site visitor is sent to that template. Once he's logged in, he's returned to the template containing the MyAccount (see page 390) server control. If no path exists, the control checks the following key the web.config file for a path.

```
<add key=”ek_RedirectToLoginURL” value=”” />
```

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("This value may be set to point to a login page, when an unauthenticated visitor
is detected.")]
[DefaultValue("")]
public string RedirectUrl;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("This value may be set to point to a login page, when an unauthenticated visitor
is detected.")>
<DefaultValue("")>
Public Property RedirectUrl() As string
```

### MyAccount.Stylesheet Property

Specify the path to a style sheet for use with this server control. The location can be relative or absolute. Leave blank to use the default style sheet.

**C#**

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(MyAccountCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/MyAccount.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(MyAccountCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/MyAccount.css")>
Public Property Stylesheet() As string
```

### MyAccount.UseMsAjax Property

When true, uses Microsoft Ajax.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
```

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")]
[DefaultValue("false")]
public bool UseMsAjax;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")>
<DefaultValue("false")>
Public Property UseMsAjax() As bool
```

## OrderList Class

Summary description for OrderList control

### Class Hierarchy

**C#**

```
[DefaultProperty("DefaultUserID"), ToolboxData("<{0}:OrderList runat=server></{0}:OrderList>")]
public class OrderList : ECommerceControl, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultUserID"), ToolboxData("<{0}:OrderList runat=server></{0}:OrderList>")>
Public Class OrderList
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

**File**

OrderList.cs

### OrderList Methods

The methods of the OrderList class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 395) | Populates the data for the Favorites (see page 229). |
| | ToString (see page 396) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 396) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### OrderList.Fill Method

Populates the data for the Favorites (see page 229).

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

### OrderList.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

### OrderList.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

**Returns**

### OrderList Properties

The properties of the OrderList class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 396) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, set the CacheInterval property to 300 (three hundred). |
| | DefaultUserID (see page 397) | The default ID of the user for which to display a list of orders. |
| | DisplayXslt (see page 397) | Specify an external XSLT file to render the control. By default, the control uses OrderList.xsl. This file is located in: |
| | DynamicOrderParameter (see page 398) | Sets the QueryString parameter name which is used to pass an order ID dynamically. By default, this parameter is OID. |
| | DynamicProductParameter (see page 398) | The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors click a product in their order list, this parameter is passed with the product's ID to the product details page. |
| | EkItem (see page 398) | Provides access to List of OrderData (see page 1298). |
| | PageSize (see page 399) | Specifies the number of orders to show per page. |
| | Stylesheet (see page 399) | Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | TemplateProduct (see page 399) | The URL path of the template that allows a site visitor to view the details of a product in an order. When a path is entered, the product's title becomes a link that allows site visitors to navigate to the template containing details of the product; for example, the template containing a Product (see page 412) server control. |

### OrderList.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

### OrderList.DefaultUserID Property

The default ID of the user for which to display a list of orders.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The default ID of the user for which to display a list of orders.")]
public long DefaultUserID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The default ID of the user for which to display a list of orders.")>
Public Property DefaultUserID() As long
```

### OrderList.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses OrderList.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

**Important:** If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses OrderList.xsl. This file is located in: [SiteRoot]\\Workarea\\Xslt\\Commerce")]
[TypeConverter(typeof(XsltOrderTypeConverter))]
[DefaultValue("OrderList.xsl")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
```

```
<Description("Specify an external XSLT file to render the control. By default, the control
uses OrderList.xsl. This file is located in: [SiteRoot]\\Workarea\\Xslt\\Commerce")>
<TypeConverter(typeof(XsltOrderTypeConverter))>
<DefaultValue("OrderList.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### OrderList.DynamicOrderParameter Property

Sets the QueryString parameter name which is used to pass an order ID dynamically. By default, this parameter is OID.

**C#**

```
[Category("Ektron")]
[DefaultValue("oid")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Sets the QueryString parameter name which is used to pass an order ID
dynamically. By default, this parameter is OID.")]
public string DynamicOrderParameter;
```

**Visual Basic**

```
<Category("Ektron")>
<DefaultValue("oid")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Sets the QueryString parameter name which is used to pass an order ID
dynamically. By default, this parameter is OID.")>
Public Property DynamicOrderParameter() As string
```

### OrderList.DynamicProductParameter Property

The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors click a product in their order list, this parameter is passed with the product's ID to the product details page.

**C#**

```
[Category("Ektron")]
[DefaultValue("id")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("The QueryString parameter name which is used to pass the product ID to the
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors click a product in their order list, this
parameter is passed with the product?s ID to the product details page.")]
public string DynamicProductParameter;
```

**Visual Basic**

```
<Category("Ektron")>
<DefaultValue("id")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("The QueryString parameter name which is used to pass the product ID to the
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors click a product in their order list, this
parameter is passed with the product?s ID to the product details page.")>
Public Property DynamicProductParameter() As string
```

### OrderList.EkItem Property

Provides access to List of OrderData (see page 1298).

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Provides access to List of OrderData.")]
[Personalizable()]
```

```
public List<OrderData> EkItem;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Provides access to List of OrderData.")>
<Personalizable()>
Public Property EkItem() As List<OrderData>
```

### OrderList.PageSize Property

Specifies the number of orders to show per page.

**C#**

```
[Category("Ektron")]
[DefaultValue(10)]
[TypeConverter(typeof(int))]
[Description("Specifies the number of orders to show per page")]
public int PageSize;
```

**Visual Basic**

```
<Category("Ektron")>
<DefaultValue(10)>
<TypeConverter(typeof(int))>
<Description("Specifies the number of orders to show per page")>
Public Property PageSize() As Integer
```

### OrderList.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

**C#**

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[TypeConverter(typeof(OrderListCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/OrderList.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<TypeConverter(typeof(OrderListCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/OrderList.css")>
Public Property Stylesheet() As string
```

### OrderList.TemplateProduct Property

The URL path of the template that allows a site visitor to view the details of a product in an order. When a path is entered, the product's title becomes a link that allows site visitors to navigate to the template containing details of the product; for example, the template containing a Product (see page 412) server control.

**C#**

```
[Category("Ektron")]
[DefaultValue("")]
[Description("The URL path of the template that allows a site visitor to view the details of a
product in an order. When a path is entered, the product?s title becomes a link that allows
site visitors to navigate to the template containing details of the product; for example, the
template containing a Product server control.")]
public string TemplateProduct;
```

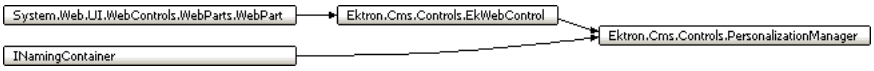
**Visual Basic**

```
<Category("Ektron")>
<DefaultValue("")>
<Description("The URL path of the template that allows a site visitor to view the details of a
product in an order. When a path is entered, the product?s title becomes a link that allows
site visitors to navigate to the template containing details of the product; for example, the
template containing a Product server control.")>
Public Property TemplateProduct() As string
```

## PersonalizationManager Class

The PersonalizationManager server control allows site users to personalize, add, edit and hide content on a Web page. The level of privileges assigned to a site user in CMS400.NET determines what personalization functionality is available when the control is active. The Personalization manager only appears when a user is logged in to the site.

**Class Hierarchy**




**C#**

```
[DefaultProperty("Text")]
[ToolboxData("<{0}:PersonalizationManager runat=server></{0}:PersonalizationManager>")]
public class PersonalizationManager : EkWebControl, INamingContainer;
```

**Visual Basic**

```
<DefaultProperty("Text")>
<ToolboxData("<{0}:PersonalizationManager runat=server></{0}:PersonalizationManager>")>
Public Class PersonalizationManager
Inherits EkWebControl
Inherits INamingContainer
```

**File**

PersonalizationManager.cs

**See Also**

The Administrator Manual section "Personalization Feature" and the Developer Manual section "Personalization Server Controls (see page 7)" > "PersonalizationManager Server Control".

### PersonalizationManager Methods

The methods of the PersonalizationManager class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 400) | Method that can be overridden and used to grab data from the back end for each server control. |

### PersonalizationManager.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## PersonalizationManager Properties

The properties of the PersonalizationManager class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AddContentExitText (see page 401) | Gets or sets the text for the Add Content (see page 788) Exit link. The default text is **Done Add Content (see page 788)**. |
| | AddContentText (see page 401) | Gets or sets the text for the Add Content (see page 788) link. The default is **Add Content (see page 788).** |
| | CurrentUserButtonText (see page 402) | Gets or sets the text for the Current User (see page 913) radio button. The default is **Current User (see page 913).** |
| | Display (see page 402) | Gets or sets the display format to vertical or horizontal for the Personalization Manager. The default is Vertical. |
| | HeaderBackColor (see page 402) | `HeaderBackColor` sets the background color of the personalization manager header. |
| | HeaderText (see page 403) | The text that appears in the header of the Personalization Manager. |
| | PersonalizeExitText (see page 403) | Gets or sets the text for the Exit link. The default is **Done**. |
| | PersonalizeScopeText (see page 403) | Gets or sets the text for the Scope. The default is **Personalization For.** |
| | PersonalizeStartText (see page 404) | Gets or sets the text for the Start link. The default is **Personalize.** |
| | PublicUserButtonText (see page 404) | Gets or sets the text for the Public User (see page 913) radio button. The default is **Public User (see page 913).** |
| | ResetLinkText (see page 404) | Gets or sets the text for the Reset link text. The default is **Reset To Default.** |
| | ResetLinkToolTip (see page 405) | Gets or sets the tool tip text for the Reset link. The default is **Reset the current user's personalization data for the page.** |
| | UrlCompareMode (see page 405) | Please select Path \| PathAndQuery |

## PersonalizationManager.AddContentExitText Property

Gets or sets the text for the Add Content (see page 788) Exit link. The default text is **Done Add Content (see page 788)**.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_ADDEXIT)]
[Description("Set text for exit add content link. Default value is set as \"" +
EkWebControl.RES_PMC_ADDEXIT + "\".")]
public string AddContentExitText;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_ADDEXIT)>
<Description("Set text for exit add content link. Default value is set as \"" +
EkWebControl.RES_PMC_ADDEXIT + "\".")>
Public Property AddContentExitText() As string
```

## PersonalizationManager.AddContentText Property

Gets or sets the text for the Add Content (see page 788) link. The default is **Add Content (see page 788).**

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
```

```
[DefaultValue(EkWebControl.RES_PMC_ADD)]
[Description("Set text for add content link. Default value is set as \"" +
EkWebControl.RES_PMC_ADD + "\".")]
public string AddContentText;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_ADD)>
<Description("Set text for add content link. Default value is set as \"" +
EkWebControl.RES_PMC_ADD + "\".")>
Public Property AddContentText() As string
```

### PersonalizationManager.CurrentUserButtonText Property

Gets or sets the text for the Current User (see page 913) radio button. The default is **Current User (see page 913).**

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_USRSCOPE)]
[Description("The display text for the current user radio button. Default value is set as \""
+ EkWebControl.RES_PMC_USRSCOPE + "\".")]
public string CurrentUserButtonText;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_USRSCOPE)>
<Description("The display text for the current user radio button. Default value is set as \""
+ EkWebControl.RES_PMC_USRSCOPE + "\".")>
Public Property CurrentUserButtonText() As string
```

### PersonalizationManager.Display Property

Gets or sets the display format to vertical or horizontal for the Personalization Manager. The default is Vertical.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(DisplayFormat.Vertical)]
[Description("The display format of the image - horizontal or vertical.")]
public DisplayFormat Display;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(DisplayFormat.Vertical)>
<Description("The display format of the image - horizontal or vertical.")>
Public Property Display() As DisplayFormat
```

### PersonalizationManager.HeaderBackColor Property

`HeaderBackColor` sets the background color of the personalization manager header.

**C#**

```
[Description("HeaderBackColor sets the header background color."), Category("Ektron")]
[Bindable(false)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color HeaderBackColor;
```

**Visual Basic**

```
<Description("HeaderBackColor sets the header background color."), Category("Ektron")>
<Bindable(false)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property HeaderBackColor() As Color
```

### PersonalizationManager.HeaderText Property

The text that appears in the header of the Personalization Manager.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_HEADER)]
[Description("The HeaderText of the control. Default value is set as \"" +
EkWebControl.RES_PMC_HEADER + "\".")]
public string HeaderText;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_HEADER)>
<Description("The HeaderText of the control. Default value is set as \"" +
EkWebControl.RES_PMC_HEADER + "\".")>
Public Property HeaderText() As string
```

### PersonalizationManager.PersonalizeExitText Property

Gets or sets the text for the Exit link. The default is **Done**.

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_EXIT)]
[Description("Set text for exit personalization mode link. Default value is set as \"" +
EkWebControl.RES_PMC_EXIT + "\".")]
public string PersonalizeExitText;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_EXIT)>
<Description("Set text for exit personalization mode link. Default value is set as \"" +
EkWebControl.RES_PMC_EXIT + "\".")>
Public Property PersonalizeExitText() As string
```

### PersonalizationManager.PersonalizeScopeText Property

Gets or sets the text for the Scope. The default is **Personalization For.**

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_SCPTXT)]
[Description("Set legend text for Personalize Scope fields. Default value is set as \"" +
EkWebControl.RES_PMC_SCPTXT + "\".")]
public string PersonalizeScopeText;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_SCPTXT)>
<Description("Set legend text for Personalize Scope fields. Default value is set as \"" +
EkWebControl.RES_PMC_SCPTXT + "\".")>
Public Property PersonalizeScopeText() As string
```

### PersonalizationManager.PersonalizeStartText Property

Gets or sets the text for the Start link. The default is **Personalize.**

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_START)]
[Description("Set text for start personalize link. Default value is set as \"" +
EkWebControl.RES_PMC_START + "\".")]
public string PersonalizeStartText;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_START)>
<Description("Set text for start personalize link. Default value is set as \"" +
EkWebControl.RES_PMC_START + "\".")>
Public Property PersonalizeStartText() As string
```

### PersonalizationManager.PublicUserButtonText Property

Gets or sets the text for the Public User (see page 913) radio button. The default is **Public User (see page 913).**

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_SHRSCOPE)]
[Description("The display text for the public user radio button. Default value is set as \""
+EkWebControl.RES_PMC_SHRSCOPE + "\".")]
public string PublicUserButtonText;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_SHRSCOPE)>
<Description("The display text for the public user radio button. Default value is set as \""
+EkWebControl.RES_PMC_SHRSCOPE + "\".")>
Public Property PublicUserButtonText() As string
```

### PersonalizationManager.ResetLinkText Property

Gets or sets the text for the Reset link text. The default is **Reset To Default.**

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_RESET)]
[Description("The display text for the reset link. Default value is set as \"" +
```

```
EkWebControl.RES_PMC_RESET + "\".")]
public string ResetLinkText;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_RESET)>
<Description("The display text for the reset link. Default value is set as \"" +
EkWebControl.RES_PMC_RESET + "\".")>
Public Property ResetLinkText() As string
```

### PersonalizationManager.ResetLinkToolTip Property

Gets or sets the tool tip text for the Reset link. The default is **Reset the current user's personalization data for the page.**

**C#**

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_RESET_ALT)]
[Description("The display text for the reset link. Default value is set as \"" +
EkWebControl.RES_PMC_RESET_ALT + ".\".")]
public string ResetLinkToolTip;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_RESET_ALT)>
<Description("The display text for the reset link. Default value is set as \"" +
EkWebControl.RES_PMC_RESET_ALT + ".\".")>
Public Property ResetLinkToolTip() As string
```

### PersonalizationManager.UrlCompareMode Property

Please select Path | PathAndQuery

**C#**

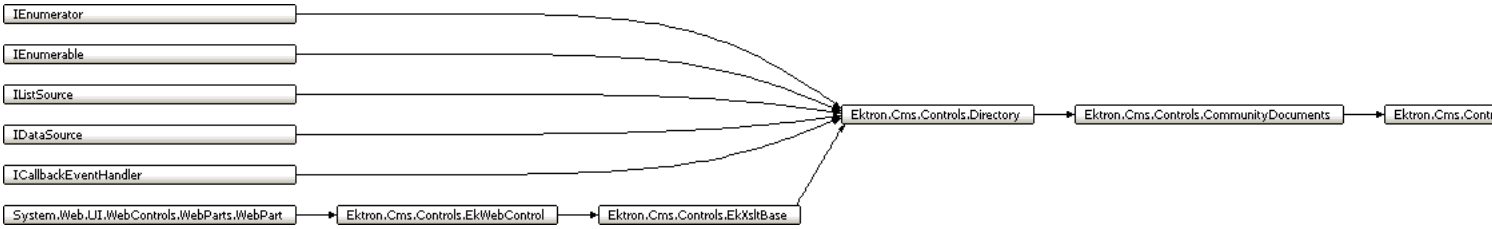
```
[Bindable(false)]
[Category("Ektron")]
[Description("")]
public string UrlCompareMode;
```

**Visual Basic**

```
<Bindable(false)>
<Category("Ektron")>
<Description("")>
Public Property UrlCompareMode() As string
```

# PhotoGallery Class

### Class Hierarchy




### C#

```
[DefaultProperty("DefaultUserID")]
[ToolboxData("<{0}:PhotoGallery runat=server></{0}:PhotoGallery>")]
public class PhotoGallery : CommunityDocuments;
```

### Visual Basic

```
<DefaultProperty("DefaultUserID")>
<ToolboxData("<{0}:PhotoGallery runat=server></{0}:PhotoGallery>")>
Public Class PhotoGallery
Inherits CommunityDocuments
```

### File

PhotoGallery.cs

### Description

This is class Ektron.Cms.Controls.PhotoGallery.

## PhotoGallery Methods

The methods of the PhotoGallery class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 406) | Populates the data for the Taxonomy (see page 842). |

## PhotoGallery.Fill Method

Populates the data for the Taxonomy (see page 842).

### C#

```
public override void Fill();
```

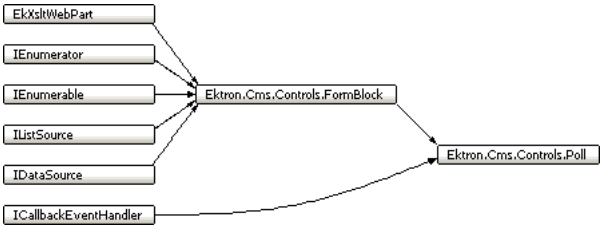
### Visual Basic

```
Public override Function Fill() As void
```

# Poll Class

The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page.

### Class Hierarchy




**C#**

```
[DefaultProperty("DefaultFormID")]
[ToolboxData("<{0}:Poll runat=server></{0}:Poll>")]
[ToolboxBitmap("Poll.bmp")]
public class Poll : Ektron.Cms.Controls.FormBlock, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultFormID")>
<ToolboxData("<{0}:Poll runat=server></{0}:Poll>")>
<ToolboxBitmap("Poll.bmp")>
Public Class Poll
Inherits Ektron.Cms.Controls.FormBlock
Inherits ICallbackEventHandler
```

**File**

Poll.cs

## Poll Methods

The methods of the Poll class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 407) | Fills in the poll area with the content information from the content database. The form data is retrieved and displayed. Only use if the content must be refreshed after an form ID change or other operation that will require regenerating the display. |
| | WrapPoll (see page 407) | Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## Poll.Fill Method

Fills in the poll area with the content information from the content database. The form data is retrieved and displayed. Only use if the content must be refreshed after an form ID change or other operation that will require regenerating the display.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## Poll.WrapPoll Method

Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public string WrapPoll(string Html, FormData formData);
```

### Visual Basic

```
Public Function WrapPoll(Html As string, formData As FormData) As string
```

### Parameters

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar |

### Returns

Html string wrapped with a toolbar

## Poll Properties

The properties of the Poll class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | EnableAjax (see page 408) | Determines if the results will show in the area of the page that contains the poll. If the value is 'true' then the results will be shown in the area containing the poll without any modification or refresh of the surrounding contents. Otherwise, the poll results are displayed in place of the page content. |
| | PollID (see page 408) | Gets or Sets the Default FormID for this Control. |

## Poll.EnableAjax Property

Determines if the results will show in the area of the page that contains the poll. If the value is 'true' then the results will be shown in the area containing the poll without any modification or refresh of the surrounding contents. Otherwise, the poll results are displayed in place of the page content.

### C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Browsable(false)]
[Obsolete("No longer supported. It is always defaults to true.")]
[Description("Whether the results shown be displayed in the poll location without disturbing
the surrounding contents.")]
public bool EnableAjax;
```

### Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Browsable(false)>
<Obsolete("No longer supported. It is always defaults to true.")>
<Description("Whether the results shown be displayed in the poll location without disturbing
the surrounding contents.")>
Public Property EnableAjax() As bool
```

## Poll.PollID Property

Gets or Sets the Default FormID for this Control.

### C#

```
[Bindable(true)]
[Category("Ektron")]
[ParenthesizePropertyName(true)]
[Editor(typeof(PollUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("The ID of the Poll to display.")]
[Personalizable(), WebBrowsable()]
public long PollID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<ParenthesizePropertyName(true)>
<Editor(typeof(PollUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("The ID of the Poll to display.")>
<Personalizable(), WebBrowsable()>
Public Property PollID() As long
```

## PostHistory Class

The PostHistory server control displays a list of posts for a given user. The posts are displayed in order by date and contain the following: Topic - the topic that contains the post Posted - the date the post was created Content (see page 788) - the content with in the post

**Class Hierarchy**

**C#**

```
[DefaultProperty("BoardID"), ToolboxData("<{0}:PostHistory runat=server></{0}:PostHistory>")]
public class PostHistory : EkWebControl;
```

**Visual Basic**

```
<DefaultProperty("BoardID"), ToolboxData("<{0}:PostHistory runat=server></{0}:PostHistory>")>
Public Class PostHistory
Inherits EkWebControl
```

**File**

PostHistory.cs

### PostHistory Methods

The methods of the PostHistory class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 409) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 409) | 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 410) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### PostHistory.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

### PostHistory.ToString Method

Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

### PostHistory.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 410) | The ID of the discussion board from which to get a user's posts. |
| | CacheInterval (see page 410) | Sets the amount of time, in seconds, the server control's data is cached. The default is 0 (zero). For example, if you want to cache the data for five minutes, you set this property to 300. |
| | MaxNumber (see page 411) | The maximum number of posts listed. The default is **10**. |
| | URLPath (see page 411) | The URL path to the page the where the Forum (see page 256) server control is located. |
| | UserID (see page 411) | The ID of the user for whom to get the post history. |

### PostHistory.BoardID Property

The ID of the discussion board from which to get a user's posts.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the discussion board from which to get a user's posts.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long BoardID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the discussion board from which to get a user's posts.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As long
```

### PostHistory.CacheInterval Property

Sets the amount of time, in seconds, the server control's data is cached. The default is 0 (zero). For example, if you want to

cache the data for five minutes, you set this property to 300.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time, in seconds, the server control?s data is cached. The
default is 0 (zero). For example, if you want to cache the data for five minutes, you set this
property to 300.")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time, in seconds, the server control?s data is cached. The
default is 0 (zero). For example, if you want to cache the data for five minutes, you set this
property to 300.")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

### PostHistory.MaxNumber Property

The maximum number of posts listed. The default is **10**.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("The maximum number of posts to display.")]
public int MaxNumber;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("The maximum number of posts to display.")>
Public Property MaxNumber() As Integer
```

### PostHistory.URLPath Property

The URL path to the page the where the Forum (see page 256) server control is located.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("The URL path to the page the where the Forum server control is located.")]
public string URLPath;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("The URL path to the page the where the Forum server control is located.")>
Public Property URLPath() As string
```

### PostHistory.UserID Property

The ID of the user for whom to get the post history.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
```

```
[DefaultValue(0)]
[Description("The ID of the user for whom to get the post history.")]
public long UserID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the user for whom to get the post history.")>
Public Property UserID() As long
```

## PostLoginEventArgs Class

Holds the Post-Login (see page 337) event arguments.

**Class Hierarchy**

**C#**

```
public class PostLoginEventArgs : EventArgs;
```

**Visual Basic**

```
Public Class PostLoginEventArgs
Inherits EventArgs
```

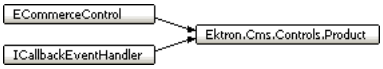
**File**

ECommerceControl.cs

## Product Class

Summary description for Product

**Class Hierarchy**




**C#**

```
[DefaultProperty("DefaultProductID"), ToolboxData("<{0}:Product runat=server></{0}:Product>")]
public class Product : ECommerceControl, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultProductID"), ToolboxData("<{0}:Product runat=server></{0}:Product>")>
Public Class Product
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

**File**

Product.cs

### Product Methods

The methods of the Product class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 413) | Populates the data for the Favorites (see page 229). |
| | RedirectToBasketPage (see page 413) | Redirects to the basket page. |

| | | |
|---|---|---|
| | ToString (see page 413) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 413) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### Product.Fill Method

Populates the data for the Favorites (see page 229).

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

### Product.RedirectToBasketPage Method

Redirects to the basket page.

**C#**

```
public bool RedirectToBasketPage();
```

**Visual Basic**

```
Public Function RedirectToBasketPage() As bool
```

**Returns**

bool

### Product.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

### Product.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

**Returns**

### Product Properties

The properties of the Product class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 414) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultProductID (see page 415) | Enter a default product's ID. This is the default item that's displayed when the template containing this control is viewed by a site visitor and a product ID is not dynamically passed. To make this server control dynamic, enter zero (0) for this property and set the DynamicProductParameter to the QueryString parameter used to pass the product ID. |
| | DisplayXslt (see page 415) | Specify an external XSLT file to render the control. By default, the control uses product.xsl. This file is located in: |
| | DynamicParameter (see page 415) | The QueryString parameter name which is used to read the product ID. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors views an item's details, the product's ID is passed to this control. |
| | EkItems (see page 416) | Provides access to EntryData (see page 1264) object. |
| | ImageGallery (see page 416) | Select the type of image gallery that's displayed with a product's information. Images that appear in the gallery are set in the Workarea's Media tab when creating or editing a product. The Gallery Display property for each image on the Media tab must be set to Yes. Choices are:<br>• List = display gallery images<br>• None = hide gallery images<br>In the Image Gallery, images are displayed at their smallest size. Once a site visitor clicks an image, a larger version of it is displayed in a thick box. |
| | OverrideXslt (see page 417) | Specifies an XSLT identified in the Edit Product (see page 412) Type Configuration screen. Use this to specify an override for the products smart-form XSLT; 0 (default) uses the XSLT defined in the product type (e.g. PackageDisplayXslt), 1 for Xslt1, 2 for Xslt2, etc. |
| | ShowAddToCart (see page 417) | Set to true if you want to the Add to Cart (see page 73) button to appear. The default is True. Setting this property to false allows you to show details of an item, but not offer it for sale. For example, you have an item that is no longer for sale, but you want to allow people who purchased the item to view its details. Also, by using code behind to dynamically set the property, you could create code that looks at your inventory system and hides the button depending on whether a product is in stock. |
| | Stylesheet (see page 417) | Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. |
| | TemplateCart (see page 418) | Enter the URL path of the template that contains the Cart (see page 73) server control. This path can be relative or absolute. |

## Product.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

### Product.DefaultProductID Property

Enter a default product's ID. This is the default item that's displayed when the template containing this control is viewed by a site visitor and a product ID is not dynamically passed. To make this server control dynamic, enter zero (0) for this property and set the DynamicProductParameter to the QueryString parameter used to pass the product ID.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enter a default product?s ID. This is the default item that?s displayed when the
template containing this control is viewed by a site visitor and a product ID is not
dynamically passed. To make this server control dynamic, enter zero (0) for this property and
set the DynamicProductParameter to the QueryString parameter used to pass the product ID. ")]
public long DefaultProductID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enter a default product?s ID. This is the default item that?s displayed when the
template containing this control is viewed by a site visitor and a product ID is not
dynamically passed. To make this server control dynamic, enter zero (0) for this property and
set the DynamicProductParameter to the QueryString parameter used to pass the product ID. ")>
Public Property DefaultProductID() As long
```

### Product.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses product.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses product.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")]
[TypeConverter(typeof(XsltProductTypeConverter))]
[DefaultValue("Product.xsl")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses product.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")>
<TypeConverter(typeof(XsltProductTypeConverter))>
<DefaultValue("Product.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### Product.DynamicParameter Property

The QueryString parameter name which is used to read the product ID. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors views an item's details, the product's ID is passed to this control.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("id")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
```

```
[Description("The QueryString parameter name which is used to read the product ID. For
example, if your QueryString parameter for products is PID, enter that in this property. Then,
when site visitors views an item?s details, the product?s ID is passed to this control.")]
[Personalizable()]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("id")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("The QueryString parameter name which is used to read the product ID. For
example, if your QueryString parameter for products is PID, enter that in this property. Then,
when site visitors views an item?s details, the product?s ID is passed to this control.")>
<Personalizable()>
Public Property DynamicParameter() As string
```

### Product.EkItems Property

Provides access to EntryData (see page 1264) object.

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Provides access to EntryData object.")]
public EntryData EkItems;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Provides access to EntryData object.")>
Public Property EkItems() As EntryData
```

### Product.ImageGallery Property

Select the type of image gallery that's displayed with a product's information. Images that appear in the gallery are set in the Workarea's Media tab when creating or editing a product. The Gallery Display property for each image on the Media tab must be set to Yes. Choices are:

- List = display gallery images
- None = hide gallery images

In the Image Gallery, images are displayed at their smallest size. Once a site visitor clicks an image, a larger version of it is displayed in a thick box.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(ImageGalleryType.List)]
[Description("elect the type of image gallery that?s displayed with a product?s information.
Images that appear in the gallery are set in the Workarea?s Media tab when creating or editing
a product.")]
public ImageGalleryType ImageGallery;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(ImageGalleryType.List)>
<Description("elect the type of image gallery that?s displayed with a product?s information.
Images that appear in the gallery are set in the Workarea?s Media tab when creating or editing
a product.")>
```

```
Public Property ImageGallery() As ImageGalleryType
```

### Product.OverrideXslt Property

Specifies an XSLT identified in the Edit Product (see page 412) Type Configuration screen. Use this to specify an override for the products smart-form XSLT; 0 (default) uses the XSLT defined in the product type (e.g. PackageDisplayXslt), 1 for Xslt1, 2 for Xslt2, etc.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Specifies an XSLT identified in the Edit Product Type Configuration screen. Used
to override the products smart-form XSLT; 0 (default) uses the XSLT defined in the product
type (e.g. PackageDisplayXslt), 1 for Xslt1, 2 for Xslt2, etc. Note that this XSLT is run on
only the smart-forms markup, and the results are then available for the DisplayXslt to
transform, as children of node <markupTransformed>.")]
public int OverrideXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Specifies an XSLT identified in the Edit Product Type Configuration screen. Used
to override the products smart-form XSLT; 0 (default) uses the XSLT defined in the product
type (e.g. PackageDisplayXslt), 1 for Xslt1, 2 for Xslt2, etc. Note that this XSLT is run on
only the smart-forms markup, and the results are then available for the DisplayXslt to
transform, as children of node <markupTransformed>.")>
Public Property OverrideXslt() As Integer
```

### Product.ShowAddToCart Property

Set to true if you want to the Add to Cart (see page 73) button to appear. The default is True. Setting this property to false allows you to show details of an item, but not offer it for sale. For example, you have an item that is no longer for sale, but you want to allow people who purchased the item to view its details. Also, by using code behind to dynamically set the property, you could create code that looks at your inventory system and hides the button depending on whether a product is in stock.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true if you want to the Add to Cart button to appear. The default is
True. Setting this property to false allows you to show details of an item, but not offer it
for sale.")]
public bool ShowAddToCart;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true if you want to the Add to Cart button to appear. The default is
True. Setting this property to false allows you to show details of an item, but not offer it
for sale.")>
Public Property ShowAddToCart() As bool
```

### Product.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute.

**C#**

```
[Category("Ektron")]
[Description("Allows selecting a different CSS file for the Product server control.")]
```

```
[TypeConverter(typeof(ProductCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/Product.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Allows selecting a different CSS file for the Product server control.")>
<TypeConverter(typeof(ProductCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/Product.css")>
Public Property Stylesheet() As string
```

### Product.TemplateCart Property

Enter the URL path of the template that contains the Cart (see page 73) server control. This path can be relative or absolute.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Enter the URL path of the template that contains the Cart server control. This
path can be relative or absolute. ")]
public string TemplateCart;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Enter the URL path of the template that contains the Cart server control. This
path can be relative or absolute. ")>
Public Property TemplateCart() As string
```

## ProductList Class

**Class Hierarchy**

**C#**

```
[DefaultProperty("DefaultProductListID"), ToolboxData("<{0}:ProductList
runat=server></{0}:ProductList>")]
public class ProductList : ECommerceControl, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultProductListID"), ToolboxData("<{0}:ProductList
runat=server></{0}:ProductList>")>
Public Class ProductList
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

**File**

ProductList.cs

**Description**

This is class Ektron.Cms.Controls.ProductList.

### ProductList Methods

The methods of the ProductList class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 419) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 419) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 419) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## ProductList.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## ProductList.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

## ProductList.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

### Returns

## ProductList Properties

The properties of the ProductList class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DisplayXslt (see page 420) | Specify an external XSLT file to render the control. By default, the control uses ProductList.xsl. This file is located in: |
| | DynamicParameter (see page 421) | Sets the QueryString parameter to read a source's ID dynamically. For example, if your QueryString parameter for a source ID is SID, enter that. |
| | EkItems (see page 421) | Returns a List of Product (see page 412) Entry Data. |
| | IdList (see page 421) | Enter a comma separated list of IDs when the SourceType (see page 424) property is set to IdList, TaxonomyList or CatalogList. |

| | | |
|---|---|---|
| | LoadingImage (see page 422) | Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif**. |
| | PageSize (see page 422) | Specify the number of items to show per page when a site visitor views the product list. If the number of items exceeds the quantity defined in this property, the list is paged and the site visitor can use the paging navigation system to move through the list. |
| | PageSpan (see page 422) | The number of pages to show before and after the current page. Enter zero to show all pages. |
| | Recursive (see page 423) | When SourceType (see page 424) is set to SourceObjectType.Catalog, setting Recursive to true will cause the control to include products from all child catalogs contained within the catalog specified by the supplied catalog ID. |
| | SortMode (see page 423) | Select the default way the product list is sorted when a site visitor first views the page. Once the page is loaded, a site visitor can sort the list from a drop down list. Choices are:<br>• **Title** - sorts in alphabetical order.<br>• **SKU** - sorts by the product number. This number is typically a unique number supplied by the producer of the item.<br>• **Price Descending** - sorts by price from highest to lowest.<br>• **Price Ascending** - sorts by price from lowest to highest. |
| | SourceId (see page 424) | The ID of the catalog or taxonomy that is being used as a product list. This property is used when the SourceType (see page 424) property is set to Catalog or Taxonomy (see page 842). |
| | SourceType (see page 424) | The type of source being used to create the list. Choices are:<br>• **Catalog** - use the SourceId (see page 424) property to specify the ID of a single catalog when this source type is selected.<br>• **CatalogList** - use the IdList (see page 421) property to specify a list of catalog IDs when this source type is selected.<br>• **Taxonomy (see page 842)** - use the SourceId (see page 424) property to specify the ID of the single taxonomy when this source type is selected.<br>• **TaxonomyList** - Use IdList (see page 421) property to specify a list of taxonomy IDs when this type is selected.<br>• **IdList (see page 421)** - Use the IdList (see page 421) property to specify a list of product... more (see page 424) |
| | Stylesheet (see page 424) | Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | TaxonomyDepth (see page 425) | Enter the number of taxonomy levels to retrieve below each taxonomy/category. Only the top level is displayed in the control. The rest of the levels are available through XML in code behind. For example, if the taxonomy is DVDs > Movies > Comedies , and you set Taxonomy (see page 842) Depth to 2, only DVDs and Movie are available in code behind. To retrieve the all of the categories for a taxonomy recursively, enter -1. A depth greater than 1 or using -1 is only useful if you create a custom output using the Taxonomy (see page 842)'s XML. The default value is 1.... more (see page 425) |
| | TemplateProduct (see page 425) | Specify the URL that contains the Product (see page 412) server control. This allows a site visitor to see the details of the product when the product link is clicked. |
| | UseMsAjax (see page 426) | When true, uses Microsoft Ajax. |

### ProductList.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses ProductList.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

**Important:** If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses ProductList.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")]
[TypeConverter(typeof(XsltProductListTypeConverter))]
[DefaultValue("ProductList.xsl")]
```

```
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses ProductList.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")>
<TypeConverter(typeof(XsltProductListTypeConverter))>
<DefaultValue("ProductList.xsl")>
Public override Property DisplayXslt() As string
```

### ProductList.DynamicParameter Property

Sets the QueryString parameter to read a source's ID dynamically. For example, if your QueryString parameter for a source ID is SID, enter that.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("Sets the QueryString parameter to read a source?s ID dynamically. For example,
if your QueryString parameter for a source ID is SID, enter that.")]
[DefaultValue("")]
public string DynamicParameter;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("Sets the QueryString parameter to read a source?s ID dynamically. For example,
if your QueryString parameter for a source ID is SID, enter that.")>
<DefaultValue("")>
Public Property DynamicParameter() As string
```

### ProductList.EkItems Property

Returns a List of Product (see page 412) Entry Data.

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Returns a List of Product Entry Data.")]
[Personalizable()]
public List<EntryData> EkItems;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Returns a List of Product Entry Data.")>
<Personalizable()>
Public Property EkItems() As List<EntryData>
```

### ProductList.IdList Property

Enter a comma separated list of IDs when the SourceType (see page 424) property is set to IdList, TaxonomyList or CatalogList.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("Enter a comma separated list of IDs when the SourceType property is set to
IdList, TaxonomyList or CatalogList.")]
```

```
[Personalizable(), WebBrowsable()]
public string IdList;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("Enter a comma separated list of IDs when the SourceType property is set to
IdList, TaxonomyList or CatalogList.")>
<Personalizable(), WebBrowsable()>
Public Property IdList() As string
```

### ProductList.LoadingImage Property

Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif**.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

### ProductList.PageSize Property

Specify the number of items to show per page when a site visitor views the product list. If the number of items exceeds the quantity defined in this property, the list is paged and the site visitor can use the paging navigation system to move through the list.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("Specify the number of items to show per page when a site visitor views the
product list. If the number of items exceeds the quantity defined in this property, the list
is paged and the site visitor can use the paging navigation system to move through the list.")]
[DefaultValue(0)]
public int PageSize;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("Specify the number of items to show per page when a site visitor views the
product list. If the number of items exceeds the quantity defined in this property, the list
is paged and the site visitor can use the paging navigation system to move through the list.")>
<DefaultValue(0)>
Public Property PageSize() As Integer
```

### ProductList.PageSpan Property

The number of pages to show before and after the current page. Enter zero to show all pages.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
```

```
[Description("The number of pages to show before and after the current page. Enter zero to
show all pages.")]
[DefaultValue(0)]
public int PageSpan;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("The number of pages to show before and after the current page. Enter zero to
show all pages.")>
<DefaultValue(0)>
Public Property PageSpan() As Integer
```

### ProductList.Recursive Property

When SourceType (see page 424) is set to SourceObjectType.Catalog, setting Recursive to true will cause the control to include products from all child catalogs contained within the catalog specified by the supplied catalog ID.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("When SourceType is set to SourceObjectType.Catalog, setting Recursive to true
will cause the control to include products from all child catalogs contained within the
catalog specified by the supplied catalog ID.")]
[DefaultValue("false")]
public bool Recursive;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("When SourceType is set to SourceObjectType.Catalog, setting Recursive to true
will cause the control to include products from all child catalogs contained within the
catalog specified by the supplied catalog ID.")>
<DefaultValue("false")>
Public Property Recursive() As bool
```

### ProductList.SortMode Property

Select the default way the product list is sorted when a site visitor first views the page. Once the page is loaded, a site visitor can sort the list from a drop down list. Choices are:

- **Title** - sorts in alphabetical order.
- **SKU** - sorts by the product number. This number is typically a unique number supplied by the producer of the item.
- **Price Descending** - sorts by price from highest to lowest.
- **Price Ascending** - sorts by price from lowest to highest.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(SortModes))]
[Description("Select the default way the product list is sorted when a site visitor first
views the page. Once the page is loaded, a site visitor can sort the list from a drop down
list. Choices are: Title, SKU, Price Ascending and Price Descending")]
[DefaultValue("SortModes.None")]
public SortModes SortMode;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(SortModes))>
<Description("Select the default way the product list is sorted when a site visitor first
views the page. Once the page is loaded, a site visitor can sort the list from a drop down
list. Choices are: Title, SKU, Price Ascending and Price Descending")>
<DefaultValue("SortModes.None")>
```

```
Public Property SortMode() As SortModes
```

### ProductList.SourceId Property

The ID of the catalog or taxonomy that is being used as a product list. This property is used when the SourceType (see page 424) property is set to Catalog or Taxonomy (see page 842).

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(long))]
[Description("The ID of the catalog or taxonomy that is being used as a product list. This
property is used when the SourceType property is set to Catalog or Taxonomy.")]
[DefaultValue(0)]
public long SourceId;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(long))>
<Description("The ID of the catalog or taxonomy that is being used as a product list. This
property is used when the SourceType property is set to Catalog or Taxonomy.")>
<DefaultValue(0)>
Public Property SourceId() As long
```

### ProductList.SourceType Property

The type of source being used to create the list. Choices are:

- **Catalog** - use the SourceId (see page 424) property to specify the ID of a single catalog when this source type is selected.
- **CatalogList** - use the IdList (see page 421) property to specify a list of catalog IDs when this source type is selected.
- **Taxonomy (see page 842)** - use the SourceId (see page 424) property to specify the ID of the single taxonomy when this source type is selected.
- **TaxonomyList** - Use IdList (see page 421) property to specify a list of taxonomy IDs when this type is selected.
- **IdList (see page 421)** - Use the IdList (see page 421) property to specify a list of product IDs when this source type is selected.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(SourceObjectType))]
[DefaultValue("SourceObjectType.Catalog")]
[Description("The type of source being used to create the list. Choices are: Catalog,
CatalogList, Taxonomy, TaxonomyList and IdList.")]
[Personalizable()]
public SourceObjectType SourceType;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(SourceObjectType))>
<DefaultValue("SourceObjectType.Catalog")>
<Description("The type of source being used to create the list. Choices are: Catalog,
CatalogList, Taxonomy, TaxonomyList and IdList.")>
<Personalizable()>
Public Property SourceType() As SourceObjectType
```

### ProductList.Stylesheet Property

Specify the path to a style sheet for use with this sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

**C#**

```
[Category("Ektron")]
[Description("Specify the path to a style sheet for use with this server control. The location
```

```
can be relative or absolute.")]
[TypeConverter(typeof(ProductListCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/ProductListIE.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Specify the path to a style sheet for use with this server control. The location
can be relative or absolute.")>
<TypeConverter(typeof(ProductListCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/ProductListIE.css")>
Public Property Stylesheet() As string
```

### ProductList.TaxonomyDepth Property

Enter the number of taxonomy levels to retrieve below each taxonomy/category. Only the top level is displayed in the control. The rest of the levels are available through XML in code behind. For example, if the taxonomy is DVDs > Movies > Comedies , and you set Taxonomy (see page 842) Depth to 2, only DVDs and Movie are available in code behind. To retrieve the all of the categories for a taxonomy recursively, enter -1. A depth greater than 1 or using -1 is only useful if you create a custom output using the Taxonomy (see page 842)'s XML. The default value is 1.

**Important!** - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("Enter the number of taxonomy levels to retrieve below each taxonomy/category.
Only the top level is displayed in the control. The rest of the levels are available through
XML in code behind.")]
[DefaultValue(0)]
public int TaxonomyDepth;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("Enter the number of taxonomy levels to retrieve below each taxonomy/category.
Only the top level is displayed in the control. The rest of the levels are available through
XML in code behind.")>
<DefaultValue(0)>
Public Property TaxonomyDepth() As Integer
```

### ProductList.TemplateProduct Property

Specify the URL that contains the Product (see page 412) server control. This allows a site visitor to see the details of the product when the product link is clicked.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(String))]
[Description("Specify the URL that contains the Product server control. This allows a site
visitor to see the details of the product when the product link is clicked.")]
[DefaultValue("")]
public string TemplateProduct;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(String))>
<Description("Specify the URL that contains the Product server control. This allows a site
visitor to see the details of the product when the product link is clicked.")>
<DefaultValue("")>
Public Property TemplateProduct() As string
```

### ProductList.UseMsAjax Property

When true, uses Microsoft Ajax.

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
[Category("Ektron")]
[TypeConverter(typeof(bool))]
[Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")]
[DefaultValue("false")]
public bool UseMsAjax;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
<Category("Ektron")>
<TypeConverter(typeof(bool))>
<Description("If true, uses Microsoft Ajax to perform callback. Otherwise uses Ektron Library
callback (default).")>
<DefaultValue("false")>
Public Property UseMsAjax() As bool
```

## ProductSearch Class

The Web Search (see page 455) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 694)" > "Web Site (see page 694) Search (see page 455)."

**Class Hierarchy**

**C#**

```
[DefaultProperty("CatalogId"), ToolboxData("<{0}:ProductSearch
runat=server></{0}:ProductSearch>")]
public class ProductSearch : ECommerceControl, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("CatalogId"), ToolboxData("<{0}:ProductSearch
runat=server></{0}:ProductSearch>")>
Public Class ProductSearch
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

**File**

ProductSearch.cs

### ProductSearch.ProductSearch Constructor

Constructor for the WebSearch (see page 505) class.

**C#**

```
public ProductSearch();
```

**Visual Basic**

```
Public Sub New()
```

## ProductSearch Methods

The methods of the ProductSearch class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 427) | Method that can be overridden and used to grab data from the back end for each server control. |
| | GetCallbackResult (see page 427) | Returns search results to the client callback function |
| | RaiseCallbackEvent (see page 427) | Processes the callback event by performing the search. |
| | Wrap (see page 427) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## ProductSearch.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## ProductSearch.GetCallbackResult Method

Returns search results to the client callback function

**C#**

```
public string GetCallbackResult();
```

**Visual Basic**

```
Public Function GetCallbackResult() As string
```

**Returns**

String containing search results.

## ProductSearch.RaiseCallbackEvent Method

Processes the callback event by performing the search.

**C#**

```
public void RaiseCallbackEvent(string eventArgs);
```

**Visual Basic**

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| eventArgs | Name-value pairs representing form fields passed by client callback function |

## ProductSearch.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

### Visual Basic

```
Public override Function Wrap(Html As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar |

### Returns

Html string wrapped with a toolbar

## ProductSearch Properties

The properties of the ProductSearch class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | ButtonImgSrc (see page 430) | If you want to display an image on the Submit button, enter the server path to it. |
| | ButtonText (see page 430) | The text used for the Submit button if no image source is identified. If an image source is identified, this is alternative text for the button. |
| | CacheInterval (see page 431) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | CatalogId (see page 431) | The folder at which the search begins. The starting folder need not be the root folder. The **Recursive (see page 436)** property determines if the search examines this folder's subfolders. |
| | CustomOrderBy (see page 431) | Provide a property's Friendly Name defined in the Indexing Service to sort search results by that property. Results can be ascending or descending based on `OrderDirection` (see page 435). If you enter an invalid property, no search results are returned. If you specify `CustomOrderBy` and `OrderBy` (see page 435), the `OrderBy` (see page 435) property is ignored. The Friendly Name of a property can be found in **Computer Management** > **Services (see page 955) and Applications** > **Indexing Service** > **Your Index** > **Properties** > **Friendly Name** column. |
| | DisplayXslt (see page 432) | Specify an external XSLT file to render the control. By default, the control uses ProductSearch.xsl. This file is located in: |
| | DynamicProductParameter (see page 432) | The QueryString parameter name which is used to pass the product ID to the product details page. |
| | EnableAdvancedLink (see page 432) | Set to **True** to display an additional tab (Advanced) on the ProductSearch (see page 426) control. |
| | EndCount (see page 433) | Get the numerical record of the first item on a page. |
| | IsInWorkArea (see page 433) | Is server control in workarea |
| | LinkTarget (see page 433) | Gets or sets the target window for the hyperlinks in the search result. The choices are:<br>• **_blank** - loads the link in a new blank window. This window is not named.<br>• **_self** - loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.<br>• **_parent** - loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.<br>• **_top** - loads the link in the full body of the window. This defaults to acting like "_self" if the... more (see page 433) |
| | LoadingImage (see page 434) | Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**. |
| | MaxCharacters (see page 434) | The maximum number of characters the Search (see page 455) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 441) property to the same number. |

| | | |
|---|---|---|
| | MaxTeaserLength (see page 434) | Limits the length of any returned content's abstract. To allow unlimited length, set to zero. This property is active only if both of these conditions are true.<br>• you use the DisplayXslt (see page 432) property to identify an XSLT and ecmteaser as a value of that property.<br>• the ShowCustomSummary (see page 438) property is set to false. If ShowCustomSummary (see page 438) is set to true, the entire summary appears in search results. |
| | OrderBy (see page 435) | The order of the search results. Choices are:<br>• **Title** - The content title (alphabetical)<br>• **ID** - The content ID number<br>• **Date Created** - The date the content was created<br>• **Date Modified** - The date the content was most recently modified<br>• **Editor** - The user who last edited the content (alphabetical)<br>• **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 694)" > "Display of Search (see page 455) Results" > "Search (see page 455) Result Ranking" |
| | OrderDirection (see page 435) | The direction in which search results are sorted. The default is Ascending.<br>• **Ascending** - Alphabetical results from A to Z; numeric values low to high; dates from oldest to most recent<br>• Descending - Alphabetical results from Z to A; numeric values high to low; dates from most recent to oldest |
| | Recursive (see page 436) | Set to True to search sub-folders of the starting folder. The starting folder is identified in the `CatalogID` property. |
| | ResultCount (see page 436) | The number of search results. This is a read-only property. |
| | ResultDataTable (see page 436) | Allows a developer to get results from the search control as a DataTable |
| | ResultsPageSize (see page 437) | An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT. |
| | ResultTagId (see page 437) | Lets you designate where search results appear. You can place search criteria in one area of a Web form and the results in another.<br>For example, you have the following tag.<br>`<span id="results"></span>`<br>In this case, enter **results** for the property value. |
| | ResultXml (see page 438) | Allows a developer to get results from the search control as XML. |
| | SearchText (see page 438) | Get or set default search text. This is optional. |
| | ShowCategories (see page 438) | If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False.<br>• **True** = Search (see page 455) across taxonomy categories<br>• **False** = Do not search across taxonomy categories |
| | ShowCustomSummary (see page 438) | If set to **True**, custom summary is used in search results. The default is False.<br>• **True** = Display custom summary in search results.<br>• **False** = Display characterization generated by indexing service in search results. |
| | ShowSearchBoxAlways (see page 439) | If set to **False**, the search box does not appear on PostBack. The default is True.<br>• **True** = Display search box on PostBack<br>• **False** = Do not display search box on PostBack |
| | StartCount (see page 439) | Get the numerical record of the last item on a page. |
| | Stylesheet (see page 439) | Specify the location of the style sheet for the search results page. Set the location relative to the site root folder. For example: `csslib\mytest.css`.<br>Leave blank to use the default style sheet, `\webroot\Workarea\csslib\global.css`. |

| | | |
|---|---|---|
| | TaxCategories (see page 440) | Is server control in workarea |
| | TaxonomyOperator (see page 440) | Select whether to use an **And** or **Or** operator when filtering results by taxonomy. By default, the property is set to **Or**. This allows for a wider range of returned results.<br>• **And** - Only results that match all categories selected in the Filter by Category tree appear. For example, if you are searching for a medical document in the Hospital and Doctor's Office categories, the document must be assigned to both categories or it is not shown.<br>• **Or** - when more than one category is selected in the Filter by Category tree, results must match at least one category... more (see page 440) |
| | TemplateCart (see page 441) | The URL for the page that contains a basket control (the URL is used when the customer clicks the 'Add to cart' button). |
| | TemplateProduct (see page 441) | The URL for the page that contains a product control (the URL is used when the customer clicks the image or title link). |
| | TextBoxSize (see page 441) | The size of the Search (see page 455) text box for user input. The width is based on the number of characters. |
| | TimeTaken (see page 442) | Gets the amount of time, in seconds, it has taken to perform the search. |
| | XmlDoc (see page 442) | This property is not supported by WebSearch (see page 505). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 432) property of the WebSearch (see page 505) server control. The XSLT filename is specified relative to the current directory. |

### ProductSearch.ButtonImgSrc Property

If you want to display an image on the Submit button, enter the server path to it.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("If you want to display an image on the Submit button, enter the server path to
it.")]
public string ButtonImgSrc;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("If you want to display an image on the Submit button, enter the server path to
it.")>
Public Property ButtonImgSrc() As string
```

### ProductSearch.ButtonText Property

The text used for the Submit button if no image source is identified. If an image source is identified, this is alternative text for the button.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_SEARCH_BTN)]
public string ButtonText;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
```

```
<Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_SEARCH_BTN)>
Public Property ButtonText() As string
```

### ProductSearch.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public new double CacheInterval;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public new Property CacheInterval() As double
```

### ProductSearch.CatalogId Property

The folder at which the search begins. The starting folder need not be the root folder. The **Recursive (see page 436)** property determines if the search examines this folder's subfolders.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The catalog at which the search begins. The starting folder need not be the root
folder. The Recursive property determines if the search examines this folder?s subfolders.")]
[Editor(typeof(DefaultCatalogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long CatalogId;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The catalog at which the search begins. The starting folder need not be the root
folder. The Recursive property determines if the search examines this folder?s subfolders.")>
<Editor(typeof(DefaultCatalogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property CatalogId() As long
```

### ProductSearch.CustomOrderBy Property

Provide a property's Friendly Name defined in the Indexing Service to sort search results by that property. Results can be ascending or descending based on `OrderDirection` (see page 435). If you enter an invalid property, no search results are returned. If you specify `CustomOrderBy` and `OrderBy` (see page 435), the `OrderBy` (see page 435) property is ignored. The Friendly Name of a property can be found in **Computer Management** > **Services (see page 955) and Applications** > **Indexing Service** > **Your Index** > **Properties** > **Friendly Name** column.

**C#**

```
[Category("Ektron"), DefaultValue(""), Description("Provide a property?s Friendly Name defined
in the Indexing Service to sort search results by that property. Results can be ascending or
descending based on OrderDirection. If you enter an invalid property, no search results are
returned.")]
public string CustomOrderBy;
```

**Visual Basic**

```
<Category("Ektron"), DefaultValue(""), Description("Provide a property?s Friendly Name defined
in the Indexing Service to sort search results by that property. Results can be ascending or
descending based on OrderDirection. If you enter an invalid property, no search results are
returned.")>
Public Property CustomOrderBy() As string
```

### ProductSearch.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses ProductSearch.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

**Important:** If you want to edit this file, it is strongly recommended that you make a copy, change its name and move it outside of the Workarea folder. If you make changes to this file and do not move it out of the Workarea, this file and the changes will be lost when you upgrade.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses ProductSearch.xsl.")]
[TypeConverter(typeof(XsltProductSearchTypeConverter))]
[DefaultValue("EcommerceProductSearch")]
[Personalizable(), WebBrowsable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses ProductSearch.xsl.")>
<TypeConverter(typeof(XsltProductSearchTypeConverter))>
<DefaultValue("EcommerceProductSearch")>
<Personalizable(), WebBrowsable()>
Public override Property DisplayXslt() As string
```

### ProductSearch.DynamicProductParameter Property

The QueryString parameter name which is used to pass the product ID to the product details page.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The QueryString parameter name which is used to pass the product ID to the
product details page.")]
[DefaultValue("id")]
public string DynamicProductParameter;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The QueryString parameter name which is used to pass the product ID to the
product details page.")>
<DefaultValue("id")>
Public Property DynamicProductParameter() As string
```

### ProductSearch.EnableAdvancedLink Property

Set to **True** to display an additional tab (Advanced) on the ProductSearch (see page 426) control.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to True to display an additional tab (Advanced) on the ProductSearch
control.")]
public bool EnableAdvancedLink;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to True to display an additional tab (Advanced) on the ProductSearch
control.")>
Public Property EnableAdvancedLink() As bool
```

### ProductSearch.EndCount Property

Get the numerical record of the first item on a page.

**C#**

```
[Browsable(false)]
public int EndCount;
```

**Visual Basic**

```
<Browsable(false)>
Public Property EndCount() As Integer
```

### ProductSearch.IsInWorkArea Property

Is server control in workarea

**C#**

```
[Browsable(false)]
public bool IsInWorkArea;
```

**Visual Basic**

```
<Browsable(false)>
Public Property IsInWorkArea() As bool
```

### ProductSearch.LinkTarget Property

Gets or sets the target window for the hyperlinks in the search result. The choices are:

- **_blank** - loads the link in a new blank window. This window is not named.
- **_self** - loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- **_parent** - loads the link in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent.
- **_top** - loads the link in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(ItemLinkTargets._self)]
[Description("The target attribute of the link.")]
public ItemLinkTargets LinkTarget;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(ItemLinkTargets._self)>
<Description("The target attribute of the link.")>
Public Property LinkTarget() As ItemLinkTargets
```

### ProductSearch.LoadingImage Property

Image to show while Ajax is fetching data. The default is **ajax-loader_circle_lg.gif"**.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")]
[DefaultValue("ajax-loader_circle_lg.gif")]
public string LoadingImage;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description(" Image to show while Ajax is fetching data (defaults to
\"ajax-loader_circle_lg.gif\").")>
<DefaultValue("ajax-loader_circle_lg.gif")>
Public Property LoadingImage() As string
```

### ProductSearch.MaxCharacters Property

The maximum number of characters the Search (see page 455) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 441) property to the same number.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(50)]
[Description("The maximum number of characters the Search text box accepts. If you enter a
value less than 50, set the TextBoxSize property to the same number.")]
public int MaxCharacters;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(50)>
<Description("The maximum number of characters the Search text box accepts. If you enter a
value less than 50, set the TextBoxSize property to the same number.")>
Public Property MaxCharacters() As Integer
```

### ProductSearch.MaxTeaserLength Property

Limits the length of any returned content's abstract. To allow unlimited length, set to zero. This property is active only if both of these conditions are true.

- you use the DisplayXslt (see page 432) property to identify an XSLT and ecmteaser as a value of that property.
- the ShowCustomSummary (see page 438) property is set to false. If ShowCustomSummary (see page 438) is set to true, the entire summary appears in search results.

**C#**

```
[Bindable(true)]
```

```
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(256)]
[Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")]
public int MaxTeaserLength;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(256)>
<Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")>
Public Property MaxTeaserLength() As Integer
```

### ProductSearch.OrderBy Property

The order of the search results. Choices are:

- **Title** - The content title (alphabetical)
- **ID** - The content ID number
- **Date Created** - The date the content was created
- **Date Modified** - The date the content was most recently modified
- **Editor** - The user who last edited the content (alphabetical)
- **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 694)" > "Display of Search (see page 455) Results" > "Search (see page 455) Result Ranking"

**C#**

```
[Browsable(true)]
[Category("Ektron")]
[Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified")]
[Personalizable(), WebBrowsable()]
public WebSearchOrder OrderBy;
```

**Visual Basic**

```
<Browsable(true)>
<Category("Ektron")>
<Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified")>
<Personalizable(), WebBrowsable()>
Public Property OrderBy() As WebSearchOrder
```

### ProductSearch.OrderDirection Property

The direction in which search results are sorted. The default is Ascending.

- **Ascending** - Alphabetical results from A to Z; numeric values low to high; dates from oldest to most recent
- Descending - Alphabetical results from Z to A; numeric values high to low; dates from most recent to oldest

**C#**

```
[Browsable(true)]
[Category("Ektron")]
[TypeConverter(typeof(SortOrderTypeConverter))]
[Description("The direction in which search results are sorted. The default is Ascending.
Choices are Ascending and Descending.")]
[Personalizable(), WebBrowsable()]
public string OrderDirection;
```

**Visual Basic**

```
<Browsable(true)>
<Category("Ektron")>
<TypeConverter(typeof(SortOrderTypeConverter))>
<Description("The direction in which search results are sorted. The default is Ascending.
Choices are Ascending and Descending.")>
<Personalizable(), WebBrowsable()>
Public Property OrderDirection() As string
```

### ProductSearch.Recursive Property

Set to True to search sub-folders of the starting folder. The starting folder is identified in the `CatalogID` property.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to True to search sub-folders of the starting folder. The starting folder is
identified in the CatalogID property.")]
public bool Recursive;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to True to search sub-folders of the starting folder. The starting folder is
identified in the CatalogID property.")>
Public Property Recursive() As bool
```

### ProductSearch.ResultCount Property

The number of search results. This is a read-only property.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The number of search results. This is a read-only property.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int ResultCount;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The number of search results. This is a read-only property.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property ResultCount() As Integer
```

### ProductSearch.ResultDataTable Property

Allows a developer to get results from the search control as a DataTable

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to get results from the search control as a DataTable.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual DataTable ResultDataTable;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to get results from the search control as a DataTable.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property ResultDataTable() As DataTable
```

### ProductSearch.ResultsPageSize Property

An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("An optional page size setting, defaults to web.config search provider page
size.")]
[Personalizable(), WebBrowsable()]
public int ResultsPageSize;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("An optional page size setting, defaults to web.config search provider page
size.")>
<Personalizable(), WebBrowsable()>
Public Property ResultsPageSize() As Integer
```

### ProductSearch.ResultTagId Property

Lets you designate where search results appear. You can place search criteria in one area of a Web form and the results in another.

For example, you have the following tag.

```
<span id="results"></span>
```

In this case, enter **results** for the property value.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("an optional parameter to specify the div control id to display the search
result")]
public string ResultTagId;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("an optional parameter to specify the div control id to display the search
result")>
Public Property ResultTagId() As string
```

### ProductSearch.ResultXml Property

Allows a developer to get results from the search control as XML.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to get results from the search control as XML.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual string ResultXml;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to get results from the search control as XML.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property ResultXml() As string
```

### ProductSearch.SearchText Property

Get or set default search text. This is optional.

**C#**

```
[Browsable(false)]
public string SearchText;
```

**Visual Basic**

```
<Browsable(false)>
Public Property SearchText() As string
```

### ProductSearch.ShowCategories Property

If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False.

- **True** = Search (see page 455) across taxonomy categories
- **False** = Do not search across taxonomy categories

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")]
public bool ShowCategories;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")>
Public Property ShowCategories() As bool
```

### ProductSearch.ShowCustomSummary Property

If set to **True**, custom summary is used in search results. The default is False.

- **True** = Display custom summary in search results.
- **False** = Display characterization generated by indexing service in search results.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, custom summary is displayed in search
results.")]
public bool ShowCustomSummary;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, custom summary is displayed in search
results.")>
Public Property ShowCustomSummary() As bool
```

### ProductSearch.ShowSearchBoxAlways Property

If set to **False**, the search box does not appear on PostBack. The default is True.

- **True** = Display search box on PostBack
- **False** = Do not display search box on PostBack

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If set to False, the search box does not appear on PostBack. The default is
True.")]
public bool ShowSearchBoxAlways;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If set to False, the search box does not appear on PostBack. The default is
True.")>
Public Property ShowSearchBoxAlways() As bool
```

### ProductSearch.StartCount Property

Get the numerical record of the last item on a page.

**C#**

```
[Browsable(false)]
public int StartCount;
```

**Visual Basic**

```
<Browsable(false)>
Public Property StartCount() As Integer
```

### ProductSearch.Stylesheet Property

Specify the location of the style sheet for the search results page. Set the location relative to the site root folder. For example: `csslib\mytest.css`.

Leave blank to use the default style sheet, `\webroot\Workarea\csslib\global.css`.

**C#**

```
[Description("Specify the location of the style sheet for the search results page. Set the
location relative to the site root folder.")]
[Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(ProductSearchCssTypeConverter))]
[DefaultValue("Workarea/csslib/Commerce/ProductSearch.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Description("Specify the location of the style sheet for the search results page. Set the
location relative to the site root folder.")>
<Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(ProductSearchCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/ProductSearch.css")>
Public Property Stylesheet() As string
```

### ProductSearch.TaxCategories Property

Is server control in workarea

**C#**

```
[Browsable(false)]
public string TaxCategories;
```

**Visual Basic**

```
<Browsable(false)>
Public Property TaxCategories() As string
```

### ProductSearch.TaxonomyOperator Property

Select whether to use an **And** or **Or** operator when filtering results by taxonomy. By default, the property is set to **Or**. This allows for a wider range of returned results.

- **And** - Only results that match all categories selected in the Filter by Category tree appear. For example, if you are searching for a medical document in the Hospital and Doctor's Office categories, the document must be assigned to both categories or it is not shown.
- **Or** - when more than one category is selected in the Filter by Category tree, results must match at least one category to be shown.

**Note:** For this property to be active, the `ShowCategories (` see page 438`)` property must be set to **True**.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Taxonomy operator used in filtering search results by taxonomy category.")]
[Personalizable(), WebBrowsable()]
public TaxCategoryOperator TaxonomyOperator;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Taxonomy operator used in filtering search results by taxonomy category.")>
<Personalizable(), WebBrowsable()>
```

```
Public Property TaxonomyOperator() As TaxCategoryOperator
```

### ProductSearch.TemplateCart Property

The URL for the page that contains a basket control (the URL is used when the customer clicks the 'Add to cart' button).

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL for the page that contains a basket control (the URL is used when the
customer clicks the 'Add to cart' button).")]
[DefaultValue("")]
public string TemplateCart;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL for the page that contains a basket control (the URL is used when the
customer clicks the 'Add to cart' button).")>
<DefaultValue("")>
Public Property TemplateCart() As string
```

### ProductSearch.TemplateProduct Property

The URL for the page that contains a product control (the URL is used when the customer clicks the image or title link).

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL for the page that contains a product control (the URL is used when the
customer clicks the image or title link).")]
[DefaultValue("")]
public string TemplateProduct;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL for the page that contains a product control (the URL is used when the
customer clicks the image or title link).")>
<DefaultValue("")>
Public Property TemplateProduct() As string
```

### ProductSearch.TextBoxSize Property

The size of the Search (see page 455) text box for user input. The width is based on the number of characters.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The size of the Search text box for user input. The width is based on the number
of characters.")]
public int TextBoxSize;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The size of the Search text box for user input. The width is based on the number
of characters.")>
Public Property TextBoxSize() As Integer
```

### ProductSearch.TimeTaken Property

Gets the amount of time, in seconds, it has taken to perform the search.

**C#**

```
[Browsable(false)]
public double TimeTaken;
```

**Visual Basic**

```
<Browsable(false)>
Public Property TimeTaken() As double
```

### ProductSearch.XmlDoc Property

This property is not supported by WebSearch (see page 505). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 432) property of the WebSearch (see page 505) server control. The XSLT filename is specified relative to the current directory.

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Description("This property is not supported by WebSearch. To retrieve the XML structure, use
the DisplayXslt property in the WebSearch server control to get search results formatted using
an XSLT. You need to either specify an XSLT name or use the default XSLTs (ecmNavigation,
ecmTeaser) provided with the server control.")]
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Description("This property is not supported by WebSearch. To retrieve the XML structure, use
the DisplayXslt property in the WebSearch server control to get search results formatted using
an XSLT. You need to either specify an XSLT name or use the default XSLTs (ecmNavigation,
ecmTeaser) provided with the server control.")>
Public override Property XmlDoc() As XmlDocument
```

## Recommendation Class

**Class Hierarchy**

**C#**

```
[DefaultProperty("DefaultProductID"), ToolboxData("<{0}:Recommendation
runat=server></{0}:Recommendation>")]
public class Recommendation : ECommerceControl, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("DefaultProductID"), ToolboxData("<{0}:Recommendation
runat=server></{0}:Recommendation>")>
Public Class Recommendation
Inherits ECommerceControl
Inherits ICallbackEventHandler
```

**File**

Recommendation.cs

## Recommendation Methods

The methods of the Recommendation class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 443) | Populates the data for the Favorites (see page 229). |
| | ToString (see page 443) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 443) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## Recommendation.Fill Method

Populates the data for the Favorites (see page 229).

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## Recommendation.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

## Recommendation.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

### Returns

## Recommendation Properties

The properties of the Recommendation class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 444) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |

| | | |
|---|---|---|
| | DefaultProductID (see page 445) | Enter a default product ID that contains either cross sell or up sell items. To make the server control dynamic, enter zero (0) in this property and set the `DynamicProductParameter` (see page 446) to QueryString parameter used to pass the product's ID. |
| | DisplayXslt (see page 445) | Specify an external XSLT file to render the control. By default, the control uses `Recommendation.xsl`. This file is located in: |
| | DynamicCartParameter (see page 445) | The QueryString parameter name which is used to pass the product ID to the Cart (see page 73) server control. For example, if your QueryString parameter for products is Product (see page 412), enter that in this property. Then, when site visitors click the Add to Cart (see page 73) button, this parameter is passed with the product's ID Web form containing the Cart (see page 73) server control. |
| | DynamicProductParameter (see page 446) | The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors click a product's title, this parameter is passed with the product's ID to the product details page. |
| | EkItem (see page 446) | Maximum number of results to be returned. |
| | PageSize (see page 447) | Specify the number of items to show per page when a site visitor views recommendations. If the number of items exceeds the quantity defined in this property, the list is paged and the site visitor can use the paging navigation system to move through the list. |
| | RecommendationType (see page 447) | The type of recommendation to show. Choices are:<br>• CrossSell - used to help sell additional items that a site visitor might be interested in. For example, if you are selling Denim Jackets, you might want to cross sell them pants or shirts.<br>• UpSell - used to sell additional items that improve on the item being purchased. For example, if you are selling 7 megapixel digital cameras, you might want show the site visitor 10 megapixel digital cameras.<br>For information on Cross Selling and Upselling, see the Administrator Manual section "Adding Cross Sell and Upsell Recommendations to a Catalog Entry" |
| | Stylesheet (see page 447) | Specify the location of a style sheet to use for the recommendation control. Set the location relative to the site root folder. For example: |
| | TemplateCart (see page 448) | The URL to navigate to when the site visitor clicks the **Add to Cart (see page 73) button**. This path can be relative or absolute. When a path is entered, the **Add to Cart (see page 73)** button appears next to the item and allows a site visitor to add the item directly to their cart. |
| | TemplateProduct (see page 448) | Specify the URL that contains the Product (see page 412) server control. This allows a site visitor to see the details of the product when the product link is clicked. This path can be relative or absolute. |

## Recommendation.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

### Recommendation.DefaultProductID Property

Enter a default product ID that contains either cross sell or up sell items. To make the server control dynamic, enter zero (0) in this property and set the DynamicProductParameter (see page 446) to QueryString parameter used to pass the product's ID.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enter a default product ID that contains either cross sell or up sell items. To
make the server control dynamic, enter zero (0) in this property and set the
DynamicProductParameter to QueryString parameter used to pass the product?s ID.")]
public long DefaultProductID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enter a default product ID that contains either cross sell or up sell items. To
make the server control dynamic, enter zero (0) in this property and set the
DynamicProductParameter to QueryString parameter used to pass the product?s ID.")>
Public Property DefaultProductID() As long
```

### Recommendation.DisplayXslt Property

Specify an external XSLT file to render the control. By default, the control uses Recommendation.xsl. This file is located in:

```
[SiteRoot]\Workarea\Xslt\Commerce
```

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Specify an external XSLT file to render the control. By default, the control
uses Recommendation.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")]
[TypeConverter(typeof(XsltRecTypeConverter))]
[DefaultValue("Recommendation.xsl")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Specify an external XSLT file to render the control. By default, the control
uses Recommendation.xsl. This file is located in [SiteRoot]\\Workarea\\Xslt\\Commerce")>
<TypeConverter(typeof(XsltRecTypeConverter))>
<DefaultValue("Recommendation.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### Recommendation.DynamicCartParameter Property

The QueryString parameter name which is used to pass the product ID to the Cart (see page 73) server control. For example, if your QueryString parameter for products is Product (see page 412), enter that in this property. Then, when site visitors click the Add to Cart (see page 73) button, this parameter is passed with the product's ID Web form containing the Cart (see page 73) server control.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The QueryString parameter name which is used to pass the product ID to the Cart
server control. For example, if your QueryString parameter for products is Product, enter that
```

```
in this property. Then, when site visitors click the \"Add to Cart\" button, this parameter is
passed with the product?s ID Web form containing the Cart server control.")]
[DefaultValue("product")]
public string DynamicCartParameter;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The QueryString parameter name which is used to pass the product ID to the Cart
server control. For example, if your QueryString parameter for products is Product, enter that
in this property. Then, when site visitors click the \"Add to Cart\" button, this parameter is
passed with the product?s ID Web form containing the Cart server control.")>
<DefaultValue("product")>
Public Property DynamicCartParameter() As string
```

### Recommendation.DynamicProductParameter Property

The QueryString parameter name which is used to pass the product ID to the product details page. For example, if your QueryString parameter for products is PID, enter that in this property. Then, when site visitors click a product's title, this parameter is passed with the product's ID to the product details page.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("id")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("The QueryString parameter name which is used to pass the product ID to the
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors click a product?s title, this parameter is
passed with the product?s ID to the product details page.")]
[Personalizable()]
public string DynamicProductParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("id")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("The QueryString parameter name which is used to pass the product ID to the
product details page. For example, if your QueryString parameter for products is PID, enter
that in this property. Then, when site visitors click a product?s title, this parameter is
passed with the product?s ID to the product details page.")>
<Personalizable()>
Public Property DynamicProductParameter() As string
```

### Recommendation.EkItem Property

Maximum number of results to be returned.

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Returns Array of Recommendation data.")]
[Personalizable()]
public RecommendationItemData EkItem;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Returns Array of Recommendation data.")>
<Personalizable()>
Public Property EkItem() As RecommendationItemData
```

### Recommendation.PageSize Property

Specify the number of items to show per page when a site visitor views recommendations. If the number of items exceeds the quantity defined in this property, the list is paged and the site visitor can use the paging navigation system to move through the list.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(int))]
[Description("Specify the number of items to show per page when a site visitor views
recommendations. If the number of items exceeds the quantity defined in this property, the
list is paged and the site visitor can use the paging navigation system to move through the
list.")]
[DefaultValue(0)]
public int PageSize;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(int))>
<Description("Specify the number of items to show per page when a site visitor views
recommendations. If the number of items exceeds the quantity defined in this property, the
list is paged and the site visitor can use the paging navigation system to move through the
list.")>
<DefaultValue(0)>
Public Property PageSize() As Integer
```

### Recommendation.RecommendationType Property

The type of recommendation to show. Choices are:

- CrossSell - used to help sell additional items that a site visitor might be interested in. For example, if you are selling Denim Jackets, you might want to cross sell them pants or shirts.
- UpSell - used to sell additional items that improve on the item being purchased. For example, if you are selling 7 megapixel digital cameras, you might want show the site visitor 10 megapixel digital cameras.

For information on Cross Selling and Upselling, see the Administrator Manual section "Adding Cross Sell and Upsell Recommendations to a Catalog Entry"

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(EkEnumeration.RecommendationType.CrossSell),
Description("The type of recommendation to show. Choices are CrossSell and Upsell.")]
public EkEnumeration.RecommendationType RecommendationType;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(EkEnumeration.RecommendationType.CrossSell),
Description("The type of recommendation to show. Choices are CrossSell and Upsell.")>
Public Property RecommendationType() As EkEnumeration.RecommendationType
```

### Recommendation.Stylesheet Property

Specify the location of a style sheet to use for the recommendation control. Set the location relative to the site root folder. For example:

```
Workarea\csslib\mytest.css
```

Leave blank to use the default style sheet.

**C#**

```
[Category("Ektron")]
[Description("Specify the location of a style sheet to use for the recomendation. Set the
location relative to the site root folder.")]
[TypeConverter(typeof(RecommendationCssTypeConverter))]
```

```
[DefaultValue("Workarea/csslib/Commerce/Recommendation.css")]
public string Stylesheet;
```

**Visual Basic**

```
<Category("Ektron")>
<Description("Specify the location of a style sheet to use for the recomendation. Set the
location relative to the site root folder.")>
<TypeConverter(typeof(RecommendationCssTypeConverter))>
<DefaultValue("Workarea/csslib/Commerce/Recommendation.css")>
Public Property Stylesheet() As string
```

### Recommendation.TemplateCart Property

The URL to navigate to when the site visitor clicks the **Add to Cart (see page 73) button**. This path can be relative or absolute. When a path is entered, the **Add to Cart (see page 73)** button appears next to the item and allows a site visitor to add the item directly to their cart.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("The URL to navigate to when the site visitor clicks the \"Add to Cart\" button.
This path can be relative or absolute. When a path is entered, the \"Add to Cart\" button
appears next to the item and allows a site visitor to add the item directly to their cart.")]
[DefaultValue("")]
public string TemplateCart;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("The URL to navigate to when the site visitor clicks the \"Add to Cart\" button.
This path can be relative or absolute. When a path is entered, the \"Add to Cart\" button
appears next to the item and allows a site visitor to add the item directly to their cart.")>
<DefaultValue("")>
Public Property TemplateCart() As string
```

### Recommendation.TemplateProduct Property

Specify the URL that contains the Product (see page 412) server control. This allows a site visitor to see the details of the product when the product link is clicked. This path can be relative or absolute.

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("Specify the URL that contains the Product server control. This allows a site
visitor to see the details of the product when the product link is clicked. This path can be
relative or absolute.")]
[DefaultValue("")]
public string TemplateProduct;
```

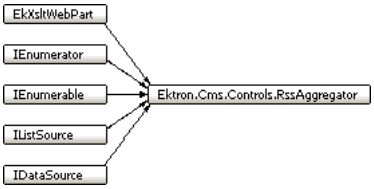
**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("Specify the URL that contains the Product server control. This allows a site
visitor to see the details of the product when the product link is clicked. This path can be
relative or absolute.")>
<DefaultValue("")>
Public Property TemplateProduct() As string
```

## RssAggregator Class

Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces.

### Class Hierarchy




**C#**

```
public class RssAggregator : EkXsltWebPart, IEnumerator, IEnumerable, IListSource, IDataSource;
```

**Visual Basic**

```
Public Class RssAggregator
Inherits EkXsltWebPart
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
```

**File**

RssAggregator.cs

## RssAggregator Methods

The methods of the RssAggregator class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | CopyTo (see page 449) | 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 449) | 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 450) | Gets a System.Collections.IEnumerator for the collection. |
| | GetView (see page 450) | Gets the view data using dataset |
| | GetViewNames (see page 450) | gets the view names |
| | MoveNext (see page 450) | Moves up the index of the Collection (see page 88). Returns false if the end of the collection has been surpassed. |
| | Reset (see page 451) | Brings the index of the collection back to zero. Does not clear the collection. |
| | this (see page 451) | Index operator into a strongly typed dataset holding some common properties for a Collection (see page 88) item. |

## RssAggregator.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

**C#**

```
public void CopyTo(Array array, int i);
```

**Visual Basic**

```
Public Function CopyTo(array As Array, i As Integer) As void
```

## RssAggregator.Fill Method

Calls up the database and updates the contents of the collection if the data is out of sync with the specified parameters.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

### RssAggregator.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

**C#**

```
public IEnumerator GetEnumerator();
```

**Visual Basic**

```
Public Function GetEnumerator() As IEnumerator
```

### RssAggregator.GetView Method

Gets the view data using dataset

**C#**

```
public DataSourceView GetView(string viewName);
```

**Visual Basic**

```
Public Function GetView(viewName As string) As DataSourceView
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 88). 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 88) 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 451) | Indicates the number of elements in your Collection (see page 88). |
| | Current (see page 452) | Gets the current object in the collection. |
| | EkItems (see page 452) | Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 88). Holds more information than the exposed strongly typed this[i] index operator. |
| | IsSynchronized (see page 452) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | LinkTarget (see page 452) | Gets or Sets the target window for the result. Choices are:<br>• **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named.<br>• **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.<br>• **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.<br>• **_top** - This target makes the link load in the full... more (see page 452) |
| | MaxResults (see page 453) | The maximum number of results to fetch. If set to zero, there is no limit. |
| | SyncRoot (see page 453) | Gets an object that can be used to synchronize the collection. |
| | URL (see page 453) | Gets or Sets the RSS Feed URL for this Control. The RSS feed path for the server control. For example: http://msdn.microsoft.com/rss.xml |

### RssAggregator.Count Property

Indicates the number of elements in your Collection (see page 88).

**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 88). Holds more information than the exposed strongly typed this[i] index operator.

**C#**

```
[Browsable(false)]
public RssItemBase EkItems;
```

**Visual Basic**

```
<Browsable(false)>
Public Property EkItems() As RssItemBase
```

### RssAggregator.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

```
[Browsable(false)]
public bool IsSynchronized;
```

**Visual Basic**

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

### RssAggregator.LinkTarget Property

Gets or Sets the target window for the result. Choices are:

- **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named.
- **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like “_self” if the document has no parent.
- **_top** - This target makes the link load in the full body of the window. This defaults to acting like “_self” if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
[Personalizable(), WebBrowsable()]
public ItemLinkTargets LinkTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
```

```
<Personalizable(), WebBrowsable()>
Public Property LinkTarget() As ItemLinkTargets
```

### RssAggregator.MaxResults Property

The maximum number of results to fetch. If set to zero, there is no limit.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
[Personalizable(), WebBrowsable()]
public int MaxResults;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit.")>
<Personalizable(), WebBrowsable()>
Public Property MaxResults() As Integer
```

### RssAggregator.SyncRoot Property

Gets an object that can be used to synchronize the collection.

**C#**

```
[Browsable(false)]
public object SyncRoot;
```

**Visual Basic**

```
<Browsable(false)>
Public Property SyncRoot() As object
```

### RssAggregator.URL Property

Gets or Sets the RSS Feed URL for this Control. The RSS feed path for the server control. For example: http://msdn.microsoft.com/rss.xml

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Gets or Sets the RSS Feed URL for this Control.")]
[Personalizable(), WebBrowsable()]
public string URL;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Gets or Sets the RSS Feed URL for this Control.")>
<Personalizable(), WebBrowsable()>
Public Property URL() As string
```

## RssItemBase Class

Container for Title (see page 454), Link (see page 454), Description (see page 454) and DatePosted (see page 454) article information.

### Class Hierarchy

**C#**

```
public class RssItemBase;
```

**Visual Basic**

```
Public Class RssItemBase
```

**File**

RssAggregator.cs

## RssItemBase Fields

The fields of the RssItemBase class are listed here.

### Public Fields

| | Name | Description |
|---|---|---|
| | DatePosted (see page 454) | The date the article was posted. |
| | Description (see page 454) | The description of the article. |
| | Link (see page 454) | The link for the article. |
| | Title (see page 454) | The title of the article. |

## RssItemBase.DatePosted Field

The date the article was posted.

**C#**

```
public DateTime DatePosted = new DateTime(0);
```

**Visual Basic**

```
Public DatePosted As DateTime = new DateTime(0)
```

## RssItemBase.Description Field

The description of the article.

**C#**

```
public string Description = "";
```

**Visual Basic**

```
Public Description As string = ""
```

## RssItemBase.Link Field

The link for the article.

**C#**

```
public string Link = "";
```

**Visual Basic**

```
Public Link As string = ""
```

## RssItemBase.Title Field

The title of the article.

**C#**

```
public string Title = "";
```

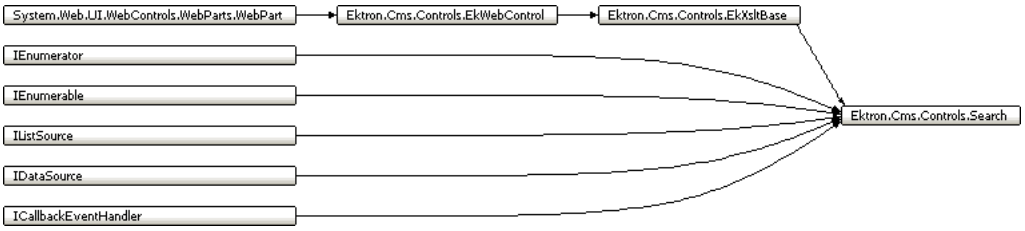
**Visual Basic**

```
Public Title As string = ""
```

## Search Class

Search object returns the control and results.

**Class Hierarchy**




**C#**

```
[DefaultProperty("FolderID"), ToolboxData("<{0}:Search runat=server></{0}:Search>")]
[Obsolete()]
[ToolboxItemFilter("Ektron.Cms.Controls.Search", ToolboxItemFilterType.Prevent)]
public class Search : EkXsltBase, IEnumerator, IEnumerable, IListSource, IDataSource,
ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("FolderID"), ToolboxData("<{0}:Search runat=server></{0}:Search>")>
<Obsolete()>
<ToolboxItemFilter("Ektron.Cms.Controls.Search", ToolboxItemFilterType.Prevent)>
Public Class Search
Inherits EkXsltBase
Inherits IEnumerator
Inherits IEnumerable
Inherits IListSource
Inherits IDataSource
Inherits ICallbackEventHandler
```

**File**

Search.cs

### Search.Search Constructor

Constructor for Search class

**C#**

```
public Search();
```

**Visual Basic**

```
Public Sub New()
```

### Search Methods

The methods of the Search class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | CopyTo (see page 456) | 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 456) | Method that can be overridden and used to grab data from the back end for each server control. |
| | GetEnumerator (see page 456) | Gets a System.Collections.IEnumerator for the collection. |
| | LoadSearch (see page 456) | |
| | MoveNext (see page 457) | Moves up the index of the Collection (see page 88). Returns false if the end of the collection has been surpassed. |
| | Reset (see page 457) | Brings the index of the collection back to zero. Does not clear the collection. |
| | SearchDisplayUI (see page 457) | |
| | this (see page 457) | Index operator into a strongly typed dataset holding some common properties for a Collection (see page 88) item. |
| | Wrap (see page 457) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## Search.CopyTo Method

Copies all the elements of the current System.Data.InternalDataCollectionBase to a one-dimensional System.Array, starting at the specified System.Data.InternalDataCollectionBase index.

**C#**

```
public void CopyTo(Array array, int index);
```

**Visual Basic**

```
Public Function CopyTo(array As Array, index As Integer) As void
```

## Search.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## Search.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

**C#**

```
public IEnumerator GetEnumerator();
```

**Visual Basic**

```
Public Function GetEnumerator() As IEnumerator
```

## Search.LoadSearch Method

**C#**

```
public void LoadSearch(ekSearchConfig SearchReq);
```

**Visual Basic**

```
Public Function LoadSearch(SearchReq As ekSearchConfig) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| SearchReq | |

### Search.MoveNext Method

Moves up the index of the Collection (see page 88). 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 88) 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 |
|---|---|---|
| | ButtonImgSrc (see page 460) | Optional: The path to the image you wish to use for the button. |
| | ButtonText (see page 460) | The text of the search button if no image is specified. |
| | ContentFieldsetLegend (see page 461) | Text (see page 221) that appears in the fieldset border around the content check boxes. This property is used only when the EnableContentFieldset (see page 464) property is set to True. The default is: **Content (see page 788) Items:** |
| | Count (see page 461) | Indicates the number of elements in the search results. |
| | Current (see page 461) | Gets the current object in the collection. |
| | Display (see page 461) | Determines if the search options are in a row (**Horizontal**) or a column (**Vertical**). |
| | EkItems (see page 462) | Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 88). Holds more information than the exposed strongly typed this[i] index operator. |
| | EmptyResultMsg (see page 462) | Optional: The return massage if the search returns no hits. |
| | EnableAdvancedLink (see page 462) | Set to true to display the link that makes the advanced properties visible. |
| | EnableAjax (see page 462) | By default, the search is set to PostBack. To enable an Ajax search, set this property to true. Ajax provides a quicker response time because the client only loads the results not the whole page. |
| | EnableArchived (see page 463) | Set to true to include Archived items in the search results. |
| | EnableArchivedBtn (see page 463) | Set to true to provide a check box so users can decide if they want to perform a Archived search. When set to False the check box is hidden but still used. |
| | EnableContent (see page 463) | Set to true to include Content (see page 788) items in the search results. |
| | EnableContentBtn (see page 464) | Set to true to provide a check box so users can decide if they want to perform a content search. When set to False the check box is hidden but still used. |
| | EnableContentFieldset (see page 464) | Creates a border around the group of content items in an advanced search. These content items include Content (see page 788), Forms and DMS Assets. The default is False. |
| | EnableDMSAssets (see page 464) | Set to true to include DMS-Asset (see page 789) items in the search results. |
| | EnableDMSAssetsBtn (see page 465) | Set to true to provide a check box so users can decide if they want to perform a DMS-Asset (see page 789) search. When set to False the check box is hidden but still used. |
| | EnableFolderFieldset (see page 465) | Set to true to wrap the Folder (see page 652)-Descriptions in a fieldset border. |
| | EnableForms (see page 465) | Set to true to include Form (see page 836) items in the search results. |
| | EnableFormsBtn (see page 465) | Set to true to provide a check box so users can decide if they want to perform a Forms search. When set to False the check box is hidden but still used. |
| | EnableLibFiles (see page 466) | Set to true to include Library (see page 664)-File items in the search results. |
| | EnableLibFilesBtn (see page 466) | Set to true to provide a check box so users can decide if they want to perform a Library (see page 664)-Files search. When set to False the check box is hidden but still used. |
| | EnableLibHyperlinks (see page 466) | Set to true to include Library (see page 664)-Hyperlink items in the search results. |
| | EnableLibHyperlinksBtn (see page 467) | Set to true to provide a check box so users can decide if they want to perform a Library (see page 664)-Hyperlinks search. When set to False the check box is hidden but still used. |
| | EnableLibImages (see page 467) | Set to true to include Library (see page 664)-Image items in the search results. |

| | | |
|---|---|---|
| | EnableLibImagesBtn (see page 467) | Set to true to display the Library (see page 664)-Images check box (false to hide, but still use). |
| | EnableLibraryFieldset (see page 468) | Set to true to wrap the Library (see page 664)-Group in a fieldset border. |
| | FolderFieldsetLegend (see page 468) | The Legend-text to display, for the Folder (see page 652)-Descriptions fieldset. |
| | FolderID (see page 468) | The folder at which the search begins. The starting folder need not be the root folder. The Recursive (see page 472) property determines if the search examines this folder's subfolders. |
| | FolderPath (see page 469) | The Path of the folder you wish to start the search. Root starts at "\". |
| | IncludeIcons (see page 469) | Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary AND ecmTeaser or or when the [$imageIcon] variable is used in an EkML file. |
| | IsSynchronized (see page 469) | Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized. |
| | LibraryFieldsetLegend (see page 469) | The Legend-text to display, for the Library (see page 664)-Group in a fieldset. |
| | LinkTarget (see page 470) | Gets or Sets the target window for the result. The choices are:<br>• **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named.<br>• **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.<br>• **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.<br>• **_top** - This target makes the link load in the... more (see page 470) |
| | MaxCharacters (see page 470) | The maximum number of characters the user can enter in the search box. |
| | MaxResults (see page 470) | The total number of results to display. If set to zero, unlimited results will be set. |
| | MaxTeaserLength (see page 471) | An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary will be truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT). |
| | OrderBy (see page 471) | Specify the order of the results. Choices are:<br>• **Title** - the title of the content<br>• **Id** - the ID number content<br>• **DateCreated** - the date the creation date of the content<br>• **DateModified** - the date the content was last modified<br>• **UserId** - ID of a user<br>• **LastEditorLname** - the last name of the last content editor<br>• **LastEditorFname** - the first name of the last content editor |
| | OrderDirection (see page 472) | Specify the order direction of the results. Choices are Ascending or Descending. |
| | PostResultsAt (see page 472) | The Path of the page you wish to post search results. (NOTE: The search control MUST exists on the posting page.) |
| | Recursive (see page 472) | Get or set whether to search subfolders of the identified root folder. The root folder is identified in the FolderID (see page 468) property. |
| | RemoveTeaserHtml (see page 472) | An optional result formatting parameter. If true, any HTML in the content items' teaser/summary is removed before it is sent to the browser. If false then any existing HTML is passed along, (this may then be further filtered by the XSLT). |
| | ResultTagId (see page 473) | Specify the div tag control ID to display the search results. This property is only active when the EnableAjax (see page 462) property is set to true. |
| | SearchText (see page 473) | Get or set the default search text. |
| | SearchType (see page 473) | Get or set the Search (see page 455) Type |
| | ShowDate (see page 473) | Select true if you wish to display the publication date on each search result. |

| | | |
|---|---|---|
| | ShowExtendedSearch (see page 474) | Allows a user to search via Searchable Property metadata.<br>• **True** = Show extended search<br>• **False** = Hide (see page 219) extended search<br>Search (see page 455) type metadata only appears on the search screen if the following are true:<br>• this property is set to True<br>• the ShowSearchOptions (see page 475) property is set to True<br>• at least one content item in the starting folder (and its subfolders if recursive is set to true) has searchable metadata applied to it |
| | ShowSearchBoxAlways (see page 474) | If set to false, the search box does not appear on PostBack. The default is True.<br>• **True** = display search box on PostBack<br>• **False** = do not display search box on PostBack |
| | ShowSearchOptions (see page 475) | Specify "True" to show search options. Set to false to see a very basic UI. |
| | Stylesheet (see page 475) | Specify the style sheet relative to site root. Leave blank for default. |
| | SyncRoot (see page 475) | Gets an object that can be used to synchronize the collection. |
| | TargetFolderDescriptionList (see page 475) | 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 476) | The width of the search box, in characters. |
| | XmlDoc (see page 476) | Returns the Xslt object for the collection. |

### Search.ButtonImgSrc Property

Optional: The path to the image you wish to use for the button.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: The path to the image you wish to use for the button.")]
public string ButtonImgSrc;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: The path to the image you wish to use for the button.")>
Public Property ButtonImgSrc() As string
```

### Search.ButtonText Property

The text of the search button if no image is specified.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_SEARCH_BTN)]
public string ButtonText;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
```

```
<Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_SEARCH_BTN)>
Public Property ButtonText() As string
```

### Search.ContentFieldsetLegend Property

Text (see page 221) that appears in the fieldset border around the content check boxes. This property is used only when the EnableContentFieldset (see page 464) property is set to True. The default is: **Content (see page 788) Items:**

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Content Items: ")]
[Description("The Legend-text to display, for the Content-Group fieldset.")]
public string ContentFieldsetLegend;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Content Items: ")>
<Description("The Legend-text to display, for the Content-Group fieldset.")>
Public Property ContentFieldsetLegend() As string
```

### Search.Count Property

Indicates the number of elements in the search results.

**C#**

```
[Category("Data")]
[Description("Indicates the number of elements in the search results.")]
public int Count;
```

**Visual Basic**

```
<Category("Data")>
<Description("Indicates the number of elements in the search results.")>
Public Property Count() As Integer
```

### Search.Current Property

Gets the current object in the collection.

**C#**

```
[Browsable(false)]
public object Current;
```

**Visual Basic**

```
<Browsable(false)>
Public Property Current() As object
```

### Search.Display Property

Determines if the search options are in a row (**Horizontal**) or a column (**Vertical**).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The display format of the image - horizontal or vertical.")]
public DisplayFormat Display;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The display format of the image - horizontal or vertical.")>
Public Property Display() As DisplayFormat
```

### Search.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection (see page 88). Holds more information than the exposed strongly typed this[i] index operator.

**C#**

```
public Ektron.Cms.Common.SearchContentItem EkItems;
```

**Visual Basic**

```
Public Property EkItems() As Ektron.Cms.Common.SearchContentItem
```

### Search.EmptyResultMsg Property

Optional: The return massage if the search returns no hits.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Optional: The return message if the search returns no hits. The default value is
set to \"res_srch_noHits\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_SEARCH_NOHITS)]
public string EmptyResultMsg;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Optional: The return message if the search returns no hits. The default value is
set to \"res_srch_noHits\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_SEARCH_NOHITS)>
Public Property EmptyResultMsg() As string
```

### Search.EnableAdvancedLink Property

Set to true to display the link that makes the advanced properties visible.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the link that makes the advanced properties visible.")]
public bool EnableAdvancedLink;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the link that makes the advanced properties visible.")>
Public Property EnableAdvancedLink() As bool
```

### Search.EnableAjax Property

By default, the search is set to PostBack. To enable an Ajax search, set this property to true. Ajax provides a quicker response time because the client only loads the results not the whole page.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Enables Ajax type of search. Ajax provides a quicker response time because the
client only loads the results not the whole page.")]
public bool EnableAjax;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Enables Ajax type of search. Ajax provides a quicker response time because the
client only loads the results not the whole page.")>
Public Property EnableAjax() As bool
```

### Search.EnableArchived Property

Set to true to include Archived items in the search results.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to include Archived items in the search results.")]
public bool EnableArchived;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to include Archived items in the search results.")>
Public Property EnableArchived() As bool
```

### Search.EnableArchivedBtn Property

Set to true to provide a check box so users can decide if they want to perform a Archived search. When set to False the check box is hidden but still used.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to display the Archived check box (false to hide, but still use).")]
public bool EnableArchivedBtn;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to display the Archived check box (false to hide, but still use).")>
Public Property EnableArchivedBtn() As bool
```

### Search.EnableContent Property

Set to true to include Content (see page 788) items in the search results.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include Content items in the search results.")]
```

```
public bool EnableContent;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include Content items in the search results.")>
Public Property EnableContent() As bool
```

### Search.EnableContentBtn Property

Set to true to provide a check box so users can decide if they want to perform a content search. When set to False the check box is hidden but still used.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the Content check box (false to hide, but still use).")]
public bool EnableContentBtn;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the Content check box (false to hide, but still use).")>
Public Property EnableContentBtn() As bool
```

### Search.EnableContentFieldset Property

Creates a border around the group of content items in an advanced search. These content items include Content (see page 788), Forms and DMS Assets. The default is False.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to wrap the Content-Group in a fieldset border.")]
public bool EnableContentFieldset;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to wrap the Content-Group in a fieldset border.")>
Public Property EnableContentFieldset() As bool
```

### Search.EnableDMSAssets Property

Set to true to include DMS-Asset (see page 789) items in the search results.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include DMS-Asset items in the search results.")]
public bool EnableDMSAssets;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include DMS-Asset items in the search results.")>
```

```
Public Property EnableDMSAssets() As bool
```

### Search.EnableDMSAssetsBtn Property

Set to true to provide a check box so users can decide if they want to perform a DMS-Asset (see page 789) search. When set to False the check box is hidden but still used.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the DMS-Asset check box (false to hide, but still use).")]
public bool EnableDMSAssetsBtn;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the DMS-Asset check box (false to hide, but still use).")>
Public Property EnableDMSAssetsBtn() As bool
```

### Search.EnableFolderFieldset Property

Set to true to wrap the Folder (see page 652)-Descriptions in a fieldset border.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to wrap the Folder-Descriptions in a fieldset border.")]
public bool EnableFolderFieldset;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to wrap the Folder-Descriptions in a fieldset border.")>
Public Property EnableFolderFieldset() As bool
```

### Search.EnableForms Property

Set to true to include Form (see page 836) items in the search results.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include Form items in the search results.")]
public bool EnableForms;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include Form items in the search results.")>
Public Property EnableForms() As bool
```

### Search.EnableFormsBtn Property

Set to true to provide a check box so users can decide if they want to perform a Forms search. When set to False the check box is hidden but still used.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the Forms check box (false to hide, but still use).")]
public bool EnableFormsBtn;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the Forms check box (false to hide, but still use).")>
Public Property EnableFormsBtn() As bool
```

### Search.EnableLibFiles Property

Set to true to include Library (see page 664)-File items in the search results.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include Library-File items in the search results.")]
public bool EnableLibFiles;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include Library-File items in the search results.")>
Public Property EnableLibFiles() As bool
```

### Search.EnableLibFilesBtn Property

Set to true to provide a check box so users can decide if they want to perform a Library (see page 664)-Files search. When set to False the check box is hidden but still used.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the Library-Files check box (false to hide, but still
use).")]
public bool EnableLibFilesBtn;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the Library-Files check box (false to hide, but still
use).")>
Public Property EnableLibFilesBtn() As bool
```

### Search.EnableLibHyperlinks Property

Set to true to include Library (see page 664)-Hyperlink items in the search results.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to include Library-Hyperlink items in the search results.")]
```

```
public bool EnableLibHyperlinks;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to include Library-Hyperlink items in the search results.")>
Public Property EnableLibHyperlinks() As bool
```

### Search.EnableLibHyperlinksBtn Property

Set to true to provide a check box so users can decide if they want to perform a Library (see page 664)-Hyperlinks search. When set to False the check box is hidden but still used.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the Library-Hyperlinks check box (false to hide, but
still use).")]
public bool EnableLibHyperlinksBtn;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the Library-Hyperlinks check box (false to hide, but
still use).")>
Public Property EnableLibHyperlinksBtn() As bool
```

### Search.EnableLibImages Property

Set to true to include Library (see page 664)-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 664)-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 664)-Group in a fieldset border.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set to true to wrap the Library-Group in a fieldset border.")]
public bool EnableLibraryFieldset;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set to true to wrap the Library-Group in a fieldset border.")>
Public Property EnableLibraryFieldset() As bool
```

### Search.FolderFieldsetLegend Property

The Legend-text to display, for the Folder (see page 652)-Descriptions fieldset.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Folder Descriptions: ")]
[Description("The Legend-text to display, for the Folder-Descriptions fieldset.")]
public string FolderFieldsetLegend;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Folder Descriptions: ")>
<Description("The Legend-text to display, for the Folder-Descriptions fieldset.")>
Public Property FolderFieldsetLegend() As string
```

### Search.FolderID Property

The folder at which the search begins. The starting folder need not be the root folder. The Recursive (see page 472) property determines if the search examines this folder's subfolders.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The folder at which the search begins. The starting folder need not be the root
folder. The Recursive property determines if the search examines this folder?s subfolders.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long FolderID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The folder at which the search begins. The starting folder need not be the root
folder. The Recursive property determines if the search examines this folder?s subfolders.")>
```

```
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderID() As long
```

### Search.FolderPath Property

The Path of the folder you wish to start the search. Root starts at "\".

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Browsable(false)]
[DefaultValue("")]
[Description("The Path of the folder you wish to start the search. Root starts at \"\\\".")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public string FolderPath;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Browsable(false)>
<DefaultValue("")>
<Description("The Path of the folder you wish to start the search. Root starts at \"\\\".")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderPath() As string
```

### Search.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary AND ecmTeaser or or when the [$imageIcon] variable is used in an EkML file.

**C#**

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
public bool IncludeIcons;
```

**Visual Basic**

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
Public Property IncludeIcons() As bool
```

### Search.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

**C#**

```
[Browsable(false)]
public bool IsSynchronized;
```

**Visual Basic**

```
<Browsable(false)>
Public Property IsSynchronized() As bool
```

### Search.LibraryFieldsetLegend Property

The Legend-text to display, for the Library (see page 664)-Group in a fieldset.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Library Items: ")]
[Description("The Legend-text to display, for the Library-Group fieldset.")]
public string LibraryFieldsetLegend;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Library Items: ")>
<Description("The Legend-text to display, for the Library-Group fieldset.")>
Public Property LibraryFieldsetLegend() As string
```

### Search.LinkTarget Property

Gets or Sets the target window for the result. The choices are:

- **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named.
- **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- **_top** - This target makes the link load in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") ]
public ItemLinkTargets LinkTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the link.") >
Public Property LinkTarget() As ItemLinkTargets
```

### Search.MaxCharacters Property

The maximum number of characters the user can enter in the search box.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(50)]
[Description("The maximum number of characters the user can enter in the search box.")]
public int MaxCharacters;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(50)>
<Description("The maximum number of characters the user can enter in the search box.")>
Public Property MaxCharacters() As Integer
```

### Search.MaxResults Property

The total number of results to display. If set to zero, unlimited results will be set.

**C#**

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue(0)]
[Description("The total number of results to display. If set to zero, unlimited results will
be set.")]
public int MaxResults;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The total number of results to display. If set to zero, unlimited results will
be set.")>
Public Property MaxResults() As Integer
```

### Search.MaxTeaserLength Property

An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary will be truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(100)]
[Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")]
public int MaxTeaserLength;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(100)>
<Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")>
Public Property MaxTeaserLength() As Integer
```

### Search.OrderBy Property

Specify the order of the results. Choices are:

- **Title** - the title of the content
- **Id** - the ID number content
- **DateCreated** - the date the creation date of the content
- **DateModified** - the date the content was last modified
- **UserId** - ID of a user
- **LastEditorLname** - the last name of the last content editor
- **LastEditorFname** - the first name of the last content editor

**C#**

```
[Browsable(true), Category("Ektron"), Description("Specify the order of the results. Choices
are: Title, Id, DateCreated, DateModified, UserId, LastEditorLname, LastEditorFname.")]
public Ektron.Cms.Controls.SearchOrder OrderBy;
```

**Visual Basic**

```
<Browsable(true), Category("Ektron"), Description("Specify the order of the results. Choices
are: Title, Id, DateCreated, DateModified, UserId, LastEditorLname, LastEditorFname.")>
Public Property OrderBy() As Ektron.Cms.Controls.SearchOrder
```

### Search.OrderDirection Property

Specify the order direction of the results. Choices are Ascending or Descending.

**C#**

```
[Category("Ektron"), TypeConverter(typeof(SortOrderTypeConverter)), Description("Specify the
order direction of the results.")]
public string OrderDirection;
```

**Visual Basic**

```
<Category("Ektron"), TypeConverter(typeof(SortOrderTypeConverter)), Description("Specify the
order direction of the results.")>
Public Property OrderDirection() As string
```

### Search.PostResultsAt Property

The Path of the page you wish to post search results. (NOTE: The search control MUST exists on the posting page.)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("#")]
[Description("The Path of the page you wish to post search results. (NOTE: The search control
MUST exists on the posting page.) ")]
public string PostResultsAt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("#")>
<Description("The Path of the page you wish to post search results. (NOTE: The search control
MUST exists on the posting page.) ")>
Public Property PostResultsAt() As string
```

### Search.Recursive Property

Get or set whether to search subfolders of the identified root folder. The root folder is identified in the FolderID (see page 468) property.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Select true if you wish to search the subfolders of the selected Folder")]
public bool Recursive;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Select true if you wish to search the subfolders of the selected Folder")>
Public Property Recursive() As bool
```

### Search.RemoveTeaserHtml Property

An optional result formatting parameter. If true, any HTML in the content items' teaser/summary is removed before it is sent to the browser. If false then any existing HTML is passed along, (this may then be further filtered by the XSLT).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
```

```
[Description("An optional search-result formatting parameter that removes all HTML from each
content-items' teaser/summary before sending it to the browser.")]
public bool RemoveTeaserHtml;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("An optional search-result formatting parameter that removes all HTML from each
content-items' teaser/summary before sending it to the browser.")>
Public Property RemoveTeaserHtml() As bool
```

### Search.ResultTagId Property

Specify the div tag control ID to display the search results. This property is only active when the EnableAjax (see page 462) property is set to true.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("An optional parameter to specify the div control id to display the search
result.")]
public string ResultTagId;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("An optional parameter to specify the div control id to display the search
result.")>
Public Property ResultTagId() As string
```

### Search.SearchText Property

Get or set the default search text.

**C#**

```
[Browsable(false)]
public string SearchText;
```

**Visual Basic**

```
<Browsable(false)>
Public Property SearchText() As string
```

### Search.SearchType Property

Get or set the Search (see page 455) Type

**C#**

```
[Browsable(false)]
public Ektron.Cms.Controls.SearchTypes SearchType;
```

**Visual Basic**

```
<Browsable(false)>
Public Property SearchType() As Ektron.Cms.Controls.SearchTypes
```

### Search.ShowDate Property

Select true if you wish to display the publication date on each search result.

**C#**

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue(false)]
[Description("Select true if you wish to display the publication date on each search result.")]
public bool ShowDate;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Select true if you wish to display the publication date on each search result.")>
Public Property ShowDate() As bool
```

### Search.ShowExtendedSearch Property

Allows a user to search via Searchable Property metadata.

- **True** = Show extended search
- **False** = Hide (see page 219) extended search

Search (see page 455) type metadata only appears on the search screen if the following are true:

- this property is set to True
- the ShowSearchOptions (see page 475) property is set to True
- at least one content item in the starting folder (and its subfolders if recursive is set to true) has searchable metadata applied to it

**C#**

```
[Category("Ektron"), Description("Set to true to allow a user to search via the metadata type:
Searchable Property.")]
public bool ShowExtendedSearch;
```

**Visual Basic**

```
<Category("Ektron"), Description("Set to true to allow a user to search via the metadata type:
Searchable Property.")>
Public Property ShowExtendedSearch() As bool
```

**Notes**

The folderID property determines the search's starting folder.

### Search.ShowSearchBoxAlways Property

If set to false, the search box does not appear on PostBack. The default is True.

- **True** = display search box on PostBack
- **False** = do not display search box on PostBack

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to false, the search box will not be visible on
PostBack.")]
public bool ShowSearchBoxAlways;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to false, the search box will not be visible on
PostBack.")>
Public Property ShowSearchBoxAlways() As bool
```

### Search.ShowSearchOptions Property

Specify "True" to show search options. Set to false to see a very basic UI.

**C#**

```
[Category("Ektron"), Description("Specify True to show search options. Set to False to see a
very basic UI.")]
public bool ShowSearchOptions;
```

**Visual Basic**

```
<Category("Ektron"), Description("Specify True to show search options. Set to False to see a
very basic UI.")>
Public Property ShowSearchOptions() As bool
```

### Search.Stylesheet Property

Specify the style sheet relative to site root. Leave blank for default.

**C#**

```
[Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
public string Stylesheet;
```

**Visual Basic**

```
<Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
Public Property Stylesheet() As string
```

### Search.SyncRoot Property

Gets an object that can be used to synchronize the collection.

**C#**

```
[Browsable(false)]
public object SyncRoot;
```

**Visual Basic**

```
<Browsable(false)>
Public Property SyncRoot() As object
```

### Search.TargetFolderDescriptionList Property

An optional comma-delimited string, which allows the user to override the folders to search. For example, set to "Photos=1,NewsHeadlines=0" to search only the Photos folder by default, and allow the user to also select the "NewsHeadlines" folder. Also searches these folders' subfolders, returning any matches found there.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("An optional comma-delimited string of folder descriptions, which allows the user
to override which folders to search: For example, set to 'Health Care=1,News=0' to search only
folders with description containing 'Health Care', by default.")]
[Obsolete("This attribute has been deprecated and is no longer supported as of version 6.1")]
[Browsable(false)]
public string TargetFolderDescriptionList;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("An optional comma-delimited string of folder descriptions, which allows the user
to override which folders to search: For example, set to 'Health Care=1,News=0' to search only
folders with description containing 'Health Care', by default.")>
<Obsolete("This attribute has been deprecated and is no longer supported as of version 6.1")>
<Browsable(false)>
Public Property TargetFolderDescriptionList() As string
```

### Search.TextBoxSize Property

The width of the search box, in characters.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The width of the search box, in characters.")]
public int TextBoxSize;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The width of the search box, in characters.")>
Public Property TextBoxSize() As Integer
```

### Search.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## Sitemap Class

The Sitemap server control utilizes the folder breadcrumb information in the Workarea to display a sitemap of your site. By choosing the starting point of the sitemap, the max levels to display and applying a style class, you can customize the sitemap.

**Class Hierarchy**

**C#**

```
[DefaultProperty("Text"), ToolboxData("<{0}:Sitemap runat=\"server\"></{0}:Sitemap>")]
public class Sitemap : EkXsltBase;
```

**Visual Basic**

```
<DefaultProperty("Text"), ToolboxData("<{0}:Sitemap runat=\"server\"></{0}:Sitemap>")>
Public Class Sitemap
Inherits EkXsltBase
```

**File**

Sitemap.cs

## Sitemap Methods

The methods of the Sitemap class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 477) | Method that can be overridden and used to grab data from the back end for each server control. |

## Sitemap.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## Sitemap Properties

The properties of the Sitemap class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 477) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | ClassName (see page 478) | Stylesheet (see page 480) class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default. |
| | DisplayType (see page 478) | Determines how the information on the page is displayed.<br>• **None** - databind only<br>• **List** - lists the title of every content block in the folder<br>• **ListWithDescriptions** - lists the title of every content block in the folder plus the content summary<br>• **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page |
| | DisplayXslt (see page 479) | The Xslt to use to render the control (optional). |
| | FlatTopLevel (see page 479) | Includes site nodes from the parent folder but not their items. |
| | FolderID (see page 479) | The ID of the folder from which to select data. Root is 0. |
| | MaxLevel (see page 480) | Get or set the maximum amount of levels to show. Set it to 0 (zero) for unlimited. |
| | StartingLevel (see page 480) | Get or set to the starting level of the site map. If set to 0 (zero), starts from the root. |
| | Stylesheet (see page 480) | Specify the location of the style sheet used to display results. Leave blank to use the default style sheet, `\webroot\Workarea\csslib\sitemap.css`. |
| | XmlDoc (see page 480) | Returns the XML object for the sitemap. |

## Sitemap.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
```

```
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
Public virtual Property CacheInterval() As double
```

### Sitemap.ClassName Property

Stylesheet (see page 480) class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Browsable(true)]
[Description("Stylesheet class to format the HTML. It must exists in
[site]/workarea/csslib/sitemap.css. Leave blank to accept default.")]
public string ClassName;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Browsable(true)>
<Description("Stylesheet class to format the HTML. It must exists in
[site]/workarea/csslib/sitemap.css. Leave blank to accept default.")>
Public Property ClassName() As string
```

### Sitemap.DisplayType Property

Determines how the information on the page is displayed.

- **None** - databind only
- **List** - lists the title of every content block in the folder
- **ListWithDescriptions** - lists the title of every content block in the folder plus the content summary
- **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Browsable(true)]
[Description("Display type controls how sitemap gets rendered on the page.")]
[Editor(typeof(SitemapDisplayTypeConverter), typeof(System.Drawing.Design.UITypeEditor))]
public string DisplayType;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Browsable(true)>
<Description("Display type controls how sitemap gets rendered on the page.")>
```

```
<Editor(typeof(SitemapDisplayTypeConverter), typeof(System.Drawing.Design.UITypeEditor))>
Public Property DisplayType() As string
```

### Sitemap.DisplayXslt Property

The Xslt to use to render the control (optional).

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltTypeConverter))]
[DefaultValue("ecmNavigation")]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltTypeConverter))>
<DefaultValue("ecmNavigation")>
Public override Property DisplayXslt() As string
```

### Sitemap.FlatTopLevel Property

Includes site nodes from the parent folder but not their items.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Includes site nodes from the parent folder but not their items.")]
[DefaultValue(false)]
public bool FlatTopLevel;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Includes site nodes from the parent folder but not their items.")>
<DefaultValue(false)>
Public Property FlatTopLevel() As bool
```

### Sitemap.FolderID Property

The ID of the folder from which to select data. Root is 0.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder from which to select data. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long FolderID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder from which to select data. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderID() As long
```

### Sitemap.MaxLevel Property

Get or set the maximum amount of levels to show. Set it to 0 (zero) for unlimited.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("Set the maximum level of the sitemap. Set it to 0 (zero) for unlimited.")]
public int MaxLevel;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("Set the maximum level of the sitemap. Set it to 0 (zero) for unlimited.")>
Public Property MaxLevel() As Integer
```

### Sitemap.StartingLevel Property

Get or set to the starting level of the site map. If set to 0 (zero), starts from the root.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Browsable(true)]
[Description("Set the starting level of the sitemap. Set it to 0 for start from root.")]
public int StartingLevel;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Browsable(true)>
<Description("Set the starting level of the sitemap. Set it to 0 for start from root.")>
Public Property StartingLevel() As Integer
```

### Sitemap.Stylesheet Property

Specify the location of the style sheet used to display results. Leave blank to use the default style sheet, `\webroot\Workarea\csslib\sitemap.css`.

**C#**

```
[Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
public string Stylesheet;
```

**Visual Basic**

```
<Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
Public Property Stylesheet() As string
```

### Sitemap.XmlDoc Property

Returns the XML object for the sitemap.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## SmartMenu Class

Class extends the Menu (see page 358)-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

### SmartMenu Properties

The properties of the SmartMenu class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AutoCollapseBranches (see page 482) | If you set to **True**, whenever a new submenu opens, all other submenus close. If **False**, other submenus remain open when a new one opens. |
| | DisplayXslt (see page 482) | The Xslt to use to render the Menu (optional) |
| | Enable508Compliance (see page 482) | If you set to **True**, the menu will comply with Section 508. This makes the menus accessible to disabled users. |
| | Enable508NavigationHeadings (see page 483) | Set to **True** to enable accessibility (508-Compliant) navigation headings. This is only used if the Enable508Compliance (see page 482) property is set to True. |
| | EnableMouseOverPopUp (see page 483) | If you set to **True**, submenus appear as soon as the cursor moves over them. If you set to **False**, submenus only appear if a site visitor clicks them or a keyboard equivalent. |
| | EnableSmartOpen (see page 483) | If you set to **True**, any submenu on the Smart Menu (see page 358) can be set to open automatically. To learn about how to do this, see the Ektron CMS400.NET Administrator Manual section “Menu (see page 358) Feature” > “Assigning Folders to a Menu (see page 358).”<br>If you set to **False**, even if all other conditions are present, submenus on a Smart Menu (see page 358) do not automatically open. This property lets you design pages whose submenus do not normally open by default. Under some circumstances, such submenus look cluttered. |
| | EnableSubmenuLinkPathMatching (see page 484) | Set to **True** to enable auto-submenu opening based on link match (if all else fails). |
| | EnableSubmenuTemplateMatching (see page 484) | Set to **True** to enable auto-submenu opening based on a template filename match (if all else fails). |
| | LaunchLinksInNewWindow (see page 484) | If you set to **True** and a site visitor selects a menu option, it appears in a new browser window. If **False**, the new page replaces the current page in the browser. |
| | ShowRootFolder (see page 485) | If you set to **True**, a menu option listing the title of the root folder appears (normally not desired). If you set to **False**, the title of the root folder does not appear. |
| | SingleEditButton (see page 485) | Set to **True** if you want a single Edit menu item (and no Add menu item) to appear on the menu when you are logged in. If you set to **False**, Edit Menu (see page 358) and Add Menu (see page 358) options appear at the bottom of every menu and submenu. |
| | StartCollapsed (see page 486) | If you set to **True**, all submenus are closed when the menu initially displays. |
| | StartWithRootCollapsed (see page 486) | If you set to **True**, all submenus are closed when the menu initially displays. |

| | | |
|---|---|---|
| | Stylesheet (see page 486) | Enter the style sheet that will determine the appearance of the menus. Menu (see page 358) style sheets reside in the `site root/workarea/csslib/smartmenu` folder. You can modify any of the standard style sheets or create your own. (optional) |

### SmartMenu.AutoCollapseBranches Property

If you set to **True**, whenever a new submenu opens, all other submenus close. If **False**, other submenus remain open when a new one opens.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to automatically close other menu branches when a new one is
opened.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool AutoCollapseBranches;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to automatically close other menu branches when a new one is
opened.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property AutoCollapseBranches() As bool
```

### SmartMenu.DisplayXslt Property

The Xslt to use to render the Menu (optional)

**C#**

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string DisplayXslt;
```

**Visual Basic**

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property DisplayXslt() As string
```

### SmartMenu.Enable508Compliance Property

If you set to **True**, the menu will comply with Section 508. This makes the menus accessible to disabled users.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable 508 Compliance (make menus accessible to disabled
users).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool Enable508Compliance;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable 508 Compliance (make menus accessible to disabled
users).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property Enable508Compliance() As bool
```

### SmartMenu.Enable508NavigationHeadings Property

Set to **True** to enable accessibility (508-Compliant) navigation headings. This is only used if the Enable508Compliance (see page 482) property is set to True.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable accessibility (508-Compliant) navigation headings. (Only
used if Enable508Compliance is true.)")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool Enable508NavigationHeadings;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable accessibility (508-Compliant) navigation headings. (Only
used if Enable508Compliance is true.)")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property Enable508NavigationHeadings() As bool
```

### SmartMenu.EnableMouseOverPopUp Property

If you set to **True**, submenus appear as soon as the cursor moves over them. If you set to **False**, submenus only appear if a site visitor clicks them or a keyboard equivalent.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable submenu activation (pop-ups) on mouse-over.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool EnableMouseOverPopUp;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable submenu activation (pop-ups) on mouse-over.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property EnableMouseOverPopUp() As bool
```

### SmartMenu.EnableSmartOpen Property

If you set to **True**, any submenu on the Smart Menu (see page 358) can be set to open automatically. To learn about how to do this, see the Ektron CMS400.NET Administrator Manual section “Menu (see page 358) Feature” > “Assigning Folders to a Menu (see page 358).”

If you set to **False**, even if all other conditions are present, submenus on a Smart Menu (see page 358) do not automatically open. This property lets you design pages whose submenus do not normally open by default. Under some circumstances, such submenus look cluttered.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("When true, menu will auto-open to show matching IDs or folder associations.")]
[TypeConverter(typeof(BooleanConverter))]
```

```
[DefaultValue(true)]
public bool EnableSmartOpen;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("When true, menu will auto-open to show matching IDs or folder associations.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property EnableSmartOpen() As bool
```

### SmartMenu.EnableSubmenuLinkPathMatching Property

Set to **True** to enable auto-submenu opening based on link match (if all else fails).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable auto-submenu opening based on link match (if all else
fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuLinkPathMatching;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable auto-submenu opening based on link match (if all else
fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property EnableSubmenuLinkPathMatching() As bool
```

### SmartMenu.EnableSubmenuTemplateMatching Property

Set to **True** to enable auto-submenu opening based on a template filename match (if all else fails).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to enable auto-submenu opening based on a template filename match
(if all else fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuTemplateMatching;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to enable auto-submenu opening based on a template filename match
(if all else fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property EnableSubmenuTemplateMatching() As bool
```

### SmartMenu.LaunchLinksInNewWindow Property

If you set to **True** and a site visitor selects a menu option, it appears in a new browser window. If **False**, the new page replaces the current page in the browser.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to launch external links in a new window.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool LaunchLinksInNewWindow;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to launch external links in a new window.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property LaunchLinksInNewWindow() As bool
```

### SmartMenu.ShowRootFolder Property

If you set to **True**, a menu option listing the title of the root folder appears (normally not desired). If you set to **False**, the title of the root folder does not appear.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to make the root menu folder-button visible (normally not
desired).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool ShowRootFolder;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to make the root menu folder-button visible (normally not
desired).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property ShowRootFolder() As bool
```

### SmartMenu.SingleEditButton Property

Set to **True** if you want a single Edit menu item (and no Add menu item) to appear on the menu when you are logged in. If you set to **False**, Edit Menu (see page 358) and Add Menu (see page 358) options appear at the bottom of every menu and submenu.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to only render a single Edit button (and no Add button) when logged
in.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool SingleEditButton;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to only render a single Edit button (and no Add button) when logged
in.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property SingleEditButton() As bool
```

### SmartMenu.StartCollapsed Property

If you set to **True**, all submenus are closed when the menu initially displays.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to initially render menu with submenus closed.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
public bool StartCollapsed;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to initially render menu with submenus closed.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
Public Property StartCollapsed() As bool
```

### SmartMenu.StartWithRootCollapsed Property

If you set to **True**, all submenus are closed when the menu initially displays.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to True to initially render menu with root-menu closed: it's contents made
invisible.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool StartWithRootCollapsed;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to True to initially render menu with root-menu closed: it's contents made
invisible.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property StartWithRootCollapsed() As bool
```

### SmartMenu.Stylesheet Property

Enter the style sheet that will determine the appearance of the menus. Menu (see page 358) style sheets reside in the `site root/workarea/csslib/smartmenu` folder. You can modify any of the standard style sheets or create your own. (optional)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Style-Sheet to use to render the Menu.")]
[TypeConverter(typeof(StyleSheetMenuTypeConverter))]
[DefaultValue("shuttermenu_workarea_vista.css")]
public override string Stylesheet;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Style-Sheet to use to render the Menu.")>
<TypeConverter(typeof(StyleSheetMenuTypeConverter))>
<DefaultValue("shuttermenu_workarea_vista.css")>
Public override Property Stylesheet() As string
```

# SocialBar Class

Summary description for ListSummary (see page 325).

### Class Hierarchy

### C#

```
[DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:SocialBar
runat=server></{0}:SocialBar>")]
public class SocialBar : EkXsltWebPart, ICallbackEventHandler;
```

### Visual Basic

```
<DefaultProperty("DefaultObjectID"), ToolboxData("<{0}:SocialBar
runat=server></{0}:SocialBar>")>
Public Class SocialBar
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

### File

SocialBar.cs

## SocialBar Methods

The methods of the SocialBar class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 487) | Populates the data for the ListSummary (see page 325). |
| | GetCallbackResult (see page 487) | Wraps the resulting callback string. |
| | ToString (see page 488) | Overrides the system ToString function. |
| | Wrap (see page 488) | Wrap the Html toolbar. |

## SocialBar.Fill Method

Populates the data for the ListSummary (see page 325).

### C#

```
public override void Fill();
```

### Visual Basic

```
Public override Function Fill() As void
```

## SocialBar.GetCallbackResult Method

Wraps the resulting callback string.

### C#

```
public string GetCallbackResult();
```

### Visual Basic

```
Public Function GetCallbackResult() As string
```

### Returns

The callback result string

## SocialBar.ToString Method

Overrides the system ToString function.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

**Returns**

String

## SocialBar.Wrap Method

Wrap the Html toolbar.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Html | |

## SocialBar Properties

The properties of the SocialBar class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 489) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultObjectID (see page 489) | The object ID for the object that the social bar will be drawn for. |
| | DisplayXslt (see page 489) | The Xslt to use to render the Control (optional). |
| | DynamicObjectParameter (see page 490) | Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary. |
| | EkItems (see page 490) | A collection of the buttons on the Social Bar. |
| | IncludeIcons (see page 490) | Determines whether or not to display an icon next to your navigation lists. |
| | InviteURL (see page 491) | The URL to use for the Invite (see page 316) item. {2} is the object id and {3} is the object type. If for a group and left blank, will show the group invite window. |
| | Items (see page 491) | Comma separated list of the items for the social bar. |
| | MarkupLanguage (see page 491) | Enter name of a markup file that manages the display of this server control. |
| | MessagingURL (see page 492) | The URL where the messaging control is. |
| | ObjectType (see page 492) | Sort order key. |
| | Stylesheet (see page 492) | Specifies the Css to which this control will provide a link. |
| | SuppressWrapperTags (see page 493) | Suppresses the output of span/div tags around the control. |
| | XmlDoc (see page 493) | Returns the Xml object loaded for ListSummary (see page 325). |

### SocialBar.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

### SocialBar.DefaultObjectID Property

The object ID for the object that the social bar will be drawn for.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID for the object that the social bar will be drawn for.")]
public long DefaultObjectID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID for the object that the social bar will be drawn for.")>
Public Property DefaultObjectID() As long
```

### SocialBar.DisplayXslt Property

The Xslt to use to render the Control (optional).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltSocialBarTypeConverter))]
[DefaultValue("(Path To Custom Xslt)")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
```

```
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltSocialBarTypeConverter))>
<DefaultValue("(Path To Custom Xslt)")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### SocialBar.DynamicObjectParameter Property

Checks the QueryString for this value and replaces the list summary with that content block when specified. Leave blank to always display the list summary.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default object ID.")]
[Personalizable()]
public string DynamicObjectParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Description("Checks the QueryString for this value and uses that value when specified. Leave
blank to always use the default object ID.")>
<Personalizable()>
Public Property DynamicObjectParameter() As string
```

### SocialBar.EkItems Property

A collection of the buttons on the Social Bar.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("A collection of the buttons on the Social Bar.")]
[Browsable(false)]
public Ektron.Cms.SocialBarItem EkItems;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("A collection of the buttons on the Social Bar.")>
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.SocialBarItem
```

### SocialBar.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists.

**C#**

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.")]
[DefaultValue(true)]
[Category("Ektron")]
[Personalizable()]
public bool IncludeIcons;
```

**Visual Basic**

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.")>
<DefaultValue(true)>
<Category("Ektron")>
<Personalizable()>
Public Property IncludeIcons() As bool
```

### SocialBar.InviteURL Property

The URL to use for the Invite (see page 316) item. {2} is the object id and {3} is the object type. If for a group and left blank, will show the group invite window.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("The URL to use for the Invite item. {2} is the object id and {3} is the object
type. If for a group and left blank, will show the group invite window.")]
public string InviteURL;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("The URL to use for the Invite item. {2} is the object id and {3} is the object
type. If for a group and left blank, will show the group invite window.")>
Public Property InviteURL() As string
```

### SocialBar.Items Property

Comma separated list of the items for the social bar.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Addto, Email, Digg, Delicious, Print")]
[Description("Comma separated list of the items for the social bar.")]
public string Items;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Addto, Email, Digg, Delicious, Print")>
<Description("Comma separated list of the items for the social bar.")>
Public Property Items() As string
```

### SocialBar.MarkupLanguage Property

Enter name of a markup file that manages the display of this server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Enter name of a markup
file that manages the display of this server control.")>
Public Property MarkupLanguage() As string
```

### SocialBar.MessagingURL Property

The URL where the messaging control is.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("The URL where the messaging control is.")]
public string MessagingURL;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("The URL where the messaging control is.")>
Public Property MessagingURL() As string
```

### SocialBar.ObjectType Property

Sort order key.

**C#**

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Common.EkEnumeration.CMSSocialBarTypes.Content)]
[Description("The type of the object that the social bar will be drawn for.")]
[Category("Ektron")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")]
public Ektron.Cms.Common.EkEnumeration.CMSSocialBarTypes ObjectType;
```

**Visual Basic**

```
<Bindable(true)>
<DefaultValue(Ektron.Cms.Common.EkEnumeration.CMSSocialBarTypes.Content)>
<Description("The type of the object that the social bar will be drawn for.")>
<Category("Ektron")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")>
Public Property ObjectType() As Ektron.Cms.Common.EkEnumeration.CMSSocialBarTypes
```

### SocialBar.Stylesheet Property

Specifies the Css to which this control will provide a link.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

### SocialBar.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

**C#**

```
public override bool SuppressWrapperTags;
```

**Visual Basic**

```
Public override Property SuppressWrapperTags() As bool
```

### SocialBar.XmlDoc Property

Returns the Xml object loaded for ListSummary (see page 325).

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## TagCloud Class

The TagCloud server control allows you to visually depict a weighted list of tags assigned to users or community groups. The weight of each tag is determined by the amount of times a tag is assigned and represented by font size. The large the font size the more times a tag is assigned. Tags (see page 771) are keywords that are associated with a user or community group in CMS400.NET.

**Class Hierarchy**

**C#**

```
[DefaultProperty(""), ToolboxData("<{0}:TagCloud runat=server></{0}:TagCloud>")]
public class TagCloud : EkXsltWebPart, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty(""), ToolboxData("<{0}:TagCloud runat=server></{0}:TagCloud>")>
Public Class TagCloud
Inherits EkXsltWebPart
Inherits ICallbackEventHandler
```

**File**

TagCloud.cs

### TagCloud Methods

The methods of the TagCloud class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 494) | Populates the data for the Favorites (see page 229). |
| | ToString (see page 494) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 494) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### TagCloud.Fill Method

Populates the data for the Favorites (see page 229).

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

### TagCloud.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

### TagCloud.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

**Returns**

### TagCloud Properties

The properties of the TagCloud class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CacheInterval (see page 495) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DisplayXslt (see page 496) | Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.(optional). |
| | DynamicParameter (see page 496) | |
| | MarkupLanguage (see page 496) | Identify the template markup file that controls the display of the server control. |
| | MaxTagsDisplayed (see page 496) | The maximum number of tags to display. Set to 0 (zero) for unlimited. The tags that are displayed are always the top tags based on usage. For example, if the number is ten, the ten most used tags are displayed. When using the OrderBy (see page 497) property in conjunction with this property, it gets the given number of tags and then sorts them alphabetically. |

| | | |
|---|---|---|
| | OrderBy (see page 497) | The criteria by which the tags are ordered. Choices are:<br>• **TaggedCount** - sort by how many times the tags has been assigned.<br>• **Text** - sort the list in order A to Z. |
| | OrderByDirection (see page 497) | Select the direction of the TagOrderBy property. Choose Ascending or Descending. |
| | Stylesheet (see page 498) | Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet. |
| | TagTemplate (see page 498) | Add a path to another Web form to create links for the tag text. This path can be relative or absolute. There are five parameters that are automatically added to the link's QueryString that allow you to pass information about tag.<br>• **searchtag** or **searchgrptag** - searchtag represents the tag's text for users and searchgrptag represent the tag's text for community groups. When the template contains a CommunitySearch (see page 127) server control, the control sees these parameters and performs a search based on the text.<br>• **TagId** - represents the tag's ID.<br>• **TagLanguage** - represent the tags language.<br>• **TagCount** - represent the tag's count.... more (see page 498) |
| | TagTemplateTarget (see page 499) | Determines the type of window that appears on this Web form when a user clicks an item in the TagCloud (see page 493) control. The default is _self. |
| | TagType (see page 499) | Select whether the control displays tags for a user or a community group. Choices are:<br>• **User (see page 913)** - show tags that are assigned to users.<br>• **CommunityGroup (see page 719)** - show tags that are assigned to community group.<br>• **Content (see page 788)** - show tags that are assigned to content.<br>• **Library (see page 664)** - show tags that are assigned to library items. |
| | XmlDoc (see page 499) | Returns the Xslt object for the collection. |

### TagCloud.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

### TagCloud.DisplayXslt Property

Determines how information is displayed on the page. Enter the path to the XSL file. It can be relative or absolute.(optional).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. (optional)")]
[TypeConverter(typeof(XsltFavoriteTypeConverter))]
[DefaultValue("Builtin")]
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Determines how information is displayed on the page. Enter the path to the XSL
file. It can be relative or absolute. (optional)")>
<TypeConverter(typeof(XsltFavoriteTypeConverter))>
<DefaultValue("Builtin")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### TagCloud.DynamicParameter Property

**C#**

```
[Category("Ektron")]
[TypeConverter(typeof(string))]
[Description("")]
[DefaultValue("id")]
public string DynamicParameter;
```

**Visual Basic**

```
<Category("Ektron")>
<TypeConverter(typeof(string))>
<Description("")>
<DefaultValue("id")>
Public Property DynamicParameter() As string
```

### TagCloud.MarkupLanguage Property

Identify the template markup file that controls the display of the server control.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue(""), Description("Identify the template
markup file that controls the display of the server control.")]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue(""), Description("Identify the template
markup file that controls the display of the server control.")>
Public Property MarkupLanguage() As string
```

### TagCloud.MaxTagsDisplayed Property

The maximum number of tags to display. Set to 0 (zero) for unlimited. The tags that are displayed are always the top tags based on usage. For example, if the number is ten, the ten most used tags are displayed. When using the OrderBy (see page 497) property in conjunction with this property, it gets the given number of tags and then sorts them alphabetically.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The maximum number of tags to display. Set to 0 (zero) for unlimited. The tags
that are displayed are always the top tags based on usage. For example, if the number is ten,
the ten most used tags are displayed. When using the OrderBy property in conjunction with this
property, it gets the given number of tags and then sorts them alphabetically.")]
[DefaultValue("50")]
public int MaxTagsDisplayed;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The maximum number of tags to display. Set to 0 (zero) for unlimited. The tags
that are displayed are always the top tags based on usage. For example, if the number is ten,
the ten most used tags are displayed. When using the OrderBy property in conjunction with this
property, it gets the given number of tags and then sorts them alphabetically.")>
<DefaultValue("50")>
Public Property MaxTagsDisplayed() As Integer
```

### TagCloud.OrderBy Property

The criteria by which the tags are ordered. Choices are:

- **TaggedCount** - sort by how many times the tags has been assigned.
- **Text** - sort the list in order A to Z.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Text")]
[TypeConverter(typeof(TagOrderByTypeConverter))]
[Description("The criteria by which the tags are ordered. Choices are TaggedCount and Text")]
public string OrderBy;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Text")>
<TypeConverter(typeof(TagOrderByTypeConverter))>
<Description("The criteria by which the tags are ordered. Choices are TaggedCount and Text")>
Public Property OrderBy() As string
```

### TagCloud.OrderByDirection Property

Select the direction of the TagOrderBy property. Choose Ascending or Descending.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select the direction of the TagOrderBy property. Choose Ascending or
Descending.")]
[TypeConverter(typeof(OrderByDirectionTypeConverter))]
[DefaultValue("Ascending")]
public string OrderByDirection;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select the direction of the TagOrderBy property. Choose Ascending or
Descending.")>
<TypeConverter(typeof(OrderByDirectionTypeConverter))>
```

```
<DefaultValue("Ascending")>
Public Property OrderByDirection() As string
```

### TagCloud.Stylesheet Property

Specify the path to a style sheet for use with the sever control. The location can be relative or absolute. Leave blank to use the default style sheet.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Specify the path to a style sheet for use with the sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Specify the path to a style sheet for use with the sever control. The location
can be relative or absolute. Leave blank to use the default style sheet.")>
<Personalizable()>
Public Property Stylesheet() As string
```

### TagCloud.TagTemplate Property

Add a path to another Web form to create links for the tag text. This path can be relative or absolute. There are five parameters that are automatically added to the link's QueryString that allow you to pass information about tag.

- **searchtag** or **searchgrptag** - searchtag represents the tag's text for users and searchgrptag represent the tag's text for community groups. When the template contains a CommunitySearch (see page 127) server control, the control sees these parameters and performs a search based on the text.
- **TagId** - represents the tag's ID.
- **TagLanguage** - represent the tags language.
- **TagCount** - represent the tag's count. The tag's count is the amount of times a tag has been used.
- **TagType (see page 499)** - represents the tag's type. The tag's type will be user or community group.

In addition to these parameters, you can add your own by defining them in the path. When you do, these parameters will be appended to your parameters.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(""), Description("Add a path to another Web
form to create links for the tag text. This path can be relative or absolute. There are five
parameters that are automatically added to the link?s QueryString that allow you to pass
information about tag. They are searchtag or searchgrptag depending whether the control is set
to user or group and TagId, TagLanguage, TagCount and TagType.") ]
public string TagTemplate;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(""), Description("Add a path to another Web
form to create links for the tag text. This path can be relative or absolute. There are five
parameters that are automatically added to the link?s QueryString that allow you to pass
information about tag. They are searchtag or searchgrptag depending whether the control is set
to user or group and TagId, TagLanguage, TagCount and TagType.") >
Public Property TagTemplate() As string
```

### TagCloud.TagTemplateTarget Property

Determines the type of window that appears on this Web form when a user clicks an item in the TagCloud (see page 493) control. The default is _self.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self),
Description("Determines the type of window that appears on this Web form when a user clicks an
item in the TagCloud control. The default is _self.") ]
public ItemLinkTargets TagTemplateTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self),
Description("Determines the type of window that appears on this Web form when a user clicks an
item in the TagCloud control. The default is _self.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

### TagCloud.TagType Property

Select whether the control displays tags for a user or a community group. Choices are:

- **User (see page 913)** - show tags that are assigned to users.
- **CommunityGroup (see page 719)** - show tags that are assigned to community group.
- **Content (see page 788)** - show tags that are assigned to content.
- **Library (see page 664)** - show tags that are assigned to library items.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Select whether the control displays tags for a user, community group, content,
or library item. Choices are User, CommunityGroup, Content, and Library.")]
[TypeConverter(typeof(TagTypeTypeConverter))]
[DefaultValue("User")]
public string TagType;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Select whether the control displays tags for a user, community group, content,
or library item. Choices are User, CommunityGroup, Content, and Library.")>
<TypeConverter(typeof(TagTypeTypeConverter))>
<DefaultValue("User")>
Public Property TagType() As string
```

### TagCloud.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## UserProfile Class

**Class Hierarchy**

**C#**

```
[DefaultProperty("DefaultUserID"), ToolboxData("<{0}:UserProfile
runat=server></{0}:UserProfile>")]
public class UserProfile : EkXsltWebPart;
```

**Visual Basic**

```
<DefaultProperty("DefaultUserID"), ToolboxData("<{0}:UserProfile
runat=server></{0}:UserProfile>")>
Public Class UserProfile
Inherits EkXsltWebPart
```

**File**

Profile.cs

## UserProfile Methods

The methods of the UserProfile class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Fill (see page 500) | Method that can be overridden and used to grab data from the back end for each server control. |
| | ToString (see page 500) | Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in. |
| | Wrap (see page 500) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

## UserProfile.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

## UserProfile.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user is logged in.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

## UserProfile.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

**C#**

```
public override string Wrap(string Html);
```

**Visual Basic**

```
Public override Function Wrap(Html As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar. |

### Returns

## UserProfile Properties

The properties of the UserProfile class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AvatarHeight (see page 501) | Gets or Sets the avatar height. Use 0 for the actual image dimensions. |
| | AvatarWidth (see page 501) | Gets or Sets the avatar width. Use 0 for the actual image dimensions. |
| | CacheInterval (see page 502) | Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred). |
| | DefaultUserID (see page 502) | Gets or Sets the default user ID for this control. This is the user ID that will be shown if the DynamicParameter (see page 503) does not get a value. (e.g. userprofile.aspx?userid=2 will get the value 2 for a DynamicParameter (see page 503) of "id", and will display DefaultUserID otherwise.) |
| | DisplayXslt (see page 502) | The Xslt to use to render the control (optional). |
| | DynamicParameter (see page 503) | To make this control dynamic, select ID. When you do, this server control is attached to the user passed as a URL parameter. |
| | EnableEmailDisplay (see page 503) | Enables or disables showing the users email in the profile information. |
| | Stylesheet (see page 503) | Specifies the Css to which this control will provide a link.Select either Use Default, None, or enter a path to a custom Css file. |
| | TagTemplate (see page 504) | Gets or Sets the template-url for the personal tag links. |
| | TagTemplateTarget (see page 504) | Gets or Sets the target (window) attribute for the personal-tag links. |
| | TaxonomyId (see page 504) | Enter the ID number of the taxonomy or category to appear in this server control. Root is 0. |
| | XmlDoc (see page 505) | Returns the Xslt object for the collection. |

## UserProfile.AvatarHeight Property

Gets or Sets the avatar height. Use 0 for the actual image dimensions.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(74), Description("Gets or Sets the avatar
height. Use 0 for the actual image dimensions.") ]
public int AvatarHeight;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(74), Description("Gets or Sets the avatar
height. Use 0 for the actual image dimensions.") >
Public Property AvatarHeight() As Integer
```

## UserProfile.AvatarWidth Property

Gets or Sets the avatar width. Use 0 for the actual image dimensions.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(74), Description("Gets or Sets the avatar
width. Use 0 for the actual image dimensions.") ]
public int AvatarWidth;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(74), Description("Gets or Sets the avatar
width. Use 0 for the actual image dimensions.") >
Public Property AvatarWidth() As Integer
```

### UserProfile.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")]
[DefaultValue(0)]
[Personalizable()]
public virtual double CacheInterval;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("Sets the amount of time the server control?s data is cached. The default is 0
(zero). This is the amount of time, in seconds, a control?s data is cached. For example, if
you want to cache the data for five minutes, you set the CacheInterval property to 300 (three
hundred).")>
<DefaultValue(0)>
<Personalizable()>
Public virtual Property CacheInterval() As double
```

### UserProfile.DefaultUserID Property

Gets or Sets the default user ID for this control. This is the user ID that will be shown if the DynamicParameter (see page 503) does not get a value. (e.g. userprofile.aspx?userid=2 will get the value 2 for a DynamicParameter (see page 503) of "id", and will display DefaultUserID otherwise.)

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Gets or sets the default user ID for this control. This is the user ID that will
be shown if the DynamicParameter does not get a value. (e.g. userprofile.aspx?userid=2 will
get the value 2 for a DynamicParameter of \"id\", and will use DefaultUserID otherwise.)")]
public long DefaultUserID;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Gets or sets the default user ID for this control. This is the user ID that will
be shown if the DynamicParameter does not get a value. (e.g. userprofile.aspx?userid=2 will
get the value 2 for a DynamicParameter of \"id\", and will use DefaultUserID otherwise.)")>
Public Property DefaultUserID() As long
```

### UserProfile.DisplayXslt Property

The Xslt to use to render the control (optional).

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the control (optional)")]
[TypeConverter(typeof(XsltProfileTypeConverter))]
[DefaultValue("ProfileBasic.xsl")]
```

```
[Personalizable()]
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the control (optional)")>
<TypeConverter(typeof(XsltProfileTypeConverter))>
<DefaultValue("ProfileBasic.xsl")>
<Personalizable()>
Public override Property DisplayXslt() As string
```

### UserProfile.DynamicParameter Property

To make this control dynamic, select ID. When you do, this server control is attached to the user passed as a URL parameter.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[Description("To make this control dynamic, select ID. When you do, this server control is
attached to the user passed as a URL parameter.")]
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[DefaultValue("None - Use Default")]
public string DynamicParameter;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<Description("To make this control dynamic, select ID. When you do, this server control is
attached to the user passed as a URL parameter.")>
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<DefaultValue("None - Use Default")>
Public Property DynamicParameter() As string
```

### UserProfile.EnableEmailDisplay Property

Enables or disables showing the users email in the profile information.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("false")]
[TypeConverter(typeof(bool))]
[Description("Set to true to display the users email address in the profile info, set to false
to hide this (default)")]
[Personalizable()]
public bool EnableEmailDisplay;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("false")>
<TypeConverter(typeof(bool))>
<Description("Set to true to display the users email address in the profile info, set to false
to hide this (default)")>
<Personalizable()>
Public Property EnableEmailDisplay() As bool
```

### UserProfile.Stylesheet Property

Specifies the Css to which this control will provide a link.Select either Use Default, None, or enter a path to a custom Css file.

**C#**

```
[Bindable(true)]
```

```
[Category("Ektron")]
[DefaultValue("Use Default")]
[TypeConverter(typeof(CssTypeConverter))]
[Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")]
[Personalizable()]
public string Stylesheet;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Use Default")>
<TypeConverter(typeof(CssTypeConverter))>
<Description("Sets the Css for the control.  Select either 'Use Default', 'None', or enter a
path to a custom Css file.")>
<Personalizable()>
Public Property Stylesheet() As string
```

### UserProfile.TagTemplate Property

Gets or Sets the template-url for the personal tag links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a user
search, when a tag is clicked (example: UserSearch.aspx).") ]
public string TagTemplate;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(""),
TypeConverter(typeof(FavoriteLinkConverter)), Description("The template to open and run a user
search, when a tag is clicked (example: UserSearch.aspx).") >
Public Property TagTemplate() As string
```

### UserProfile.TagTemplateTarget Property

Gets or Sets the target (window) attribute for the personal-tag links.

**C#**

```
[ Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the personal tag links (_blank, _self, _parent or _top), to select opening
in the same window, a popup, etc.") ]
public ItemLinkTargets TagTemplateTarget;
```

**Visual Basic**

```
< Bindable(true), Category("Ektron"), DefaultValue(ItemLinkTargets._self), Description("The
target attribute of the personal tag links (_blank, _self, _parent or _top), to select opening
in the same window, a popup, etc.") >
Public Property TagTemplateTarget() As ItemLinkTargets
```

### UserProfile.TaxonomyId Property

Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.

**C#**

```
[Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)]
[Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")]
[Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public long TaxonomyId;
```

**Visual Basic**

```
<Bindable(true), Category("Ektron"), DefaultValue("0"), Browsable(true)>
```

```
<Description("Enter the ID number of the taxonomy or category to appear in this server
control. Root is 0.")>
<Editor(typeof(TaxonomyIdUiTypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property TaxonomyId() As long
```

### UserProfile.XmlDoc Property

Returns the Xslt object for the collection.

**C#**

```
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
Public override Property XmlDoc() As XmlDocument
```

## WebSearch Class

The Web Search (see page 455) server control lets you customize the behavior of the search. You place this control on any Web form from which a site visitor can search your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 694)" > "Web Site (see page 694) Search (see page 455)."

**Class Hierarchy**

**C#**

```
[DefaultProperty("FolderID"), ToolboxData("<{0}:WebSearch runat=server></{0}:WebSearch>")]
public class WebSearch : EkXsltBase, ICallbackEventHandler;
```

**Visual Basic**

```
<DefaultProperty("FolderID"), ToolboxData("<{0}:WebSearch runat=server></{0}:WebSearch>")>
Public Class WebSearch
Inherits EkXsltBase
Inherits ICallbackEventHandler
```

**File**

WebSearch.cs

### WebSearch.WebSearch Constructor

Constructor for the WebSearch class.

**C#**

```
public WebSearch();
```

**Visual Basic**

```
Public Sub New()
```

### WebSearch Methods

The methods of the WebSearch class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Fill (see page 506) | Method that can be overridden and used to grab data from the back end for each server control. |
| | GetCallbackResult (see page 506) | Returns search results to the client callback function |

| | | |
|---|---|---|
| | LoadSearch (see page 506) | Search (see page 455) HTML, content, images, multimedia and documents based on content or metadata. This method can be used to perform programmatic search. |
| | RaiseCallbackEvent (see page 506) | Processes the callback event by performing the search. |
| | Wrap (see page 507) | Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text. |

### WebSearch.Fill Method

Method that can be overridden and used to grab data from the back end for each server control.

**C#**

```
public override void Fill();
```

**Visual Basic**

```
Public override Function Fill() As void
```

### WebSearch.GetCallbackResult Method

Returns search results to the client callback function

**C#**

```
public string GetCallbackResult();
```

**Visual Basic**

```
Public Function GetCallbackResult() As string
```

**Returns**

String containing search results.

### WebSearch.LoadSearch Method

Search (see page 455) HTML, content, images, multimedia and documents based on content or metadata. This method can be used to perform programmatic search.

**C#**

```
public int LoadSearch(SearchRequestData searchReqData);
```

**Visual Basic**

```
Public Function LoadSearch(searchReqData As SearchRequestData) As Integer
```

**Parameters**

| Parameters | Description |
|---|---|
| searchReqData | Search (see page 455) criteria bundled in a SearchRequestData object. |

**Returns**

Number of records satisfying search criteria

### WebSearch.RaiseCallbackEvent Method

Processes the callback event by performing the search.

**C#**

```
public void RaiseCallbackEvent(string eventArgs);
```

**Visual Basic**

```
Public Function RaiseCallbackEvent(eventArgs As string) As void
```

### Parameters

| Parameters | Description |
|---|---|
| eventArgs | Name-value pairs representing form fields passed by client callback function |

## WebSearch.Wrap Method

Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

### C#

```
public override string Wrap(string Html);
```

### Visual Basic

```
Public override Function Wrap(Html As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| Html | Html string to be wrapped with a toolbar |

### Returns

Html string wrapped with a toolbar

## WebSearch Properties

The properties of the WebSearch class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | ButtonImgSrc (see page 510) | Optional: If you want to display an image on the submit button, enter the server path to it. |
| | ButtonText (see page 510) | The text used for the button if no image source is identified. If an image source is identified, this is alternative text for the button. |
| | CustomOrderBy (see page 510) | Order search results by a custom property. Set this property to the friendly name of the property defined in the Indexing Service. If an invalid property is specified, no search results are returned. If this is left empty, the OrderBy (see page 515) property is used to order search results. |
| | CustomSearch (see page 511) | Optional: If you want to display an image on the submit button, enter the server path to it. |
| | DisableForumSearch (see page 511) | Enable Forum (see page 256) Search (see page 455). |
| | DisplayXslt (see page 511) | Determines the display of the search results page.<br>• **None** - databind only<br>• **ecmNavigation** - lists the title of every content item found by the search<br>• **ecmTeaser** - lists the title and the summary of every content item found by the search<br>• **ecmUnOrderedList** - sorts the list in no particular order. Shows the title and content summary<br>• **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page |
| | DynamicContentTemplate (see page 512) | Get or sets the template of dynamic content. If specified control will not use dynamic box. |
| | EnableAdvancedLink (see page 512) | Set to **True** to display the link that makes the advanced properties visible. |
| | EndCount (see page 512) | Get the numerical record of the first item on a page. |
| | FolderID (see page 512) | The folder at which the search begins. The starting folder need not be the root folder. The **Recursive (see page 516)** property determines if the search examines this folder's subfolders. |

| | | |
|---|---|---|
| | IsInWorkArea (see page 513) | Is server control in workarea |
| | LinkTarget (see page 513) | Gets or sets the target window for the hyperlinks in the search result. The choices are:<br>• **_blank** - loads the link in a new blank window. This window is not named.<br>• **_self** - loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.<br>• **_parent** - loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.<br>• **_top** - loads the link in the full body of the window. This defaults to acting like "_self" if the... more (see page 513) |
| | MarkupLanguage (see page 514) | Identify the template markup file that controls the display of the search results. For example, `mysearchmarkup.ekml`.<br>Enter the path relative to site root. For example, `\workarea\customfiles\markup\mysearchmarkup.ekml`.<br>If you enter a valid EkML file, the DisplayXslt (see page 511) property value is ignored. |
| | MaxCharacters (see page 514) | The maximum number of characters the Search (see page 455) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 523) property to the same number. |
| | MaxTeaserLength (see page 514) | An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT). |
| | MultipleFolders (see page 515) | Optional: If you want to display an image on the submit button, enter the server path to it. |
| | OrderBy (see page 515) | The order of the search results. Choices are:<br>• **Title** - The content title (alphabetical)<br>• **ID** - The content ID number<br>• **Date Created** - The date the content was created<br>• **Date Modified** - The date the content was most recently modified<br>• **Editor** - The user who last edited the content (alphabetical)<br>• **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 694)" > "Display of Search (see page 455) Results" > "Search (see page 455) Result Ranking" |
| | OrderDirection (see page 515) | Specify the order direction of the results. Choices are **Ascending** or **Descending**. |
| | Recursive (see page 516) | Select **True** if you wish to search sub-folders of the starting folder. The starting folder is identified in the `FolderID` (see page 512) property. |
| | RemoveTeaserHtml (see page 516) | An optional result formatting parameter. If set to **True**, any HTML in the content items' teaser/summary is removed before it is sent to the browser. If set to **False**, any existing HTML is passed along, (this may then be further filtered by the XSLT). |
| | ResultCount (see page 516) | The number of search results. This is a read-only property. |
| | ResultDataTable (see page 517) | Allows a developer to get results from the search control as a DataTable |
| | ResultsPageSize (see page 517) | An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT). |
| | ResultTagId (see page 518) | Lets you designate where search results appear. You can place search criteria in one area of a Web form and the results in another.<br>For example, you have the following tag.<br>`<span id="results"></span>`<br>In this case, enter **results** for the property value. |
| | ResultXml (see page 518) | Allows a developer to get results from the search control as XML. |

| | | |
|---|---|---|
| | SearchFor (see page 518) | Choose the type of content to search.<br>• All<br>• HTML<br>• Documents<br>• Images<br>• Multimedia<br>• Discussion Forums<br>• Tags (see page 771) |
| | SearchSynonyms (see page 519) | If set to **True**, the search results will search for synonyms configured for term. The default is True.<br>• **True** - Display search for synonym of term specified.<br>• **False** - Do not search synonyms. |
| | SearchText (see page 519) | Get or set default search text. This is optional. |
| | ShowCategories (see page 520) | If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False.<br>**True** = Search (see page 455) across taxonomy categories<br>**False** = Do not search across taxonomy categories |
| | ShowCustomSummary (see page 520) | If set to **True**, custom summary is used in search results. The default is False.<br>**True** = Display custom summary in search results.<br>**False** = Display characterization generated by indexing service in search results. |
| | ShowSearchBoxAlways (see page 520) | If set to **False**, the search box does not appear on PostBack. The default is True.<br>**True** = Display search box on PostBack<br>**False** = Do not display search box on PostBack |
| | ShowSearchOptions (see page 521) | If set to **True**, the following options appear above the Search (see page 455) box.<br>• Site (see page 694)<br>• Html<br>• Documents<br>• Images<br>• Multimedia<br>• Tags (see page 771) |
| | ShowSuggestedResults (see page 521) | If set to **True**, the search results include Suggested Results if any exist. The default is True.<br>• **True** = Display Suggested Results<br>• **False** = Do not display Suggested Results |
| | StartCount (see page 522) | Get the numerical record of the last item on a page. |
| | Stylesheet (see page 522) | Specify the location of the style sheet for the search results page. Set the location relative to the site root folder. For example: `csslib\mytest.css`.<br>Leave blank to use the default style sheet, `\webroot\Workarea\csslib\global.css`. |
| | TaxCategories (see page 522) | Is server control in workarea |
| | TaxonomyOperator (see page 522) | Taxonomy (see page 842) operator used in filtering search results by taxonomy category.<br>• and - selected taxonomy categories are AND-ed<br>• or - selected taxonomy categories are OR-ed |
| | TextBoxSize (see page 523) | The size of the Search (see page 455) text box for user input. The width is based on the number of characters. |
| | TimeTaken (see page 523) | Gets the amount of time, in seconds, it has taken to perform the search. |
| | XmlDoc (see page 523) | This property is not supported by WebSearch (see page 505). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 511) property of the WebSearch (see page 505) server control. The XSLT filename is specified relative to the current directory. |

### WebSearch.ButtonImgSrc Property

Optional: If you want to display an image on the submit button, enter the server path to it.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: The path to the image you wish to use for the button. This path is
assumed to be relative to workarea/images.")]
public string ButtonImgSrc;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: The path to the image you wish to use for the button. This path is
assumed to be relative to workarea/images.")>
Public Property ButtonImgSrc() As string
```

### WebSearch.ButtonText Property

The text used for the button if no image source is identified. If an image source is identified, this is alternative text for the button.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_SEARCH_BTN)]
public string ButtonText;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<Description("The text of the search button if no image is specified. The default value is set
to \"res_srch_btn\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_SEARCH_BTN)>
Public Property ButtonText() As string
```

### WebSearch.CustomOrderBy Property

Order search results by a custom property. Set this property to the friendly name of the property defined in the Indexing Service. If an invalid property is specified, no search results are returned. If this is left empty, the OrderBy (see page 515) property is used to order search results.

**C#**

```
[Category("Ektron"), DefaultValue(""), Description("Order search results by a custom property.
Set this property to the friendly name of the property.")]
public string CustomOrderBy;
```

**Visual Basic**

```
<Category("Ektron"), DefaultValue(""), Description("Order search results by a custom property.
Set this property to the friendly name of the property.")>
Public Property CustomOrderBy() As string
```

### WebSearch.CustomSearch Property

Optional: If you want to display an image on the submit button, enter the server path to it.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: Custom directories to search.")]
public string CustomSearch;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: Custom directories to search.")>
Public Property CustomSearch() As string
```

### WebSearch.DisableForumSearch Property

Enable Forum (see page 256) Search (see page 455).

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true Discussion Forums option does not appear in search.")]
[Personalizable(), WebBrowsable()]
public bool DisableForumSearch;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true Discussion Forums option does not appear in search.")>
<Personalizable(), WebBrowsable()>
Public Property DisableForumSearch() As bool
```

### WebSearch.DisplayXslt Property

Determines the display of the search results page.

- **None** - databind only
- **ecmNavigation** - lists the title of every content item found by the search
- **ecmTeaser** - lists the title and the summary of every content item found by the search
- **ecmUnOrderedList** - sorts the list in no particular order. Shows the title and content summary
- **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltTypeConverter))]
[DefaultValue("None - Databind Only")]
[Personalizable(), WebBrowsable()]
```

```
public override string DisplayXslt;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltTypeConverter))>
<DefaultValue("None - Databind Only")>
<Personalizable(), WebBrowsable()>
Public override Property DisplayXslt() As string
```

### WebSearch.DynamicContentTemplate Property

Get or sets the template of dynamic content. If specified control will not use dynamic box.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Get or sets the template of dynamic content. Overwrites the content quicklink.")]
public string DynamicContentTemplate;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Get or sets the template of dynamic content. Overwrites the content quicklink.")>
Public Property DynamicContentTemplate() As string
```

### WebSearch.EnableAdvancedLink Property

Set to **True** to display the link that makes the advanced properties visible.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to display the link that makes the advanced properties visible.")]
public bool EnableAdvancedLink;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to display the link that makes the advanced properties visible.")>
Public Property EnableAdvancedLink() As bool
```

### WebSearch.EndCount Property

Get the numerical record of the first item on a page.

**C#**

```
[Browsable(false)]
public int EndCount;
```

**Visual Basic**

```
<Browsable(false)>
Public Property EndCount() As Integer
```

### WebSearch.FolderID Property

The folder at which the search begins. The starting folder need not be the root folder. The **Recursive (see page 516)** property

determines if the search examines this folder's subfolders.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The ID of the folder you wish to select data from. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable(), WebBrowsable()]
public long FolderID;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable(), WebBrowsable()>
Public Property FolderID() As long
```

### WebSearch.IsInWorkArea Property

Is server control in workarea

**C#**

```
[Browsable(false)]
public bool IsInWorkArea;
```

**Visual Basic**

```
<Browsable(false)>
Public Property IsInWorkArea() As bool
```

### WebSearch.LinkTarget Property

Gets or sets the target window for the hyperlinks in the search result. The choices are:

- _**blank** - loads the link in a new blank window. This window is not named.
- _**self** - loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target.
- _**parent** - loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent.
- _**top** - loads the link in the full body of the window. This defaults to acting like "_self" if the document is already at the top. It is useful for breaking out of an arbitrarily deep frame nesting.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(ItemLinkTargets._self)]
[Description("The target attribute of the link.")]
public ItemLinkTargets LinkTarget;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(ItemLinkTargets._self)>
<Description("The target attribute of the link.")>
Public Property LinkTarget() As ItemLinkTargets
```

### WebSearch.MarkupLanguage Property

Identify the template markup file that controls the display of the search results. For example, `mysearchmarkup.ekml`.

Enter the path relative to site root. For example, `\workarea\customfiles\markup\mysearchmarkup.ekml`.

If you enter a valid EkML file, the DisplayXslt (see page 511) property value is ignored.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Enter name of a markup file that manages the display of this server control.")]
[Personalizable(), WebBrowsable()]
public string MarkupLanguage;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Enter name of a markup file that manages the display of this server control.")>
<Personalizable(), WebBrowsable()>
Public Property MarkupLanguage() As string
```

### WebSearch.MaxCharacters Property

The maximum number of characters the Search (see page 455) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 523) property to the same number.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(50)]
[Description("The maximum number of characters the user can enter in the search box.")]
public int MaxCharacters;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(50)>
<Description("The maximum number of characters the user can enter in the search box.")>
Public Property MaxCharacters() As Integer
```

### WebSearch.MaxTeaserLength Property

An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(256)]
[Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")]
public int MaxTeaserLength;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(256)>
<Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")>
Public Property MaxTeaserLength() As Integer
```

### WebSearch.MultipleFolders Property

Optional: If you want to display an image on the submit button, enter the server path to it.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: Multiple CMS Folders to search, seperate each folder with comma.")]
public string MultipleFolders;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: Multiple CMS Folders to search, seperate each folder with comma.")>
Public Property MultipleFolders() As string
```

### WebSearch.OrderBy Property

The order of the search results. Choices are:

- **Title** - The content title (alphabetical)
- **ID** - The content ID number
- **Date Created** - The date the content was created
- **Date Modified** - The date the content was most recently modified
- **Editor** - The user who last edited the content (alphabetical)
- **Rank** - The rank assigned to the content. See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 694)" > "Display of Search (see page 455) Results" > "Search (see page 455) Result Ranking"

**C#**

```
[Browsable(true)]
[Category("Ektron")]
[Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified")]
[Personalizable(), WebBrowsable()]
public WebSearchOrder OrderBy;
```

**Visual Basic**

```
<Browsable(true)>
<Category("Ektron")>
<Description("Specify the order of the results. Choices are: Rank, Title, User, DateCreated,
DateModified")>
<Personalizable(), WebBrowsable()>
Public Property OrderBy() As WebSearchOrder
```

### WebSearch.OrderDirection Property

Specify the order direction of the results. Choices are **Ascending** or **Descending**.

**C#**

```
[Browsable(true)]
[Category("Ektron")]
[TypeConverter(typeof(SortOrderTypeConverter))]
[Description("Specify the order direction of the results.")]
[Personalizable(), WebBrowsable()]
public string OrderDirection;
```

**Visual Basic**

```
<Browsable(true)>
<Category("Ektron")>
<TypeConverter(typeof(SortOrderTypeConverter))>
<Description("Specify the order direction of the results.")>
<Personalizable(), WebBrowsable()>
Public Property OrderDirection() As string
```

### WebSearch.Recursive Property

Select **True** if you wish to search sub-folders of the starting folder. The starting folder is identified in the `FolderID` (see page 512) property.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Select true if you wish to search the subfolders of the selected Folder")]
public bool Recursive;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Select true if you wish to search the subfolders of the selected Folder")>
Public Property Recursive() As bool
```

### WebSearch.RemoveTeaserHtml Property

An optional result formatting parameter. If set to **True**, any HTML in the content items' teaser/summary is removed before it is sent to the browser. If set to **False**, any existing HTML is passed along, (this may then be further filtered by the XSLT).

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("An optional search-result formatting parameter that removes all HTML from each
content-items' teaser/summary before sending it to the browser.")]
public bool RemoveTeaserHtml;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("An optional search-result formatting parameter that removes all HTML from each
content-items' teaser/summary before sending it to the browser.")>
Public Property RemoveTeaserHtml() As bool
```

### WebSearch.ResultCount Property

The number of search results. This is a read-only property.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The number of search results. This is a read-only property.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int ResultCount;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The number of search results. This is a read-only property.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property ResultCount() As Integer
```

### WebSearch.ResultDataTable Property

Allows a developer to get results from the search control as a DataTable

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to get results from the search control as a DataTable.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual DataTable ResultDataTable;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to get results from the search control as a DataTable.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property ResultDataTable() As DataTable
```

### WebSearch.ResultsPageSize Property

An optional result formatting parameter. If non-zero, then the matching content items' teaser/summary is truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("An optional page size setting, defaults to web.config search provider page
size).")]
[Personalizable(), WebBrowsable()]
public int ResultsPageSize;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("An optional page size setting, defaults to web.config search provider page
```

```
size).")>
<Personalizable(), WebBrowsable()>
Public Property ResultsPageSize() As Integer
```

### WebSearch.ResultTagId Property

Lets you designate where search results appear. You can place search criteria in one area of a Web form and the results in another.

For example, you have the following tag.

```
<span id="results"></span>
```

In this case, enter **results** for the property value.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("an optional parameter to specify the div control id to display the search
result")]
public string ResultTagId;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("an optional parameter to specify the div control id to display the search
result")>
Public Property ResultTagId() As string
```

### WebSearch.ResultXml Property

Allows a developer to get results from the search control as XML.

**C#**

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to get results from the search control as XML.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual string ResultXml;
```

**Visual Basic**

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to get results from the search control as XML.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property ResultXml() As string
```

### WebSearch.SearchFor Property

Choose the type of content to search.

- All
- HTML
- Documents
- Images

- Multimedia
- Discussion Forums
- Tags (see page 771)

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Choose the type of content to search. All - HTML - Documents - Images -
Multimedia - Discussion Forums - Tags")]
[Personalizable(), WebBrowsable()]
public SearchDocumentType SearchFor;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Choose the type of content to search. All - HTML - Documents - Images -
Multimedia - Discussion Forums - Tags")>
<Personalizable(), WebBrowsable()>
Public Property SearchFor() As SearchDocumentType
```

### WebSearch.SearchSynonyms Property

If set to **True**, the search results will search for synonyms configured for term. The default is True.

- **True** - Display search for synonym of term specified.
- **False** - Do not search synonyms.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to true, Synonyms configured will be used for search.")]
[Personalizable(), WebBrowsable()]
public bool SearchSynonyms;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to true, Synonyms configured will be used for search.")>
<Personalizable(), WebBrowsable()>
Public Property SearchSynonyms() As bool
```

### WebSearch.SearchText Property

Get or set default search text. This is optional.

**C#**

```
[Browsable(false)]
public string SearchText;
```

**Visual Basic**

```
<Browsable(false)>
Public Property SearchText() As string
```

### WebSearch.ShowCategories Property

If set to **True**, taxonomy categories will also be searched for the search text specified. The default is False.

**True** = Search (see page 455) across taxonomy categories

**False** = Do not search across taxonomy categories

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")]
public bool ShowCategories;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, taxonomy categories will also be searched for
the search text specified.")>
Public Property ShowCategories() As bool
```

### WebSearch.ShowCustomSummary Property

If set to **True**, custom summary is used in search results. The default is False.

**True** = Display custom summary in search results.

**False** = Display characterization generated by indexing service in search results.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If this property is set to true, custom summary is displayed in search
results.")]
public bool ShowCustomSummary;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If this property is set to true, custom summary is displayed in search
results.")>
Public Property ShowCustomSummary() As bool
```

### WebSearch.ShowSearchBoxAlways Property

If set to **False**, the search box does not appear on PostBack. The default is True.

**True** = Display search box on PostBack

**False** = Do not display search box on PostBack

**C#**

```
[Bindable(true)]
[Browsable(true)]
```

```
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to false, the search box will not be visible on
PostBack.")]
public bool ShowSearchBoxAlways;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to false, the search box will not be visible on
PostBack.")>
Public Property ShowSearchBoxAlways() As bool
```

### WebSearch.ShowSearchOptions Property

If set to **True**, the following options appear above the Search (see page 455) box.

- Site (see page 694)
- Html
- Documents
- Images
- Multimedia
- Tags (see page 771)

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set to true to show these options above the Search box: Site, Html, Documents,
Images and Multimeadia.")]
public bool ShowSearchOptions;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Set to true to show these options above the Search box: Site, Html, Documents,
Images and Multimeadia.")>
Public Property ShowSearchOptions() As bool
```

### WebSearch.ShowSuggestedResults Property

If set to **True**, the search results include Suggested Results if any exist. The default is True.

- **True** = Display Suggested Results
- **False** = Do not display Suggested Results

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to true, Suggested Results will be shown if any exist
for the search terms the user entered.")]
[Personalizable(), WebBrowsable()]
public bool ShowSuggestedResults;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to true, Suggested Results will be shown if any exist
for the search terms the user entered.")>
<Personalizable(), WebBrowsable()>
Public Property ShowSuggestedResults() As bool
```

### WebSearch.StartCount Property

Get the numerical record of the last item on a page.

**C#**

```
[Browsable(false)]
public int StartCount;
```

**Visual Basic**

```
<Browsable(false)>
Public Property StartCount() As Integer
```

### WebSearch.Stylesheet Property

Specify the location of the style sheet for the search results page. Set the location relative to the site root folder. For example: `csslib\mytest.css`.

Leave blank to use the default style sheet, `\webroot\Workarea\csslib\global.css`.

**C#**

```
[Description("Specify the style sheet relative to site root. Leave blank for default.")]
[Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(StringConverter))]
public string Stylesheet;
```

**Visual Basic**

```
<Description("Specify the style sheet relative to site root. Leave blank for default.")>
<Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(StringConverter))>
Public Property Stylesheet() As string
```

### WebSearch.TaxCategories Property

Is server control in workarea

**C#**

```
[Browsable(false)]
public string TaxCategories;
```

**Visual Basic**

```
<Browsable(false)>
Public Property TaxCategories() As string
```

### WebSearch.TaxonomyOperator Property

Taxonomy (see page 842) operator used in filtering search results by taxonomy category.

- and - selected taxonomy categories are AND-ed

- or - selected taxonomy categories are OR-ed

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Taxonomy operator used in filtering search results by taxonomy category.")]
[Personalizable(), WebBrowsable()]
public TaxCategoryOperator TaxonomyOperator;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Taxonomy operator used in filtering search results by taxonomy category.")>
<Personalizable(), WebBrowsable()>
Public Property TaxonomyOperator() As TaxCategoryOperator
```

### WebSearch.TextBoxSize Property

The size of the Search (see page 455) text box for user input. The width is based on the number of characters.

**C#**

```
[Bindable(true)]
[Browsable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The width of the search box, in characters.")]
public int TextBoxSize;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The width of the search box, in characters.")>
Public Property TextBoxSize() As Integer
```

### WebSearch.TimeTaken Property

Gets the amount of time, in seconds, it has taken to perform the search.

**C#**

```
[Browsable(false)]
public double TimeTaken;
```

**Visual Basic**

```
<Browsable(false)>
Public Property TimeTaken() As double
```

### WebSearch.XmlDoc Property

This property is not supported by WebSearch (see page 505). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 511) property of the WebSearch (see page 505) server control. The XSLT filename is specified relative to the current directory.

**C#**

```
[Bindable(true)]
[Browsable(false)]
[Description("This property is not supported by WebSearch. To retrieve the XML structure, use
the DisplayXslt property in the WebSearch server control to get search results formatted using
```

```
an XSLT. You need to either specify an XSLT name or use the default XSLTs (ecmNavigation,
ecmTeaser) provided with the server control.")]
public override XmlDocument XmlDoc;
```

**Visual Basic**

```
<Bindable(true)>
<Browsable(false)>
<Description("This property is not supported by WebSearch. To retrieve the XML structure, use
the DisplayXslt property in the WebSearch server control to get search results formatted using
an XSLT. You need to either specify an XSLT name or use the default XSLTs (ecmNavigation,
ecmTeaser) provided with the server control.")>
Public override Property XmlDoc() As XmlDocument
```

# Plug-in Extension API

The Plug-in Extension API documentation details the methods and properties that are available when creating a custom plug-in.

Plug-in Extensions expose event hooks in Ektron CMS400.NET. As a developer, you can utilize these event hooks in Visual Studio C# or VB.NET to create customized events for your site. For instance, you could create a Plug-in Extension that:

- sends an instant message when content is published
- sends an email when content is published
- filters out objectionable language from blog posts
- appends HTML content when it's published

For additional information on Plug-in Extensions, see the Developer Manual section "Plug-in Extensions"

**Namespaces**

| Name | Description |
|---|---|
| Extensibility (see page 524) | This is namespace Ektron.Cms.Extensibility. |

# Ektron.Cms.Extensibility Namespace

This is namespace Ektron.Cms.Extensibility.

**Classes**

| | Name | Description |
|---|---|---|
| | ActionRecipientInfo (see page 526) | Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | ActionRequestInfo (see page 527) | Class which holds information about the Invitation being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | ContentInfo (see page 530) | Class which holds information about the content being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: long = -1. string = "". bool = false. |
| | ExtensionAuthor (see page 533) | Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in. |
| | ExtensionDescription (see page 534) | Custom attribute - Used at the method level to describe what a particular event does |
| | ExtensionEvent (see page 535) | Override the methods in this class to customize the CMS via event hooks. |

| | | |
|---|---|---|
| | FlagInfo (see page 553) | Class which holds information about the flagged content. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | FolderInfo (see page 557) | 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 559) | Class that describes a field in a form - name / value pair |
| | FormInfo (see page 560) | Class which describes a Form (see page 836) object |
| | MessageBoardInfo (see page 562) | Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | RequestInfo (see page 563) | Reserved for future use. |
| | ReviewInfo (see page 563) | Class which holds information about the review being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | SiteInfo (see page 567) | Class which holds information about the site in which the event was fired. |
| | UserInfo (see page 568) | Class which holds information about the user being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |

# Classes

The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | ActionRecipientInfo (see page 526) | Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | ActionRequestInfo (see page 527) | Class which holds information about the Invitation being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | ContentInfo (see page 530) | Class which holds information about the content being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: long = -1. string = "". bool = false. |
| | ExtensionAuthor (see page 533) | Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in. |
| | ExtensionDescription (see page 534) | Custom attribute - Used at the method level to describe what a particular event does |
| | ExtensionEvent (see page 535) | Override the methods in this class to customize the CMS via event hooks. |
| | FlagInfo (see page 553) | Class which holds information about the flagged content. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | FolderInfo (see page 557) | 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 559) | Class that describes a field in a form - name / value pair |
| | FormInfo (see page 560) | Class which describes a Form (see page 836) object |
| | MessageBoardInfo (see page 562) | Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | RequestInfo (see page 563) | Reserved for future use. |
| | ReviewInfo (see page 563) | Class which holds information about the review being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |
| | SiteInfo (see page 567) | Class which holds information about the site in which the event was fired. |
| | UserInfo (see page 568) | Class which holds information about the user being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false. |

# ActionRecipientInfo Class

Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

### Class Hierarchy

### C#

```
public sealed class ActionRecipientInfo;
```

### Visual Basic

```
Public sealed Class ActionRecipientInfo
```

### File

CmsInfoObjects.cs

## ActionRecipientInfo.ActionRecipientInfo Constructor

Constructor

### C#

```
public ActionRecipientInfo(XmlNode xml);
```

### Visual Basic

```
Public Sub New(xml As XmlNode)
```

### Parameters

| Parameters | Description |
|---|---|
| xml | node containing Recipient info. |

## ActionRecipientInfo Properties

The properties of the ActionRecipientInfo class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DisplayName (see page 526) | Display Name of the recipient. |
| | Email (see page 527) | Email address of the recipient |
| | FirstName (see page 527) | First Name of the recipient. |
| | IsUser (see page 527) | Returns true if the recipient is a validated user. This call is only valid after the Invitation has been Validated; otherwise false is returned regardless. |
| | LastName (see page 527) | Last Name of the recipient. |
| | UserId (see page 527) | a numeric value representing the CMS User (see page 913) ID of the recipient. |

### ActionRecipientInfo.DisplayName Property

Display Name of the recipient.

### C#

```
public string DisplayName;
```

### Visual Basic

```
Public Property DisplayName() As string
```

### ActionRecipientInfo.Email Property

Email address of the recipient

**C#**

```
public string Email;
```

**Visual Basic**

```
Public Property Email() As string
```

### ActionRecipientInfo.FirstName Property

First Name of the recipient.

**C#**

```
public string FirstName;
```

**Visual Basic**

```
Public Property FirstName() As string
```

### ActionRecipientInfo.IsUser Property

Returns true if the recipient is a validated user. This call is only valid after the Invitation has been Validated; otherwise false is returned regardless.

**C#**

```
public bool IsUser;
```

**Visual Basic**

```
Public Property IsUser() As bool
```

### ActionRecipientInfo.LastName Property

Last Name of the recipient.

**C#**

```
public string LastName;
```

**Visual Basic**

```
Public Property LastName() As string
```

### ActionRecipientInfo.UserId Property

a numeric value representing the CMS User (see page 913) ID of the recipient.

**C#**

```
public long UserId;
```

**Visual Basic**

```
Public Property UserId() As long
```

## ActionRequestInfo Class

Class which holds information about the Invitation being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

**Class Hierarchy**

**C#**

```
public sealed class ActionRequestInfo;
```

**Visual Basic**

```
Public sealed Class ActionRequestInfo
```

**File**

CmsInfoObjects.cs

## ActionRequestInfo.ActionRequestInfo Constructor

**C#**

```
public ActionRequestInfo(XmlNode xml);
```

**Visual Basic**

```
Public Sub New(xml As XmlNode)
```

**Parameters**

| Parameters | Description |
|---|---|
| xml | |

## ActionRequestInfo Enumerations

The enumerations of the ActionRequestInfo class are listed here.

**Enumerations**

| | Name | Description |
|---|---|---|
| | ActionRequestType (see page 528) | Enumeration identifying the different types of ActionRequests. |

### Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestType Enumeration

Enumeration identifying the different types of ActionRequests.

**C#**

```
public enum ActionRequestType {
  Friend = 1,
  JoinGroup = 2
}
```

**Visual Basic**

```
Public Enum ActionRequestType
  Friend = 1
  JoinGroup = 2
End Enum
```

**File**

CmsInfoObjects.cs

**Members**

| Members | Description |
|---|---|
| Friend | Represents a friend invitation. |
| JoinGroup | Represents a request to join a prviate group. |

## ActionRequestInfo Properties

The properties of the ActionRequestInfo class are listed here.

Copyright (c) Ektron Inc. 2008

### Public Properties

| | Name | Description |
|---|---|---|
| | ErrorMessage (see page 529) | Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite) |
| | Id (see page 529) | A numeric value representing the ID of the invitation. Only populated after save. |
| | Recipients (see page 529) | Collection (see page 88) recipients for the invitation |
| | SenderId (see page 529) | A numeric value representing the user ID of the invitation sender. |
| | Type (see page 529) | User (see page 913) ID of the invitation sender. |

## ActionRequestInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

**C#**

```
public string ErrorMessage;
```

**Visual Basic**

```
Public Property ErrorMessage() As string
```

## ActionRequestInfo.Id Property

A numeric value representing the ID of the invitation. Only populated after save.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## ActionRequestInfo.Recipients Property

Collection (see page 88) recipients for the invitation

**C#**

```
public Collection<ActionRecipientInfo> Recipients;
```

**Visual Basic**

```
Public Property Recipients() As Collection<ActionRecipientInfo>
```

## ActionRequestInfo.SenderId Property

A numeric value representing the user ID of the invitation sender.

**C#**

```
public long SenderId;
```

**Visual Basic**

```
Public Property SenderId() As long
```

## ActionRequestInfo.Type Property

User (see page 913) ID of the invitation sender.

**C#**

```
public ActionRequestType Type;
```

**Visual Basic**

```
Public Property Type() As ActionRequestType
```

# ContentInfo Class

Class which holds information about the content being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: long = -1. string = "". bool = false.

### Class Hierarchy

### C#

```
public sealed class ContentInfo;
```

### Visual Basic

```
Public sealed Class ContentInfo
```

### File

CmsInfoObjects.cs

## ContentInfo Properties

The properties of the ContentInfo class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AssetID (see page 530) | Gets a string containing the asset ID (see page 532) in the form of a GUID if the content is an asset. (ReadOnly) |
| | Comments (see page 531) | Gets a string containing the comments a user has entered during their operation. (ReadOnly) |
| | ErrorMessage (see page 531) | Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite) |
| | FileAndPath (see page 531) | Gets a string containing the full physical path and file name on the server if content is an asset. (ReadOnly) |
| | FileExtension (see page 531) | Gets a string containing file extension if content is an asset. (ReadOnly) |
| | FileName (see page 531) | Gets a string containing file name if content is an asset. (ReadOnly) |
| | FolderID (see page 531) | Gets an long representing the folder ID (see page 532) which this content is located. (ReadOnly) |
| | Html (see page 532) | Gets or Sets a string containing all the html displayed to end users for a piece of content. (ReadWrite Note: This property only allows write access within OnBeforePublish.) |
| | ID (see page 532) | Gets an long representing the content ID of the content. (ReadOnly) |
| | Language (see page 532) | Gets an integer representing the language of the content. (ReadOnly) |
| | MimeType (see page 532) | Gets a string containing the MIME type if content is an asset. (ReadOnly) |
| | Status (see page 532) | Gets a string representing the status of the content, published, checkedOut. (ReadOnly) |
| | Title (see page 532) | Gets a string containing the title of the content. (ReadOnly) |
| | Type (see page 533) | Gets an integer representing the content type. (ReadOnly) |
| | UserID (see page 533) | Gets a long representing the user ID (see page 532) of the user performing the operation on the content. (ReadOnly) |

### ContentInfo.AssetID Property

Gets a string containing the asset ID (see page 532) in the form of a GUID if the content is an asset. (ReadOnly)

### C#

```
public string AssetID;
```

### Visual Basic

```
Public Property AssetID() As string
```

### ContentInfo.Comments Property

Gets a string containing the comments a user has entered during their operation. (ReadOnly)

**C#**

```
public string Comments;
```

**Visual Basic**

```
Public Property Comments() As string
```

### ContentInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

**C#**

```
public string ErrorMessage;
```

**Visual Basic**

```
Public Property ErrorMessage() As string
```

### ContentInfo.FileAndPath Property

Gets a string containing the full physical path and file name on the server if content is an asset. (ReadOnly)

**C#**

```
public string FileAndPath;
```

**Visual Basic**

```
Public Property FileAndPath() As string
```

### ContentInfo.FileExtension Property

Gets a string containing file extension if content is an asset. (ReadOnly)

**C#**

```
public string FileExtension;
```

**Visual Basic**

```
Public Property FileExtension() As string
```

### ContentInfo.FileName Property

Gets a string containing file name if content is an asset. (ReadOnly)

**C#**

```
public string FileName;
```

**Visual Basic**

```
Public Property FileName() As string
```

### ContentInfo.FolderID Property

Gets an long representing the folder ID (see page 532) which this content is located. (ReadOnly)

**C#**

```
public long FolderID;
```

**Visual Basic**

```
Public Property FolderID() As long
```

### ContentInfo.Html Property

Gets or Sets a string containing all the html displayed to end users for a piece of content. (ReadWrite Note: This property only allows write access within OnBeforePublish.)

**C#**

```
public string Html;
```

**Visual Basic**

```
Public Property Html() As string
```

### ContentInfo.ID Property

Gets an long representing the content ID of the content. (ReadOnly)

**C#**

```
public long ID;
```

**Visual Basic**

```
Public Property ID() As long
```

### ContentInfo.Language Property

Gets an integer representing the language of the content. (ReadOnly)

**C#**

```
public int Language;
```

**Visual Basic**

```
Public Property Language() As Integer
```

### ContentInfo.MimeType Property

Gets a string containing the MIME type if content is an asset. (ReadOnly)

**C#**

```
public string MimeType;
```

**Visual Basic**

```
Public Property MimeType() As string
```

### ContentInfo.Status Property

Gets a string representing the status of the content, published, checkedOut. (ReadOnly)

**C#**

```
public string Status;
```

**Visual Basic**

```
Public Property Status() As string
```

### ContentInfo.Title Property

Gets a string containing the title of the content. (ReadOnly)

**C#**

```
public string Title;
```

**Visual Basic**

```
Public Property Title() As string
```

### ContentInfo.Type Property

Gets an integer representing the content type. (ReadOnly)

**C#**

```
public int Type;
```

**Visual Basic**

```
Public Property Type() As Integer
```

### ContentInfo.UserID Property

Gets a long representing the user ID (see page 532) of the user performing the operation on the content. (ReadOnly)

**C#**

```
public long UserID;
```

**Visual Basic**

```
Public Property UserID() As long
```

## ExtensionAuthor Class

Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in.

**Class Hierarchy**

**C#**

```
[AttributeUsage(AttributeTargets.Method | AttributeTargets.Assembly)]
public class ExtensionAuthor : Attribute;
```

**Visual Basic**

```
<AttributeUsage(AttributeTargets.Method | AttributeTargets.Assembly)>
Public Class ExtensionAuthor
Inherits Attribute
```

**File**

ExtensibilityBase.cs

## ExtensionAuthor.ExtensionAuthor Constructor

Sets the creator of the Extension.

**C#**

```
public ExtensionAuthor(string author);
```

**Visual Basic**

```
Public Sub New(author As string)
```

**Parameters**

| Parameters | Description |
|---|---|
| author | String with name of author who created Extension. |

## ExtensionAuthor Methods

The methods of the ExtensionAuthor class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | ToString (see page 534) | Converts author name to a string. |

### ExtensionAuthor.ToString Method

Converts author name to a string.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

**Returns**

String representing the author's name.

## ExtensionAuthor Properties

The properties of the ExtensionAuthor class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Author (see page 534) | Gets the creator of the Extension. (Read Only) |

### ExtensionAuthor.Author Property

Gets the creator of the Extension. (Read Only)

**C#**

```
public string Author;
```

**Visual Basic**

```
Public Property Author() As string
```

# ExtensionDescription Class

Custom attribute - Used at the method level to describe what a particular event does

### Class Hierarchy

**C#**

```
[AttributeUsage(AttributeTargets.Method)]
public class ExtensionDescription : Attribute;
```

**Visual Basic**

```
<AttributeUsage(AttributeTargets.Method)>
Public Class ExtensionDescription
Inherits Attribute
```

**File**

ExtensibilityBase.cs

## ExtensionDescription.ExtensionDescription Constructor

Sets the description of the Extension.

**C#**

```
public ExtensionDescription(string description);
```

**Visual Basic**

```
Public Sub New(description As string)
```

**Parameters**

| Parameters | Description |
|---|---|
| description | String representing the description of the Extension |

## ExtensionDescription Methods

The methods of the ExtensionDescription class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | ToString (see page 535) | Converts the description to a string. |

### ExtensionDescription.ToString Method

Converts the description to a string.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

**Returns**

String representing the description of the Extension.

## ExtensionDescription Properties

The properties of the ExtensionDescription class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Desc (see page 535) | Gets the description of the extension. (ReadOnly) |

### ExtensionDescription.Desc Property

Gets the description of the extension. (ReadOnly)

**C#**

```
public string Desc;
```

**Visual Basic**

```
Public Property Desc() As string
```

# ExtensionEvent Class

Override the methods in this class to customize the CMS via event hooks.

### Class Hierarchy

### C#

```
[Serializable]
public abstract class ExtensionEvent : IWebServiceProxy;
```

### Visual Basic

```
<Serializable>
Public abstract Class ExtensionEvent
Inherits IWebServiceProxy
```

### File

ExtensibilityBase.cs

## ExtensionEvent Methods

The methods of the ExtensionEvent class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | GetActiveDirectoryUserProxy (see page 538) | Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 974) Web Service. |
| | GetAssetProxy (see page 538) | Returns an AssetProxy object to access the Asset (see page 789) Web Service. |
| | GetBlogProxy (see page 538) | Returns a BlogProxy object to access the Blog (see page 792) Web Service. |
| | GetBusinessRulesProxy (see page 539) | Returns a BusinessRulesProxy object to access the BusinessRules (see page 66) Web Service. |
| | GetCalendarEventTypeProxy (see page 539) | Returns a CalendarEventTypeProxy object to access the CalendarEventType Web Service. |
| | GetCalendarProxy (see page 539) | Returns a CalendarProxy object to access the Calendar (see page 69) Web Service. |
| | GetCommunityGroupProxy (see page 539) | Returns a CommunityGroupProxy object to access the CommunityGroup (see page 719) Web Service. |
| | GetContentProxy (see page 539) | Returns a ContentProxy object to access the Content (see page 552) Web Service. |
| | GetContentRatingProxy (see page 540) | Returns a ContentRatingProxy object to access the ContentRating (see page 831) Web Service. |
| | GetCustomFieldsProxy (see page 540) | Returns a CustomFieldsProxy object to access the CustomFields (see page 642) Web Service. |
| | GetFavoritesProxy (see page 540) | Returns a FavoritesProxy object to access the Favorites (see page 731) Web Service. |
| | GetFlagProxy (see page 540) | Returns a FlaggedDataProxy object to access the Content (see page 552) flagging Web Service. |
| | GetFolderProxy (see page 541) | Returns a FolderProxy object to access the Folder (see page 553) Web Service. |
| | GetFontProxy (see page 541) | Returns a FontProxy object to access the Font (see page 1055) Web Service. |
| | GetFormProxy (see page 541) | Returns a FormProxy object to access the Form (see page 553) Web Service. |
| | GetFriendsProxy (see page 541) | Returns a FriendsProxy object to access the FavoriFriendstes Web Service. |
| | GetLibraryProxy (see page 542) | Returns a LibraryProxy object to access the Library (see page 664) Web Service. |
| | GetMessageBoardProxy (see page 542) | Returns a MessageBoardProxy object to access the MessageBoard (see page 755) Web Service. |
| | GetMetadataProxy (see page 542) | Returns a MetadataProxy object to access the Metadata (see page 683) Web Service. |

| | | |
|---|---|---|
| | GetObjectStateAsXml (see page 542) | Retrieves changes made by individual plug-ins |
| | GetPermissionsProxy (see page 542) | Returns a PermissionsProxy object to access the Permissions (see page 690) Web Service. |
| | GetSearchManagerProxy (see page 543) | Returns a SearchManagerProxy object to access the SearchManager (see page 865) Web Service. |
| | GetSiteMapProxy (see page 543) | Returns a SiteMapProxy object to access the SiteMap (see page 701) Web Service. |
| | GetSiteProxy (see page 543) | Returns a SiteProxy object to access the Site (see page 553) Web Service. |
| | GetTagsProxy (see page 543) | Returns a TagsProxy object to access the Tags (see page 771) Web Service. |
| | GetTaskCategoryProxy (see page 544) | Returns a TaskCategoryProxy object to access the Task (see page 871) Category Web Service. |
| | GetTaskCategoryTypeProxy (see page 544) | Returns a TaskProxy object to access the Task (see page 871) Web Service. |
| | GetTaskProxy (see page 544) | Returns a TaskProxy object to access the Task (see page 871) Web Service. |
| | GetTaxonomyProxy (see page 544) | Returns a TaxonomyDataProxy object to access the taxonomy Web Service. |
| | GetThreadedDiscussionProxy (see page 545) | Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 848) Web Service. |
| | GetUserProxy (see page 545) | Returns a UserProxy object to access the User (see page 553) Web Service. |
| | OnAfterActionRequestSent (see page 545) | This hook is called after an ActionRequest is Sent. |
| | OnAfterContentAdd (see page 545) | This hook is called after content is added in the CMS. |
| | OnAfterContentDelete (see page 546) | This hook is called after a delete of content is completed in the CMS. |
| | OnAfterContentEdit (see page 546) | This hook is called after an edit is completed in the CMS. |
| | OnAfterContentFlagged (see page 546) | This hook is called after an Content (see page 552) is flagged. |
| | OnAfterFolderAdd (see page 546) | This hook is called after a folder is created in the CMS. |
| | OnAfterFolderDelete (see page 546) | This hook is called after a folder is deleted in the CMS. |
| | OnAfterFolderEdit (see page 547) | This hook is called after a folder edit is completed in the CMS. |
| | OnAfterFormSubmit (see page 547) | This hook is called after form data is submitted to the CMS database. |
| | OnAfterPublish (see page 547) | This hook is called after a publish is completed in the CMS. |
| | OnAfterReviewRejected (see page 547) | This hook is called after a review is rejected. |
| | OnAfterUserAdd (see page 548) | This hook is called after a user is created in the CMS. |
| | OnAfterUserDelete (see page 548) | This hook is called after a user is deleted from the CMS. |
| | OnAfterUserEdit (see page 548) | This hook is called after a user is edited in the CMS. |
| | OnAfterUserLogin (see page 548) | This hook is called after a user logs into in the CMS. |
| | OnBeforeActionRequestSent (see page 549) | This hook is called before an ActionRequest is Sent. |
| | OnBeforeContentAdd (see page 549) | This hook is called before content is added to the CMS. |
| | OnBeforeContentDelete (see page 549) | This hook is called before a delete of content in the CMS. |
| | OnBeforeContentEdit (see page 549) | This hook is called before a content is edited in the CMS. |
| | OnBeforeContentFlagged (see page 550) | This hook is called before an Content (see page 552) is flagged. |

| | | |
|---|---|---|
| | OnBeforeFolderAdd (see page 550) | This hook is called before a folder is created in the CMS. |
| | OnBeforeFolderDelete (see page 550) | This hook is called before a folder is deleted from the CMS. |
| | OnBeforeFolderEdit (see page 550) | This hook is called before a folder is edited in the CMS. |
| | OnBeforeFormSubmit (see page 550) | This hook is called before form data is submitted to the CMS database. |
| | OnBeforePublish (see page 551) | This hook is called before a publish takes place in the CMS. |
| | OnBeforeReviewRejected (see page 551) | This hook is called before a review is rejected. |
| | OnBeforeUserAdd (see page 551) | This hook is called before a user is created in the CMS. |
| | OnBeforeUserDelete (see page 551) | This hook is called before a user is delete from the CMS. |
| | OnBeforeUserEdit (see page 552) | This hook is called before a user is edited in the CMS. |
| | OnBeforeUserLogin (see page 552) | This hook is called before a user login is initiated in the CMS. |

### ExtensionEvent.GetActiveDirectoryUserProxy Method

Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 974) Web Service.

**C#**

```
public ActiveDirectoryUserProxy GetActiveDirectoryUserProxy();
```

**Visual Basic**

```
Public Function GetActiveDirectoryUserProxy() As ActiveDirectoryUserProxy
```

**Returns**

ActiveDirectoryUserProxy

### ExtensionEvent.GetAssetProxy Method

Returns an AssetProxy object to access the Asset (see page 789) Web Service.

**C#**

```
public AssetProxy GetAssetProxy();
```

**Visual Basic**

```
Public Function GetAssetProxy() As AssetProxy
```

**Returns**

AssetProxy

### ExtensionEvent.GetBlogProxy Method

Returns a BlogProxy object to access the Blog (see page 792) 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 66) Web Service.

**C#**

```
public BusinessRulesProxy GetBusinessRulesProxy();
```

**Visual Basic**

```
Public Function GetBusinessRulesProxy() As BusinessRulesProxy
```

**Returns**

BusinessRulesProxy

### ExtensionEvent.GetCalendarEventTypeProxy Method

Returns a CalendarEventTypeProxy object to access the CalendarEventType Web Service.

**C#**

```
public CalendarEventTypeProxy GetCalendarEventTypeProxy();
```

**Visual Basic**

```
Public Function GetCalendarEventTypeProxy() As CalendarEventTypeProxy
```

**Returns**

CalendarEventTypeProxy

### ExtensionEvent.GetCalendarProxy Method

Returns a CalendarProxy object to access the Calendar (see page 69) Web Service.

**C#**

```
public CalendarProxy GetCalendarProxy();
```

**Visual Basic**

```
Public Function GetCalendarProxy() As CalendarProxy
```

**Returns**

CalendarProxy

### ExtensionEvent.GetCommunityGroupProxy Method

Returns a CommunityGroupProxy object to access the CommunityGroup (see page 719) Web Service.

**C#**

```
public CommunityGroupProxy GetCommunityGroupProxy();
```

**Visual Basic**

```
Public Function GetCommunityGroupProxy() As CommunityGroupProxy
```

**Returns**

CommunityGroupProxy

### ExtensionEvent.GetContentProxy Method

Returns a ContentProxy object to access the Content (see page 552) 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 831) 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 642) Web Service.

**C#**

```
public CustomFieldsProxy GetCustomFieldsProxy();
```

**Visual Basic**

```
Public Function GetCustomFieldsProxy() As CustomFieldsProxy
```

**Returns**

CustomFieldsProxy

### ExtensionEvent.GetFavoritesProxy Method

Returns a FavoritesProxy object to access the Favorites (see page 731) Web Service.

**C#**

```
public FavoritesProxy GetFavoritesProxy();
```

**Visual Basic**

```
Public Function GetFavoritesProxy() As FavoritesProxy
```

**Returns**

FavoritesProxy

### ExtensionEvent.GetFlagProxy Method

Returns a FlaggedDataProxy object to access the Content (see page 552) flagging Web Service.

**C#**

```
public FlagProxy GetFlagProxy();
```

**Visual Basic**

```
Public Function GetFlagProxy() As FlagProxy
```

**Returns**

FlagProxy

### ExtensionEvent.GetFolderProxy Method

Returns a FolderProxy object to access the Folder (see page 553) Web Service.

**C#**

```
public FolderProxy GetFolderProxy();
```

**Visual Basic**

```
Public Function GetFolderProxy() As FolderProxy
```

**Returns**

FolderProxy

### ExtensionEvent.GetFontProxy Method

Returns a FontProxy object to access the Font (see page 1055) Web Service.

**C#**

```
public FontProxy GetFontProxy();
```

**Visual Basic**

```
Public Function GetFontProxy() As FontProxy
```

**Returns**

FontProxy

### ExtensionEvent.GetFormProxy Method

Returns a FormProxy object to access the Form (see page 553) Web Service.

**C#**

```
public FormProxy GetFormProxy();
```

**Visual Basic**

```
Public Function GetFormProxy() As FormProxy
```

**Returns**

FormProxy

### ExtensionEvent.GetFriendsProxy Method

Returns a FriendsProxy object to access the FavoriFriendstes Web Service.

**C#**

```
public FriendsProxy GetFriendsProxy();
```

**Visual Basic**

```
Public Function GetFriendsProxy() As FriendsProxy
```

**Returns**

FriendsProxy

### ExtensionEvent.GetLibraryProxy Method

Returns a LibraryProxy object to access the Library (see page 664) Web Service.

**C#**

```
public LibraryProxy GetLibraryProxy();
```

**Visual Basic**

```
Public Function GetLibraryProxy() As LibraryProxy
```

**Returns**

LibraryProxy

### ExtensionEvent.GetMessageBoardProxy Method

Returns a MessageBoardProxy object to access the MessageBoard (see page 755) Web Service.

**C#**

```
public MessageBoardProxy GetMessageBoardProxy();
```

**Visual Basic**

```
Public Function GetMessageBoardProxy() As MessageBoardProxy
```

**Returns**

MessageBoardProxy

### ExtensionEvent.GetMetadataProxy Method

Returns a MetadataProxy object to access the Metadata (see page 683) 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 690) 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 865) 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 701) Web Service.

**C#**

```
public SiteMapProxy GetSiteMapProxy();
```

**Visual Basic**

```
Public Function GetSiteMapProxy() As SiteMapProxy
```

**Returns**

SiteMapProxy

### ExtensionEvent.GetSiteProxy Method

Returns a SiteProxy object to access the Site (see page 553) Web Service.

**C#**

```
public SiteProxy GetSiteProxy();
```

**Visual Basic**

```
Public Function GetSiteProxy() As SiteProxy
```

**Returns**

SiteProxy

### ExtensionEvent.GetTagsProxy Method

Returns a TagsProxy object to access the Tags (see page 771) Web Service.

**C#**

```
public TagsProxy GetTagsProxy();
```

**Visual Basic**

```
Public Function GetTagsProxy() As TagsProxy
```

**Returns**

TagsProxy

### ExtensionEvent.GetTaskCategoryProxy Method

Returns a TaskCategoryProxy object to access the Task (see page 871) Category Web Service.

**C#**

```
public TaskCategoryProxy GetTaskCategoryProxy();
```

**Visual Basic**

```
Public Function GetTaskCategoryProxy() As TaskCategoryProxy
```

**Returns**

TaskCategoryProxy

### ExtensionEvent.GetTaskCategoryTypeProxy Method

Returns a TaskProxy object to access the Task (see page 871) Web Service.

**C#**

```
public TaskCategoryTypeProxy GetTaskCategoryTypeProxy();
```

**Visual Basic**

```
Public Function GetTaskCategoryTypeProxy() As TaskCategoryTypeProxy
```

**Returns**

TaskProxy

### ExtensionEvent.GetTaskProxy Method

Returns a TaskProxy object to access the Task (see page 871) Web Service.

**C#**

```
public TaskProxy GetTaskProxy();
```

**Visual Basic**

```
Public Function GetTaskProxy() As TaskProxy
```

**Returns**

TaskProxy

### ExtensionEvent.GetTaxonomyProxy Method

Returns a TaxonomyDataProxy object to access the taxonomy Web Service.

**C#**

```
public TaxonomyProxy GetTaxonomyProxy();
```

**Visual Basic**

```
Public Function GetTaxonomyProxy() As TaxonomyProxy
```

**Returns**

TaxonomyProxy

### ExtensionEvent.GetThreadedDiscussionProxy Method

Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 848) Web Service.

**C#**

```
public ThreadedDiscussionProxy GetThreadedDiscussionProxy();
```

**Visual Basic**

```
Public Function GetThreadedDiscussionProxy() As ThreadedDiscussionProxy
```

**Returns**

ThreadedDiscussionProxy

### ExtensionEvent.GetUserProxy Method

Returns a UserProxy object to access the User (see page 553) Web Service.

**C#**

```
public UserProxy GetUserProxy();
```

**Visual Basic**

```
Public Function GetUserProxy() As UserProxy
```

**Returns**

UserProxy

### ExtensionEvent.OnAfterActionRequestSent Method

This hook is called after an ActionRequest is Sent.

**C#**

```
public virtual bool OnAfterActionRequestSent();
```

**Visual Basic**

```
Public virtual Function OnAfterActionRequestSent() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterContentAdd Method

This hook is called after content is added in the CMS.

**C#**

```
public virtual bool OnAfterContentAdd();
```

**Visual Basic**

```
Public virtual Function OnAfterContentAdd() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterContentDelete Method

This hook is called after a delete of content is completed in the CMS.

**C#**

```
public virtual bool OnAfterContentDelete();
```

**Visual Basic**

```
Public virtual Function OnAfterContentDelete() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterContentEdit Method

This hook is called after an edit is completed in the CMS.

**C#**

```
public virtual bool OnAfterContentEdit();
```

**Visual Basic**

```
Public virtual Function OnAfterContentEdit() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterContentFlagged Method

This hook is called after an Content (see page 552) is flagged.

**C#**

```
public virtual bool OnAfterContentFlagged();
```

**Visual Basic**

```
Public virtual Function OnAfterContentFlagged() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterFolderAdd Method

This hook is called after a folder is created in the CMS.

**C#**

```
public virtual bool OnAfterFolderAdd();
```

**Visual Basic**

```
Public virtual Function OnAfterFolderAdd() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterFolderDelete Method

This hook is called after a folder is deleted in the CMS.

**C#**

```
public virtual bool OnAfterFolderDelete();
```

**Visual Basic**

```
Public virtual Function OnAfterFolderDelete() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterFolderEdit Method

This hook is called after a folder edit is completed in the CMS.

**C#**

```
public virtual bool OnAfterFolderEdit();
```

**Visual Basic**

```
Public virtual Function OnAfterFolderEdit() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterFormSubmit Method

This hook is called after form data is submitted to the CMS database.

**C#**

```
public virtual bool OnAfterFormSubmit();
```

**Visual Basic**

```
Public virtual Function OnAfterFormSubmit() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterPublish Method

This hook is called after a publish is completed in the CMS.

**C#**

```
public virtual bool OnAfterPublish();
```

**Visual Basic**

```
Public virtual Function OnAfterPublish() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterReviewRejected Method

This hook is called after a review is rejected.

**C#**

```
public virtual bool OnAfterReviewRejected();
```

**Visual Basic**

```
Public virtual Function OnAfterReviewRejected() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterUserAdd Method

This hook is called after a user is created in the CMS.

**C#**

```
public virtual bool OnAfterUserAdd();
```

**Visual Basic**

```
Public virtual Function OnAfterUserAdd() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterUserDelete Method

This hook is called after a user is deleted from the CMS.

**C#**

```
public virtual bool OnAfterUserDelete();
```

**Visual Basic**

```
Public virtual Function OnAfterUserDelete() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterUserEdit Method

This hook is called after a user is edited in the CMS.

**C#**

```
public virtual bool OnAfterUserEdit();
```

**Visual Basic**

```
Public virtual Function OnAfterUserEdit() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnAfterUserLogin Method

This hook is called after a user logs into in the CMS.

**C#**

```
public virtual bool OnAfterUserLogin();
```

**Visual Basic**

```
Public virtual Function OnAfterUserLogin() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeActionRequestSent Method

This hook is called before an ActionRequest is Sent.

**C#**

```
public virtual bool OnBeforeActionRequestSent();
```

**Visual Basic**

```
Public virtual Function OnBeforeActionRequestSent() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeContentAdd Method

This hook is called before content is added to the CMS.

**C#**

```
public virtual bool OnBeforeContentAdd();
```

**Visual Basic**

```
Public virtual Function OnBeforeContentAdd() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeContentDelete Method

This hook is called before a delete of content in the CMS.

**C#**

```
public virtual bool OnBeforeContentDelete();
```

**Visual Basic**

```
Public virtual Function OnBeforeContentDelete() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeContentEdit Method

This hook is called before a content is edited in the CMS.

**C#**

```
public virtual bool OnBeforeContentEdit();
```

**Visual Basic**

```
Public virtual Function OnBeforeContentEdit() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeContentFlagged Method

This hook is called before an Content (see page 552) is flagged.

**C#**

```
public virtual bool OnBeforeContentFlagged();
```

**Visual Basic**

```
Public virtual Function OnBeforeContentFlagged() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeFolderAdd Method

This hook is called before a folder is created in the CMS.

**C#**

```
public virtual bool OnBeforeFolderAdd();
```

**Visual Basic**

```
Public virtual Function OnBeforeFolderAdd() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeFolderDelete Method

This hook is called before a folder is deleted from the CMS.

**C#**

```
public virtual bool OnBeforeFolderDelete();
```

**Visual Basic**

```
Public virtual Function OnBeforeFolderDelete() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeFolderEdit Method

This hook is called before a folder is edited in the CMS.

**C#**

```
public virtual bool OnBeforeFolderEdit();
```

**Visual Basic**

```
Public virtual Function OnBeforeFolderEdit() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeFormSubmit Method

This hook is called before form data is submitted to the CMS database.

**C#**

```
public virtual bool OnBeforeFormSubmit();
```

**Visual Basic**

```
Public virtual Function OnBeforeFormSubmit() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforePublish Method

This hook is called before a publish takes place in the CMS.

**C#**

```
public virtual bool OnBeforePublish();
```

**Visual Basic**

```
Public virtual Function OnBeforePublish() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeReviewRejected Method

This hook is called before a review is rejected.

**C#**

```
public virtual bool OnBeforeReviewRejected();
```

**Visual Basic**

```
Public virtual Function OnBeforeReviewRejected() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeUserAdd Method

This hook is called before a user is created in the CMS.

**C#**

```
public virtual bool OnBeforeUserAdd();
```

**Visual Basic**

```
Public virtual Function OnBeforeUserAdd() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeUserDelete Method

This hook is called before a user is delete from the CMS.

**C#**

```
public virtual bool OnBeforeUserDelete();
```

**Visual Basic**

```
Public virtual Function OnBeforeUserDelete() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeUserEdit Method

This hook is called before a user is edited in the CMS.

**C#**

```
public virtual bool OnBeforeUserEdit();
```

**Visual Basic**

```
Public virtual Function OnBeforeUserEdit() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

### ExtensionEvent.OnBeforeUserLogin Method

This hook is called before a user login is initiated in the CMS.

**C#**

```
public virtual bool OnBeforeUserLogin();
```

**Visual Basic**

```
Public virtual Function OnBeforeUserLogin() As bool
```

**Returns**

A bool indicating success or failure. False indicates failure.

## ExtensionEvent Properties

The properties of the ExtensionEvent class are listed here.

**Protected Properties**

| | Name | Description |
|---|---|---|
| | Content (see page 552) | 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 553) | 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 553) | Returns object containing information about the Form, when the operation is form based. |
| | Site (see page 553) | Returns object containing information about the Site that fired the event. |
| | User (see page 553) | Returns object containing information about the User, when the operation is user based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false. |

### ExtensionEvent.Content Property

Returns object containing information about the Content, when the operation is content based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

**C#**

```
protected ContentInfo Content;
```

**Visual Basic**

```
Protected Property Content() As ContentInfo
```

### ExtensionEvent.Folder Property

Returns object containing information about the Folder, when the operation is folder based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

**C#**

```
protected FolderInfo Folder;
```

**Visual Basic**

```
Protected Property Folder() As FolderInfo
```

### ExtensionEvent.Form Property

Returns object containing information about the Form, when the operation is form based.

**C#**

```
protected FormInfo Form;
```

**Visual Basic**

```
Protected Property Form() As FormInfo
```

### ExtensionEvent.Site Property

Returns object containing information about the Site that fired the event.

**C#**

```
protected SiteInfo Site;
```

**Visual Basic**

```
Protected Property Site() As SiteInfo
```

### ExtensionEvent.User Property

Returns object containing information about the User, when the operation is user based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

**C#**

```
protected UserInfo User;
```

**Visual Basic**

```
Protected Property User() As UserInfo
```

## FlagInfo Class

Class which holds information about the flagged content. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

**Class Hierarchy**

**C#**

```
public sealed class FlagInfo;
```

**Visual Basic**

```
Public sealed Class FlagInfo
```

**File**

CmsInfoObjects.cs

## FlagInfo Properties

The properties of the FlagInfo class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | ContentHtml (see page 554) | Gets a string containing all the html displayed to end users for a piece of content. (ReadOnly) |
| | ContentID (see page 554) | Gets a numeric value representing the content ID of the content. (ReadOnly) |
| | ContentLanguage (see page 555) | Gets an integer representing the language of the content. (ReadOnly) |
| | ContentTitle (see page 555) | Gets a string containing the title of the content. (ReadOnly) |
| | EntryId (see page 555) | Gets a numeric value representing the flagged entry ID. (ReadOnly) |
| | ErrorMessage (see page 555) | Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite) |
| | FlagComment (see page 555) | Gets an string representing the flagged comment. (ReadOnly) |
| | FlagDate (see page 555) | Gets a datetime representing the flagged date time. (ReadOnly) |
| | FlagId (see page 556) | Gets a numeric value representing the flag ID associated to this content. (ReadOnly) |
| | FlagName (see page 556) | Gets an string representing the flag name associated to this content. (ReadOnly) |
| | UserAvatar (see page 556) | Gets an string representing path of the user's avatar. (ReadOnly) |
| | UserDisplayName (see page 556) | Gets an string representing the user's display name. (ReadOnly) |
| | UserFirstName (see page 556) | Gets an string representing the user's first name. (ReadOnly) |
| | UserID (see page 556) | Gets a numeric value representing the user ID of the user. (ReadOnly) |
| | UserLastName (see page 557) | Gets an string representing the user's last name. (ReadOnly) |
| | UserName (see page 557) | Gets an string representing the user's CMS user name. (ReadOnly) |
| | VisitorID (see page 557) | Gets an string representing the cookie value of the visitor. (ReadOnly) |

### FlagInfo.ContentHtml Property

Gets a string containing all the html displayed to end users for a piece of content. (ReadOnly)

**C#**

```
public string ContentHtml;
```

**Visual Basic**

```
Public Property ContentHtml() As string
```

### FlagInfo.ContentID Property

Gets a numeric value representing the content ID of the content. (ReadOnly)

**C#**

```
public long ContentID;
```

**Visual Basic**

```
Public Property ContentID() As long
```

### FlagInfo.ContentLanguage Property

Gets an integer representing the language of the content. (ReadOnly)

**C#**

```
public int ContentLanguage;
```

**Visual Basic**

```
Public Property ContentLanguage() As Integer
```

### FlagInfo.ContentTitle Property

Gets a string containing the title of the content. (ReadOnly)

**C#**

```
public string ContentTitle;
```

**Visual Basic**

```
Public Property ContentTitle() As string
```

### FlagInfo.EntryId Property

Gets a numeric value representing the flagged entry ID. (ReadOnly)

**C#**

```
public long EntryId;
```

**Visual Basic**

```
Public Property EntryId() As long
```

### FlagInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

**C#**

```
public string ErrorMessage;
```

**Visual Basic**

```
Public Property ErrorMessage() As string
```

### FlagInfo.FlagComment Property

Gets an string representing the flagged comment. (ReadOnly)

**C#**

```
public string FlagComment;
```

**Visual Basic**

```
Public Property FlagComment() As string
```

### FlagInfo.FlagDate Property

Gets a datetime representing the flagged date time. (ReadOnly)

**C#**

```
public DateTime FlagDate;
```

**Visual Basic**

```
Public Property FlagDate() As DateTime
```

### FlagInfo.FlagId Property

Gets a numeric value representing the flag ID associated to this content. (ReadOnly)

**C#**

```
public long FlagId;
```

**Visual Basic**

```
Public Property FlagId() As long
```

### FlagInfo.FlagName Property

Gets an string representing the flag name associated to this content. (ReadOnly)

**C#**

```
public string FlagName;
```

**Visual Basic**

```
Public Property FlagName() As string
```

### FlagInfo.UserAvatar Property

Gets an string representing path of the user's avatar. (ReadOnly)

**C#**

```
public string UserAvatar;
```

**Visual Basic**

```
Public Property UserAvatar() As string
```

### FlagInfo.UserDisplayName Property

Gets an string representing the user's display name. (ReadOnly)

**C#**

```
public string UserDisplayName;
```

**Visual Basic**

```
Public Property UserDisplayName() As string
```

### FlagInfo.UserFirstName Property

Gets an string representing the user's first name. (ReadOnly)

**C#**

```
public string UserFirstName;
```

**Visual Basic**

```
Public Property UserFirstName() As string
```

### FlagInfo.UserID Property

Gets a numeric value representing the user ID of the user. (ReadOnly)

**C#**

```
public long UserID;
```

**Visual Basic**

```
Public Property UserID() As long
```

### FlagInfo.UserLastName Property

Gets an string representing the user's last name. (ReadOnly)

**C#**

```
public string UserLastName;
```

**Visual Basic**

```
Public Property UserLastName() As string
```

### FlagInfo.UserName Property

Gets an string representing the user's CMS user name. (ReadOnly)

**C#**

```
public string UserName;
```

**Visual Basic**

```
Public Property UserName() As string
```

### FlagInfo.VisitorID Property

Gets an string representing the cookie value of the visitor. (ReadOnly)

**C#**

```
public string VisitorID;
```

**Visual Basic**

```
Public Property VisitorID() As string
```

## FolderInfo Class

Class which holds information about the folder being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

**Class Hierarchy**

Ektron.Cms.Extensibility.FolderInfo

**C#**

```
public sealed class FolderInfo;
```

**Visual Basic**

```
Public sealed Class FolderInfo
```

**File**

CmsInfoObjects.cs

## FolderInfo Properties

The properties of the FolderInfo class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Description (see page 558) | Gets a string containing the description of the folder. (ReadOnly) |
| | ErrorMessage (see page 558) | Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite) |
| | ID (see page 558) | Gets a numeric value representing the folder ID of the folder. (ReadOnly) |

| | | |
|---|---|---|
| | Name (see page 558) | Gets a string containing the folder name. (ReadOnly) |
| | ParentID (see page 558) | Gets a numeric value representing the folder ID (see page 558) of the parent folder. (ReadOnly) |
| | ReplicationState (see page 558) | Gets an integer representing if replication is turned on for this folder. (ReadOnly) |
| | StyleSheet (see page 559) | Gets a string containing the style sheet to be applied to content in this folder. (ReadOnly) |
| | Type (see page 559) | Gets a string containing the folder type. (ReadOnly) |

### FolderInfo.Description Property

Gets a string containing the description of the folder. (ReadOnly)

**C#**

```
public string Description;
```

**Visual Basic**

```
Public Property Description() As string
```

### FolderInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

**C#**

```
public string ErrorMessage;
```

**Visual Basic**

```
Public Property ErrorMessage() As string
```

### FolderInfo.ID Property

Gets a numeric value representing the folder ID of the folder. (ReadOnly)

**C#**

```
public long ID;
```

**Visual Basic**

```
Public Property ID() As long
```

### FolderInfo.Name Property

Gets a string containing the folder name. (ReadOnly)

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

### FolderInfo.ParentID Property

Gets a numeric value representing the folder ID (see page 558) of the parent folder. (ReadOnly)

**C#**

```
public long ParentID;
```

**Visual Basic**

```
Public Property ParentID() As long
```

### FolderInfo.ReplicationState Property

Gets an integer representing if replication is turned on for this folder. (ReadOnly)

**C#**

```
public int ReplicationState;
```

**Visual Basic**

```
Public Property ReplicationState() As Integer
```

### FolderInfo.StyleSheet Property

Gets a string containing the style sheet to be applied to content in this folder. (ReadOnly)

**C#**

```
public string StyleSheet;
```

**Visual Basic**

```
Public Property StyleSheet() As string
```

### FolderInfo.Type Property

Gets a string containing the folder type. (ReadOnly)

**C#**

```
public string Type;
```

**Visual Basic**

```
Public Property Type() As string
```

## FormField Class

Class that describes a field in a form - name / value pair

**Class Hierarchy**

**C#**

```
public sealed class FormField;
```

**Visual Basic**

```
Public sealed Class FormField
```

**File**

CmsInfoObjects.cs

### FormField.FormField Constructor

**C#**

```
public FormField(string FieldName, string FieldValue);
```

**Visual Basic**

```
Public Sub New(FieldName As string, FieldValue As string)
```

**Parameters**

| Parameters | Description |
|---|---|
| FieldName | |
| FieldValue | |

## FormField Properties

The properties of the FormField class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | FieldName (see page 560) | Gets the name of the field as a string |
| | FieldValue (see page 560) | Gets the value of the field as a string |

### FormField.FieldName Property

Gets the name of the field as a string

**C#**

```
public string FieldName;
```

**Visual Basic**

```
Public Property FieldName() As string
```

### FormField.FieldValue Property

Gets the value of the field as a string

**C#**

```
public string FieldValue;
```

**Visual Basic**

```
Public Property FieldValue() As string
```

# FormInfo Class

Class which describes a Form (see page 836) object

### Class Hierarchy

Ektron.Cms.Extensibility.FormInfo

**C#**

```
public sealed class FormInfo;
```

**Visual Basic**

```
Public sealed Class FormInfo
```

**File**

CmsInfoObjects.cs

## FormInfo Methods

The methods of the FormInfo class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | this (see page 560) | Gets a single FormField (see page 559) object indexed by Field Name |

### FormInfo.this Indexer

Gets a single FormField (see page 559) 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 561) | Gets or sets a string containing message to display to the end user if plug-in returns false. (ReadWrite) |
| | FormDescription (see page 561) | Gets the description of the form as a string |
| | FormFields (see page 561) | Gets a collection of the Form (see page 836) Field objects as a System.Collections.Generic.List |
| | FormTitle (see page 561) | Gets the title for the form as a string |

### FormInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if plug-in returns false. (ReadWrite)

**C#**

```
public string ErrorMessage;
```

**Visual Basic**

```
Public Property ErrorMessage() As string
```

### FormInfo.FormDescription Property

Gets the description of the form as a string

**C#**

```
public string FormDescription;
```

**Visual Basic**

```
Public Property FormDescription() As string
```

### FormInfo.FormFields Property

Gets a collection of the Form (see page 836) Field objects as a System.Collections.Generic.List

**C#**

```
public List<FormField> FormFields;
```

**Visual Basic**

```
Public Property FormFields() As List<FormField>
```

### FormInfo.FormTitle Property

Gets the title for the form as a string

**C#**

```
public string FormTitle;
```

**Visual Basic**

```
Public Property FormTitle() As string
```

# MessageBoardInfo Class

Class which holds information about the Invitation Recipient being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

**Class Hierarchy**

**C#**

```
public sealed class MessageBoardInfo;
```

**Visual Basic**

```
Public sealed Class MessageBoardInfo
```

**File**

CmsInfoObjects.cs

## MessageBoardInfo.MessageBoardInfo Constructor

Constructor

**C#**

```
public MessageBoardInfo(XmlNode xml);
```

**Visual Basic**

```
Public Sub New(xml As XmlNode)
```

**Parameters**

| Parameters | Description |
|---|---|
| xml | node containing Recipient info. |

## MessageBoardInfo Properties

The properties of the MessageBoardInfo class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | ErrorMessage (see page 562) | Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite) |

### MessageBoardInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

**C#**

```
public string ErrorMessage;
```

**Visual Basic**

```
Public Property ErrorMessage() As string
```

# RequestInfo Class

Reserved for future use.

### Class Hierarchy

Ektron.Cms.Extensibility.RequestInfo

### C#

```
public sealed class RequestInfo;
```

### Visual Basic

```
Public sealed Class RequestInfo
```

### File

CmsInfoObjects.cs

# ReviewInfo Class

Class which holds information about the review being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

### Class Hierarchy

Ektron.Cms.Extensibility.ReviewInfo

### C#

```
public sealed class ReviewInfo;
```

### Visual Basic

```
Public sealed Class ReviewInfo
```

### File

CmsInfoObjects.cs

## ReviewInfo Properties

The properties of the ReviewInfo class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AverageRating (see page 564) | Gets an decimal representing the average rating. (ReadOnly) |
| | ContentHtml (see page 564) | Gets a string containing all the html displayed to end users for a piece of content. (ReadOnly) |
| | ContentID (see page 564) | Gets a numeric value representing the content ID (see page 565) of the content. (ReadOnly) |
| | ContentLanguage (see page 564) | Gets an integer representing the language of the content. (ReadOnly) |
| | ContentTitle (see page 564) | Gets a string containing the title of the content. (ReadOnly) |
| | ErrorMessage (see page 565) | Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite) |
| | ID (see page 565) | Gets a numeric value representing the review ID. (ReadOnly) |
| | NumberofRatings (see page 565) | Gets an integer representing the number of ratings. (ReadOnly) |
| | Rating (see page 565) | Gets an integer representing the rating. (ReadOnly) |
| | RatingDate (see page 565) | Gets a datetime representing the rating time. (ReadOnly) |
| | State (see page 565) | Gets an integer representing the state. (ReadOnly) |

| | | |
|---|---|---|
| | UserAvatar (see page 566) | Gets an string representing path of the user's avatar. (ReadOnly) |
| | UserComments (see page 566) | Gets an string representing the review text. (ReadOnly) |
| | UserDisplayName (see page 566) | Gets an string representing the user's display name. (ReadOnly) |
| | UserFirstName (see page 566) | Gets an string representing the user's first name. (ReadOnly) |
| | UserID (see page 566) | Gets a numeric value representing the user ID (see page 565) of the user. (ReadOnly) |
| | UserLastName (see page 566) | Gets an string representing the user's last name. (ReadOnly) |
| | UserName (see page 567) | Gets an string representing the user's CMS user name. (ReadOnly) |
| | VisitorID (see page 567) | Gets an string representing the cookie value of the visitor. (ReadOnly) |

### ReviewInfo.AverageRating Property

Gets an decimal representing the average rating. (ReadOnly)

**C#**

```
public decimal AverageRating;
```

**Visual Basic**

```
Public Property AverageRating() As decimal
```

### ReviewInfo.ContentHtml Property

Gets a string containing all the html displayed to end users for a piece of content. (ReadOnly)

**C#**

```
public string ContentHtml;
```

**Visual Basic**

```
Public Property ContentHtml() As string
```

### ReviewInfo.ContentID Property

Gets a numeric value representing the content ID (see page 565) of the content. (ReadOnly)

**C#**

```
public long ContentID;
```

**Visual Basic**

```
Public Property ContentID() As long
```

### ReviewInfo.ContentLanguage Property

Gets an integer representing the language of the content. (ReadOnly)

**C#**

```
public int ContentLanguage;
```

**Visual Basic**

```
Public Property ContentLanguage() As Integer
```

### ReviewInfo.ContentTitle Property

Gets a string containing the title of the content. (ReadOnly)

**C#**

```
public string ContentTitle;
```

**Visual Basic**

```
Public Property ContentTitle() As string
```

### ReviewInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

**C#**

```
public string ErrorMessage;
```

**Visual Basic**

```
Public Property ErrorMessage() As string
```

### ReviewInfo.ID Property

Gets a numeric value representing the review ID. (ReadOnly)

**C#**

```
public long ID;
```

**Visual Basic**

```
Public Property ID() As long
```

### ReviewInfo.NumberofRatings Property

Gets an integer representing the number of ratings. (ReadOnly)

**C#**

```
public int NumberofRatings;
```

**Visual Basic**

```
Public Property NumberofRatings() As Integer
```

### ReviewInfo.Rating Property

Gets an integer representing the rating. (ReadOnly)

**C#**

```
public int Rating;
```

**Visual Basic**

```
Public Property Rating() As Integer
```

### ReviewInfo.RatingDate Property

Gets a datetime representing the rating time. (ReadOnly)

**C#**

```
public DateTime RatingDate;
```

**Visual Basic**

```
Public Property RatingDate() As DateTime
```

### ReviewInfo.State Property

Gets an integer representing the state. (ReadOnly)

**C#**

```
public int State;
```

**Visual Basic**

```
Public Property State() As Integer
```

### ReviewInfo.UserAvatar Property

Gets an string representing path of the user's avatar. (ReadOnly)

**C#**

```
public string UserAvatar;
```

**Visual Basic**

```
Public Property UserAvatar() As string
```

### ReviewInfo.UserComments Property

Gets an string representing the review text. (ReadOnly)

**C#**

```
public string UserComments;
```

**Visual Basic**

```
Public Property UserComments() As string
```

### ReviewInfo.UserDisplayName Property

Gets an string representing the user's display name. (ReadOnly)

**C#**

```
public string UserDisplayName;
```

**Visual Basic**

```
Public Property UserDisplayName() As string
```

### ReviewInfo.UserFirstName Property

Gets an string representing the user's first name. (ReadOnly)

**C#**

```
public string UserFirstName;
```

**Visual Basic**

```
Public Property UserFirstName() As string
```

### ReviewInfo.UserID Property

Gets a numeric value representing the user ID (see page 565) of the user. (ReadOnly)

**C#**

```
public long UserID;
```

**Visual Basic**

```
Public Property UserID() As long
```

### ReviewInfo.UserLastName Property

Gets an string representing the user's last name. (ReadOnly)

**C#**

```
public string UserLastName;
```

**Visual Basic**

```
Public Property UserLastName() As string
```

### ReviewInfo.UserName Property

Gets an string representing the user's CMS user name. (ReadOnly)

**C#**

```
public string UserName;
```

**Visual Basic**

```
Public Property UserName() As string
```

### ReviewInfo.VisitorID Property

Gets an string representing the cookie value of the visitor. (ReadOnly)

**C#**

```
public string VisitorID;
```

**Visual Basic**

```
Public Property VisitorID() As string
```

## SiteInfo Class

Class which holds information about the site in which the event was fired.

**Class Hierarchy**

Ektron.Cms.Extensibility.SiteInfo

**C#**

```
public sealed class SiteInfo;
```

**Visual Basic**

```
Public sealed Class SiteInfo
```

**File**

CmsInfoObjects.cs

## SiteInfo Properties

The properties of the SiteInfo class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AppPath (see page 567) | Gets a string containing the full application path of asp.net application (ReadOnly) |
| | IsSSL (see page 568) | Gets a bool indicating if calling site uses SSL. (ReadOnly) |
| | Name (see page 568) | Gets a string containing the name of the calling site. (ReadOnly) |
| | PhysicalSitePath (see page 568) | Gets a string containing the physical path of the application (ReadOnly) |
| | Port (see page 568) | Gets an integer representing the port the calling site is running on. (ReadOnly) |
| | Url (see page 568) | Gets a string containing the URL of the calling site. (ReadOnly) |

### SiteInfo.AppPath Property

Gets a string containing the full application path of asp.net application (ReadOnly)

**C#**

```
public string AppPath;
```

**Visual Basic**

```
Public Property AppPath() As string
```

### SiteInfo.IsSSL Property

Gets a bool indicating if calling site uses SSL. (ReadOnly)

**C#**

```
public bool IsSSL;
```

**Visual Basic**

```
Public Property IsSSL() As bool
```

### SiteInfo.Name Property

Gets a string containing the name of the calling site. (ReadOnly)

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

### SiteInfo.PhysicalSitePath Property

Gets a string containing the physical path of the application (ReadOnly)

**C#**

```
public string PhysicalSitePath;
```

**Visual Basic**

```
Public Property PhysicalSitePath() As string
```

### SiteInfo.Port Property

Gets an integer representing the port the calling site is running on. (ReadOnly)

**C#**

```
public int Port;
```

**Visual Basic**

```
Public Property Port() As Integer
```

### SiteInfo.Url Property

Gets a string containing the URL of the calling site. (ReadOnly)

**C#**

```
public string Url;
```

**Visual Basic**

```
Public Property Url() As string
```

## UserInfo Class

Class which holds information about the user being processed. Certain hooks do not expose all information from the CMS,

default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

### Class Hierarchy

### C#

```
public sealed class UserInfo;
```

### Visual Basic

```
Public sealed Class UserInfo
```

### File

CmsInfoObjects.cs

## UserInfo Properties

The properties of the UserInfo class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Addres (see page 569) | Gets a string containing the user addres |
| | DisplayName (see page 569) | Gets a string containing the display name of the user being added, edited or deleted. (ReadOnly) |
| | Editor (see page 570) | Gets a string containing the editor preference of the user being added, edited or deleted. (ReadOnly) |
| | EmailAddress (see page 570) | Gets a string containing the email address of the user being added, edited or deleted. (ReadOnly) |
| | ErrorMessage (see page 570) | Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite) |
| | FirstName (see page 570) | Gets a string containing the first name of the user being added, edited or deleted. (ReadOnly) |
| | ID (see page 570) | Gets a numeric value representing the user ID of the being added, edited or deleted. (ReadOnly) |
| | Language (see page 570) | Gets a string containing the language preference of the user being added, edited or deleted. (ReadOnly) |
| | LastName (see page 571) | Gets a string containing the last name of the user being added, edited or deleted. (ReadOnly) |
| | Latitude (see page 571) | Gets a double value of map co-ordinate latitude of user address |
| | Longitude (see page 571) | Gets a double value of map co-ordinate longitude of user address |
| | Notifications (see page 571) | Gets an integer representing if the user has turned on email notifications of the user being added, edited or deleted. (ReadOnly) |
| | Type (see page 571) | Integer representing the user type of the user being added, edited or deleted. Only valid with OnBeforeUserAdd and OnAfterUserAdd. (ReadOnly) |
| | UserName (see page 571) | Gets a string containing the login name of the user being added, edited or deleted. (ReadOnly) |

### UserInfo.Addres Property

Gets a string containing the user addres

### C#

```
public string Addres;
```

### Visual Basic

```
Public Property Addres() As string
```

### UserInfo.DisplayName Property

Gets a string containing the display name of the user being added, edited or deleted. (ReadOnly)

### C#

```
public string DisplayName;
```

### Visual Basic

```
Public Property DisplayName() As string
```

### UserInfo.Editor Property

Gets a string containing the editor preference of the user being added, edited or deleted. (ReadOnly)

**C#**

```
public string Editor;
```

**Visual Basic**

```
Public Property Editor() As string
```

### UserInfo.EmailAddress Property

Gets a string containing the email address of the user being added, edited or deleted. (ReadOnly)

**C#**

```
public string EmailAddress;
```

**Visual Basic**

```
Public Property EmailAddress() As string
```

### UserInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user if Plug-in returns false. (ReadWrite)

**C#**

```
public string ErrorMessage;
```

**Visual Basic**

```
Public Property ErrorMessage() As string
```

### UserInfo.FirstName Property

Gets a string containing the first name of the user being added, edited or deleted. (ReadOnly)

**C#**

```
public string FirstName;
```

**Visual Basic**

```
Public Property FirstName() As string
```

### UserInfo.ID Property

Gets a numeric value representing the user ID of the being added, edited or deleted. (ReadOnly)

**C#**

```
public long ID;
```

**Visual Basic**

```
Public Property ID() As long
```

### UserInfo.Language Property

Gets a string containing the language preference of the user being added, edited or deleted. (ReadOnly)

**C#**

```
public string Language;
```

**Visual Basic**

```
Public Property Language() As string
```

### UserInfo.LastName Property

Gets a string containing the last name of the user being added, edited or deleted. (ReadOnly)

**C#**

```
public string LastName;
```

**Visual Basic**

```
Public Property LastName() As string
```

### UserInfo.Latitude Property

Gets a double value of map co-ordinate latitude of user address

**C#**

```
public double Latitude;
```

**Visual Basic**

```
Public Property Latitude() As double
```

### UserInfo.Longitude Property

Gets a double value of map co-ordinate longitude of user address

**C#**

```
public double Longitude;
```

**Visual Basic**

```
Public Property Longitude() As double
```

### UserInfo.Notifications Property

Gets an integer representing if the user has turned on email notifications of the user being added, edited or deleted. (ReadOnly)

**C#**

```
public int Notifications;
```

**Visual Basic**

```
Public Property Notifications() As Integer
```

### UserInfo.Type Property

Integer representing the user type of the user being added, edited or deleted. Only valid with OnBeforeUserAdd and OnAfterUserAdd. (ReadOnly)

**C#**

```
public int Type;
```

**Visual Basic**

```
Public Property Type() As Integer
```

### UserInfo.UserName Property

Gets a string containing the login name of the user being added, edited or deleted. (ReadOnly)

**C#**

```
public string UserName;
```

**Visual Basic**

```
Public Property UserName() As string
```

# .NET Assemblies

The .NET Assemblies are used to access functionality typically provided in the Workarea. For example, adding content, deleting a user or retrieving information about a specific folder. By using the .NET Assemblies you can create custom functionality while keeping within the Ektron Inc. Business Rules.

This section is divided into separate subsections. The first subsection in this section contains classes derived from the Ektron.Cms.API (see page 639) Namespace.

- Ektron.Cms.API.BusinessRule (see page 640)
- Ektron.Cms.API.CustomFields (see page 642)
- Ektron.Cms.API.DateTimeSelector
- Ektron.Cms.API.Folder (see page 652)
- Ektron.Cms.API.Font
- Ektron.Cms.API.Library (see page 664)
- Ektron.Cms.API.Localization (see page 673)
- Ektron.Cms.API.Metadata (see page 683)
- Ektron.Cms.API.Permissions (see page 690)
- Ektron.Cms.API.Site (see page 694)
- Ektron.Cms.API.SiteMap (see page 701)
- Ektron.Cms.API.Subscription (see page 705)
- Ektron.Cms.API.Utilities (see page 713)

The rest of the subsections contain different Namespaces with classes below them.

- Ektron.Cms.API.Analytics.Analytics (see page 714)
- Ektron.Cms.API.Calendar.Calendar
- Ektron.Cms.API.Calendar.CalendarEvent
- Ektron.Cms.API.Calendar.CalendarEventType
- Ektron.Cms.API.Content.Asset (see page 789)
- Ektron.Cms.API.Content.Blog (see page 792)
- Ektron.Cms.API.Content.Content (see page 806)
- Ektron.Cms.API.Content.ContentRating (see page 831)
- Ektron.Cms.API.Content.Form (see page 836)
- Ektron.Cms.API.Content.Taxonomy (see page 842)
- Ektron.Cms.API.Content.ThreadedDiscussion (see page 848)
- Ektron.Cms.API.Search.SearchManager (see page 865)
- Ektron.Cms.API.Task.Task (see page 871)
- Ektron.Cms.API.Task.TaskCategory (see page 882)
- Ektron.Cms.API.Task.TaskCategoryType (see page 884)

- Ektron.Cms.API.User.ADUser (see page 913)
- Ektron.Cms.API.User.User (see page 920)

**Namespaces**

| Name | Description |
|---|---|
| API (see page 639) | The API Namespace contains classes for programmatically accessing features typically found in the Workarea. |

# eCommerce

This section includes API classes for eCommerce. These classes do not use the internal administrator to access their functionality. You must login a user or an administrator when using these methods. This is different from the rest of the API's contained in this documentation.

**Classes**

| | Name | Description |
|---|---|---|
| | AddressApi (see page 574) | Api class for managing Addresses in Cms (see page 7). |
| | BasketApi (see page 577) | The API (see page 639) class used for managing Commerce (see page 1201) shopping carts (baskets). |
| | CatalogEntryApi (see page 583) | API (see page 639) class used for managing catalogs entries for ecommerce. |
| | ContractApi (see page 588) | Api class for managing Commerce (see page 1201) Contracts in Cms (see page 7). |
| | CountryApi (see page 590) | API (see page 639) class for managing countries in CMS400.NET. |
| | CouponApi (see page 592) | Api class for managing Commerce (see page 1201) Coupons in Cms (see page 7). |
| | CurrencyApi (see page 600) | Api class for managing commerce currencies in Cms (see page 7). |
| | CustomerApi (see page 602) | Api class for managing Commerce (see page 1201) Customers in Cms (see page 7). |
| | ExchangeRateApi (see page 605) | Api class for managing commerce exchange rates in Cms (see page 7). |
| | InventoryApi (see page 608) | Api class for managing Commerce (see page 1201) Inventory in Cms (see page 7). |
| | OrderApi (see page 608) | Api class for managing Commerce (see page 1201) Orders in Cms (see page 7). |
| | PackageApi (see page 617) | API (see page 639) class used for managing shipping packages. |
| | ProductTypeApi (see page 619) | Api class for managing Commerce (see page 1201) product types in Cms (see page 7). |
| | RecommendationApi (see page 622) | Api class for managing entry reccomendations. |
| | RegionApi (see page 625) | Api class for managing regions in Cms (see page 7). |
| | ShippingMethodApi (see page 627) | Api class for managing shipping methods in CMS. |
| | TaxApi (see page 628) | Api class for managing Commerce (see page 1201) Tax rates in Cms (see page 7). |
| | TaxClassApi (see page 633) | Api class for managing Commerce (see page 1201) Tax Classes in Cms (see page 7). |
| | TaxTypeApi (see page 635) | Api class for managing Commerce (see page 1201) tax types in Cms (see page 7). |
| | WarehouseApi (see page 636) | This is class Ektron.Cms.Commerce.WarehouseApi. |

# Classes

The following table lists %CATEGORYL% in this documentation.

**Description**

This section includes API classes for eCommerce (see page 573). These classes do not use the internal administrator to access their functionality. You must login a user or an administrator when using these methods. This is different from the rest of the API's contained in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | AddressApi (see page 574) | Api class for managing Addresses in Cms (see page 7). |
| | BasketApi (see page 577) | The API (see page 639) class used for managing Commerce (see page 1201) shopping carts (baskets). |
| | CatalogEntryApi (see page 583) | API (see page 639) class used for managing catalogs entries for ecommerce. |
| | ContractApi (see page 588) | Api class for managing Commerce (see page 1201) Contracts in Cms (see page 7). |
| | CountryApi (see page 590) | API (see page 639) class for managing countries in CMS400.NET. |
| | CouponApi (see page 592) | Api class for managing Commerce (see page 1201) Coupons in Cms (see page 7). |
| | CurrencyApi (see page 600) | Api class for managing commerce currencies in Cms (see page 7). |
| | CustomerApi (see page 602) | Api class for managing Commerce (see page 1201) Customers in Cms (see page 7). |
| | ExchangeRateApi (see page 605) | Api class for managing commerce exchange rates in Cms (see page 7). |
| | InventoryApi (see page 608) | Api class for managing Commerce (see page 1201) Inventory in Cms (see page 7). |
| | OrderApi (see page 608) | Api class for managing Commerce (see page 1201) Orders in Cms (see page 7). |
| | PackageApi (see page 617) | API (see page 639) class used for managing shipping packages. |
| | ProductTypeApi (see page 619) | Api class for managing Commerce (see page 1201) product types in Cms (see page 7). |
| | RecommendationApi (see page 622) | Api class for managing entry reccomendations. |
| | RegionApi (see page 625) | Api class for managing regions in Cms (see page 7). |
| | ShippingMethodApi (see page 627) | Api class for managing shipping methods in CMS. |
| | TaxApi (see page 628) | Api class for managing Commerce (see page 1201) Tax rates in Cms (see page 7). |
| | TaxClassApi (see page 633) | Api class for managing Commerce (see page 1201) Tax Classes in Cms (see page 7). |
| | TaxTypeApi (see page 635) | Api class for managing Commerce (see page 1201) tax types in Cms (see page 7). |
| | WarehouseApi (see page 636) | This is class Ektron.Cms.Commerce.WarehouseApi. |

# AddressApi Class

Api class for managing Addresses in Cms (see page 7).

### Class Hierarchy

### C#

```
public class AddressApi : CommonApiBase;
```

### Visual Basic

```
Public Class AddressApi
Inherits CommonApiBase
```

### File

AddressApi.cs

# AddressApi Methods

The methods of the AddressApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 575) | This is the overview for the Add method overload. |
| | Delete (see page 576) | Deletes an address. |
| | GetItem (see page 576) | Retrieves an AddressData (see page 1206) object by providing its ID. |

| | | |
|---|---|---|
| | GetList (see page 576) | Retrieves a list of addresses for a given customer's ID. |
| | Update (see page 576) | Updates the supplied Address. |
| | UpdateOrderAddress (see page 577) | Updates an address associated with an order. This update only affects this specific address and does not affect the customer's actual saved address. |

## Add Method

This is the overview for the Add method overload.

### Description

The GetDefaultBasket method is overloaded.

- Add Method (AddressData) (see page 575) - Adds a new address based on information in an AddressData object.
- Add Method (AddressData, long) (see page 575) - Adds a new address based on information in an AddressData object and associates with a customer.

### Overload List

| | Name | Description |
|---|---|---|
| | AddressApi.Add (AddressData) (see page 575) | Adds a new address based on information in an AddressData (see page 1206) object. AddressData.Id will be populated with the newly created ID. |
| | AddressApi.Add (AddressData, long) (see page 575) | Adds a new address based on information in an AddressData (see page 1206) object and associates with a customer. AddressData.Id will be populated with the newly created ID. |

## AddressApi.Add Method (AddressData)

Adds a new address based on information in an AddressData (see page 1206) object. AddressData.Id will be populated with the newly created ID.

### C#

```
public void Add(AddressData addressData);
```

### Visual Basic

```
Public Function Add(addressData As AddressData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| addressData | The AddressData (see page 1206) object to add. |

## AddressApi.Add Method (AddressData, long)

Adds a new address based on information in an AddressData (see page 1206) object and associates with a customer. AddressData.Id will be populated with the newly created ID.

### C#

```
public void Add(AddressData addressData, long customerId);
```

### Visual Basic

```
Public Function Add(addressData As AddressData, customerId As long) As void
```

### Parameters

| Parameters | Description |
|---|---|
| addressData | The AddressData (see page 1206) object to add. |
| customerId | The ID of customer to which this address belongs. |

**Remarks**

Required: Name, AddressLine1, City, PostalCode, Phone.

## AddressApi.Delete Method

Deletes an address.

**C#**

```
public void Delete(long addressId);
```

**Visual Basic**

```
Public Function Delete(addressId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| addressId | The ID of the AddressData (see page 1206) object to delete. |

## AddressApi.GetItem Method

Retrieves an AddressData (see page 1206) object by providing its ID.

**C#**

```
public AddressData GetItem(long addressId);
```

**Visual Basic**

```
Public Function GetItem(addressId As long) As AddressData
```

**Parameters**

| Parameters | Description |
|---|---|
| addressId | The ID the of AddressData (see page 1206) object to retrieve. |

**Returns**

An AddressData (see page 1206) object.

## AddressApi.GetList Method

Retrieves a list of addresses for a given customer's ID.

**C#**

```
public List<AddressData> GetList(long customerId);
```

**Visual Basic**

```
Public Function GetList(customerId As long) As List<AddressData>
```

**Parameters**

| Parameters | Description |
|---|---|
| customerId | The ID of the customer for which to retrieve a list of addresses. |

**Returns**

An AddressData (see page 1206) object.

## AddressApi.Update Method

Updates the supplied Address.

**C#**

```
public void Update(AddressData addressData);
```

**Visual Basic**

```
Public Function Update(addressData As AddressData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| addressData | The ID of the AddressData (see page 1206) object to update. |

### AddressApi.UpdateOrderAddress Method

Updates an address associated with an order. This update only affects this specific address and does not affect the customer's actual saved address.

**C#**

```
public void UpdateOrderAddress(AddressData addressData, long orderId, bool isShipping, bool
isBilling);
```

**Visual Basic**

```
Public Function UpdateOrderAddress(addressData As AddressData, orderId As long, isShipping As
bool, isBilling As bool) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| addressData | Address info to update. |
| orderId | Id of order to update address for. |
| isShipping | If true, updates the shipping address for an order. |
| isBilling | If true, updates the billing address for an order. |

**Description**

///

## BasketApi Class

The API (see page 639) class used for managing Commerce (see page 1201) shopping carts (baskets).

**Class Hierarchy**

**C#**

```
public class BasketApi : CommonApiBase;
```

**Visual Basic**

```
Public Class BasketApi
Inherits CommonApiBase
```

**File**

BasketApi.cs

## BasketApi Methods

The methods of the BasketApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | CalculateBasket (see page 578) | This is the overview for the CalculateBasket method overload. |
| | CreateNewBasket (see page 579) | Creates a new cart for the supplied user ID. By setting the `openBasketYou` parameter to true, the new cart becomes the default open cart. |
| | Delete (see page 580) | Deletes a cart based on its ID. |
| | Empty (see page 580) | Deletes all the items in a cart based on the cart's ID. |
| | GetDefaultBasket (see page 580) | This is the overview for the GetDefaultBasket method overload. |
| | GetItem (see page 581) | Retrieves a single cart based on its ID. |
| | GetList (see page 582) | Retrieves a list saved carts for a user ID. |
| | SetDefaultBasket (see page 582) | This is the overview for the SetDefaultBasket method overload. |

## CalculateBasket Method

This is the overview for the CalculateBasket method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | BasketApi.CalculateBasket (Basket) (see page 578) | Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc. |
| | BasketApi.CalculateBasket (Basket, AddressData) (see page 578) | Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc. |
| | BasketApi.CalculateBasket (long) (see page 579) | Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc. |
| | BasketApi.CalculateBasket (long, AddressData) (see page 579) | Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc. |

## BasketApi.CalculateBasket Method (Basket)

Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.

**C#**

```
public BasketCalculator CalculateBasket(Basket basket);
```

**Visual Basic**

```
Public Function CalculateBasket(basket As Basket) As BasketCalculator
```

**Parameters**

| Parameters | Description |
|---|---|
| basket | Cart (see page 73) to calculate. |

## BasketApi.CalculateBasket Method (Basket, AddressData)

Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.

**C#**

```
public BasketCalculator CalculateBasket(Basket basket, AddressData shippingAddress);
```

**Visual Basic**

```
Public Function CalculateBasket(basket As Basket, shippingAddress As AddressData) As
BasketCalculator
```

**Parameters**

| Parameters | Description |
|---|---|
| basket | Cart (see page 73) to calculate. |
| shippingAddress | Shipping Address to use to calculate taxes and shipping. This Address is not saved with the basket and is only used for the returned calculations. |

## BasketApi.CalculateBasket Method (long)

Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.

**C#**

```
public BasketCalculator CalculateBasket(long basketId);
```

**Visual Basic**

```
Public Function CalculateBasket(basketId As long) As BasketCalculator
```

**Parameters**

| Parameters | Description |
|---|---|
| basketId | ID of the cart to calculate. |

## BasketApi.CalculateBasket Method (long, AddressData)

Runs calculations on the cart and returns BasketCalculator object. This method is used for calculating cart subtotals, shipping, taxes, coupons, etc.

**C#**

```
public BasketCalculator CalculateBasket(long basketId, AddressData shippingAddress);
```

**Visual Basic**

```
Public Function CalculateBasket(basketId As long, shippingAddress As AddressData) As
BasketCalculator
```

**Parameters**

| Parameters | Description |
|---|---|
| basketId | ID of the cart to calculate. |
| shippingAddress | Shipping Address to use to calculate taxes and shipping. This Address is not saved with the basket and is only used for the returned calculations. |

## BasketApi.CreateNewBasket Method

Creates a new cart for the supplied user ID. By setting the `openBasketYou` parameter to true, the new cart becomes the default open cart.

**C#**

```
public Basket CreateNewBasket(long userId, bool openBasket);
```

**Visual Basic**

```
Public Function CreateNewBasket(userId As long, openBasket As bool) As Basket
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user for which to create a cart. |

| | |
|---|---|
| openBasket | When set to true, the newly created cart becomes the default open cart. |

**Returns**

Newly created cart object.

## BasketApi.Delete Method

Deletes a cart based on its ID.

**C#**

```
public void Delete(long basketId);
```

**Visual Basic**

```
Public Function Delete(basketId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| basketId | The ID of cart to delete. |

## BasketApi.Empty Method

Deletes all the items in a cart based on the cart's ID.

**C#**

```
public void Empty(long basketId);
```

**Visual Basic**

```
Public Function Empty(basketId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| basketId | The ID of basket to empty. |

## GetDefaultBasket Method

This is the overview for the GetDefaultBasket method overload.

**Description**

The GetDefaultBasket method is overloaded.

- GetDefaultBasket Method () (see page 581) - Gets the default cart.
- GetDefaultBasket Method (long) (see page 581) - Gets the default cart based on a user's ID.
- GetDefaultBasket Method (string) (see page 581) - Gets the default cart based on a non logged in user's GUID shopping ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | BasketApi.GetDefaultBasket () (see page 581) | Gets the default cart. |
| | BasketApi.GetDefaultBasket (long) (see page 581) | Gets a default cart for a specified user. |
| | BasketApi.GetDefaultBasket (string) (see page 581) | Gets a default cart for a non logged in user based upon the shopper's GUID string. |

### BasketApi.GetDefaultBasket Method ()

Gets the default cart.

**C#**

```
public Basket GetDefaultBasket();
```

**Visual Basic**

```
Public Function GetDefaultBasket() As Basket
```

**Returns**

The default cart object.

### BasketApi.GetDefaultBasket Method (long)

Gets a default cart for a specified user.

**C#**

```
public Basket GetDefaultBasket(long userId);
```

**Visual Basic**

```
Public Function GetDefaultBasket(userId As long) As Basket
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of user for which to retrieve a basket. |

**Returns**

A default cart object for a specified.

### BasketApi.GetDefaultBasket Method (string)

Gets a default cart for a non logged in user based upon the shopper's GUID string.

**C#**

```
public Basket GetDefaultBasket(string shopperId);
```

**Visual Basic**

```
Public Function GetDefaultBasket(shopperId As string) As Basket
```

**Parameters**

| Parameters | Description |
|---|---|
| shopperId | The GUID ID for the non logged in shopper. |

**Returns**

a default cart object for a non logged in user.

### BasketApi.GetItem Method

Retrieves a single cart based on its ID.

**C#**

```
public Basket GetItem(long id);
```

**Visual Basic**

```
Public Function GetItem(id As long) As Basket
```

### Parameters

| Parameters | Description |
|---|---|
| id | ID of cart to retrieve. |

### Returns

A cart object.

## BasketApi.GetList Method

Retrieves a list saved carts for a user ID.

### C#

```
public List<Basket> GetList(long userId);
```

### Visual Basic

```
Public Function GetList(userId As long) As List<Basket>
```

### Parameters

| Parameters | Description |
|---|---|
| userId | ID of user for which to retrieve a list of saved carts. |

### Returns

A list of saved cart objects.

## SetDefaultBasket Method

This is the overview for the SetDefaultBasket method overload.

### Description

The SetDefaultBasket method is overloaded.

- SetDefaultBasket Method (long) (see page 582) - Sets the logged in user's default cart based on the cart's ID.
- SetDefaultBasket Method (long, long) (see page 583) - Sets the default cart based on the cart's ID and user's ID.

### Overload List

| | Name | Description |
|---|---|---|
| | BasketApi.SetDefaultBasket (long) (see page 582) | Sets the default cart for the current user. |
| | BasketApi.SetDefaultBasket (long, long) (see page 583) | Sets the default cart for a user. |

## BasketApi.SetDefaultBasket Method (long)

Sets the default cart for the current user.

### C#

```
public void SetDefaultBasket(long basketId);
```

### Visual Basic

```
Public Function SetDefaultBasket(basketId As long) As void
```

### Parameters

| Parameters | Description |
|---|---|
| basketId | The ID of the cart to set as the default. |

### BasketApi.SetDefaultBasket Method (long, long)

Sets the default cart for a user.

**C#**

```
public void SetDefaultBasket(long basketId, long userId);
```

**Visual Basic**

```
Public Function SetDefaultBasket(basketId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| basketId | The ID of the cart to set as the default. |
| userId | The ID of the user setting the default cart. |

## CatalogEntryApi Class

API (see page 639) class used for managing catalogs entries for ecommerce.

**Class Hierarchy**

**C#**

```
public class CatalogEntryApi : CommonApiBase;
```

**Visual Basic**

```
Public Class CatalogEntryApi
Inherits CommonApiBase
```

**File**

CatalogApi.cs

## CatalogEntryApi Methods

The methods of the CatalogEntryApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 584) | Adds and publishes a new catalog entry based on an EntryData (see page 1264) object. |
| | AddAndCheckIn (see page 584) | Adds and checks-in a new catalog entry based on an EntryData (see page 1264) object. |
| | AddAndPublish (see page 584) | Adds and publishes a new catalog entry based on an EntryData (see page 1264) object. |
| | AddAndSave (see page 584) | Adds and saves a new catalog entry based on an EntryData (see page 1264) object.<br>**Note:** The catalog entry is not checked in or published. |
| | Checkout (see page 585) | Checks out a catalog entry based on its ID. |
| | Delete (see page 585) | Deletes a catalog entry based on its ID. |
| | GetAttributeEntryList (see page 585) | Retrieves a list of entrydata objects based upon EntryAttributeCriteria. |
| | GetItem (see page 585) | This is the overview for the GetItem method overload. |
| | GetItemEdit (see page 586) | Gets the saved or checked in version of the catalog entry. If status is A, then gets the currently published catalog entry. Does NOT check out the catalog entry automatically. |

| | | |
|---|---|---|
| | GetList (see page 587) | Retrieves a list of EntryData (see page 1264) objects based upon the supplied criteria. |
| | GetTaxonomyEntryList (see page 587) | Retrieves a list of EntryData (see page 1264) objects in the supplied taxonomy based upon the supplied criteria. |
| | Restore (see page 587) | Restores a previous version of a catalog entry based on its ID and the version ID to restore. |
| | UndoCheckOut (see page 587) | Undoes a check out for a catalog entry based on its ID. **Note:** Any changes made to saved version will be lost. |
| | Update (see page 588) | Saves an existing catalog entry. |
| | UpdateAndCheckIn (see page 588) | Saves and checks-in an existing catalog entry. |
| | UpdateAndPublish (see page 588) | Saves and submits an existing catalog entry for publish. |

## CatalogEntryApi.Add Method

Adds and publishes a new catalog entry based on an EntryData (see page 1264) object.

**C#**

```
public void Add(EntryData entry);
```

**Visual Basic**

```
Public Function Add(entry As EntryData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entry | The EntryData (see page 1264) object to add. |

## CatalogEntryApi.AddAndCheckIn Method

Adds and checks-in a new catalog entry based on an EntryData (see page 1264) object.

**C#**

```
public void AddAndCheckIn(EntryData entry);
```

**Visual Basic**

```
Public Function AddAndCheckIn(entry As EntryData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entry | The EntryData (see page 1264) object to add and checkin. |

## CatalogEntryApi.AddAndPublish Method

Adds and publishes a new catalog entry based on an EntryData (see page 1264) object.

**C#**

```
public void AddAndPublish(EntryData entry);
```

**Visual Basic**

```
Public Function AddAndPublish(entry As EntryData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entry | The EntryData (see page 1264) object to add and publish. |

## CatalogEntryApi.AddAndSave Method

Adds and saves a new catalog entry based on an EntryData (see page 1264) object.

**Note:** The catalog entry is not checked in or published.

**C#**

```
public void AddAndSave(EntryData entry);
```

**Visual Basic**

```
Public Function AddAndSave(entry As EntryData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entry | The EntryData (see page 1264) object to add and save. |

### CatalogEntryApi.Checkout Method

Checks out a catalog entry based on its ID.

**C#**

```
public void Checkout(long entryId);
```

**Visual Basic**

```
Public Function Checkout(entryId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entryId | The ID of catalog entry to check out. |

### CatalogEntryApi.Delete Method

Deletes a catalog entry based on its ID.

**C#**

```
public void Delete(long entryId);
```

**Visual Basic**

```
Public Function Delete(entryId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entryId | The ID of entry to delete. |

### CatalogEntryApi.GetAttributeEntryList Method

Retrieves a list of entrydata objects based upon EntryAttributeCriteria.

**C#**

```
public List<EntryData> GetAttributeEntryList(EntryAttributeCriteria criteria);
```

**Visual Basic**

```
Public Function GetAttributeEntryList(criteria As EntryAttributeCriteria) As List<EntryData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | EntryAttributeCriteria by which to retrieve entry list. |

### GetItem Method

This is the overview for the GetItem method overload.

**Description**

The GetItem method is overloaded.

- GetItem Method (long) (see page 586) - Retrieves a single catalog entry in the current language based on the entry's ID.
- GetItem Method (long, long) (see page 586) - Retrieves a single catalog entry in the supplied language based on the catalog entry's ID and language ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | CatalogEntryApi.GetItem (long) (see page 586) | Retrieves a single catalog entry in the current language based on the entry's ID. |
| | CatalogEntryApi.GetItem (long, long) (see page 586) | Retrieves a single catalog entry in the supplied language based on the catalog entry's ID and language ID. |

## CatalogEntryApi.GetItem Method (long)

Retrieves a single catalog entry in the current language based on the entry's ID.

**C#**

```
public EntryData GetItem(long entryId);
```

**Visual Basic**

```
Public Function GetItem(entryId As long) As EntryData
```

**Parameters**

| Parameters | Description |
|---|---|
| entryId | The ID of catalog entry to retrieve. |

## CatalogEntryApi.GetItem Method (long, long)

Retrieves a single catalog entry in the supplied language based on the catalog entry's ID and language ID.

**C#**

```
public EntryData GetItem(long entryId, long entryLanguage);
```

**Visual Basic**

```
Public Function GetItem(entryId As long, entryLanguage As long) As EntryData
```

**Parameters**

| Parameters | Description |
|---|---|
| entryId | The ID of the catalog entry to retrieve. |
| entryLanguage | The ID of the language for which to retrieve a catalog entry. |

## CatalogEntryApi.GetItemEdit Method

Gets the saved or checked in version of the catalog entry. If status is A, then gets the currently published catalog entry. Does NOT check out the catalog entry automatically.

**C#**

```
public EntryData GetItemEdit(long entryId, long entryLanguage, bool checkOut);
```

**Visual Basic**

```
Public Function GetItemEdit(entryId As long, entryLanguage As long, checkOut As bool) As
EntryData
```

**Parameters**

| Parameters | Description |
|---|---|
| entryId | The ID of the catalog entry. |

### CatalogEntryApi.GetList Method

Retrieves a list of EntryData (see page 1264) objects based upon the supplied criteria.

**C#**

```
public List<EntryData> GetList(Criteria<EntryProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<EntryProperty>) As List<EntryData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | The criteria for retrieving EntryData (see page 1264). |

### CatalogEntryApi.GetTaxonomyEntryList Method

Retrieves a list of EntryData (see page 1264) objects in the supplied taxonomy based upon the supplied criteria.

**C#**

```
public List<EntryData> GetTaxonomyEntryList(long taxonomyId, int depth,
Criteria<EntryProperty> criteria);
```

**Visual Basic**

```
Public Function GetTaxonomyEntryList(taxonomyId As long, depth As Integer, criteria As
Criteria<EntryProperty>) As List<EntryData>
```

**Parameters**

| Parameters | Description |
|---|---|
| taxonomyId | The ID of taxonomy for which to reterieve EntryData (see page 1264) objects. |
| criteria | The criteria for retrieving EntryData (see page 1264). |

### CatalogEntryApi.Restore Method

Restores a previous version of a catalog entry based on its ID and the version ID to restore.

**C#**

```
public void Restore(long id, long versionId);
```

**Visual Basic**

```
Public Function Restore(id As long, versionId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| id | The ID of the catalog entry to Restore. |
| versionId | Version Id to restore. |

### CatalogEntryApi.UndoCheckOut Method

Undoes a check out for a catalog entry based on its ID. **Note:** Any changes made to saved version will be lost.

**C#**

```
public void UndoCheckOut(long entryId);
```

**Visual Basic**

```
Public Function UndoCheckOut(entryId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entryId | The ID of the catalog entry being reverted. |

### CatalogEntryApi.Update Method

Saves an existing catalog entry.

**C#**

```
public void Update(EntryData entry);
```

**Visual Basic**

```
Public Function Update(entry As EntryData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entry | The EntryData (see page 1264) object to save. |

### CatalogEntryApi.UpdateAndCheckIn Method

Saves and checks-in an existing catalog entry.

**C#**

```
public void UpdateAndCheckIn(EntryData entry);
```

**Visual Basic**

```
Public Function UpdateAndCheckIn(entry As EntryData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entry | The EntryData (see page 1264) object to save and check-in. |

### CatalogEntryApi.UpdateAndPublish Method

Saves and submits an existing catalog entry for publish.

**C#**

```
public void UpdateAndPublish(EntryData entry);
```

**Visual Basic**

```
Public Function UpdateAndPublish(entry As EntryData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entry | The EntryData (see page 1264) object to save and publish. |

## ContractApi Class

Api class for managing Commerce (see page 1201) Contracts in Cms (see page 7).

**Class Hierarchy**

**C#**

```
public class ContractApi : CommonApiBase;
```

**Visual Basic**

```
Public Class ContractApi
Inherits CommonApiBase
```

**File**

ContractApi.cs

## ContractApi Methods

The methods of the ContractApi class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Add (see page 589) | Saves a new ContractData (see page 1237) object. Returns the newly created Id. |
| | Delete (see page 589) | Deletes a ContractData (see page 1237) object. |
| | GetItem (see page 589) | Retrieves a ContractData (see page 1237) object by Id. |
| | GetList (see page 590) | Retrieves list of contract objects. |
| | Update (see page 590) | Updates the supplied ContractData (see page 1237) object. |

### ContractApi.Add Method

Saves a new ContractData (see page 1237) object. Returns the newly created Id.

**C#**

```
public void Add(ContractData contract);
```

**Visual Basic**

```
Public Function Add(contract As ContractData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| contract | ContractData (see page 1237) to save. |

### ContractApi.Delete Method

Deletes a ContractData (see page 1237) object.

**C#**

```
public void Delete(long contractId);
```

**Visual Basic**

```
Public Function Delete(contractId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| contractId | Id of contract to delete |

### ContractApi.GetItem Method

Retrieves a ContractData (see page 1237) object by Id.

**C#**

```
public ContractData GetItem(long contractId);
```

**Visual Basic**

```
Public Function GetItem(contractId As long) As ContractData
```

**Parameters**

| Parameters | Description |
|---|---|
| contractId | Id of ContractData (see page 1237) object to retrieve. |

### ContractApi.GetList Method

Retrieves list of contract objects.

**C#**

```
public List<ContractData> GetList(Criteria<ContractProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<ContractProperty>) As List<ContractData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | criteria for contract retrieval |

### ContractApi.Update Method

Updates the supplied ContractData (see page 1237) object.

**C#**

```
public void Update(ContractData contract);
```

**Visual Basic**

```
Public Function Update(contract As ContractData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| contract | ContractData (see page 1237) object to update. |

## CountryApi Class

API (see page 639) class for managing countries in CMS400.NET.

**Class Hierarchy**

**C#**

```
public class CountryApi : CommonApiBase;
```

**Visual Basic**

```
Public Class CountryApi
Inherits CommonApiBase
```

**File**

CountryApi.cs

## CountryApi Methods

The methods of the CountryApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 591) | Adds a new country to CMS400.NET based on the CountryData (see page 1239) object. The Country.Id property will be populated with the new Country's ID. |
| | CanDelete (see page 591) | Checks to see if a country can be deleted. A country cannot be deleted if a region or address uses the country. |
| | Delete (see page 591) | Deletes a country from the CMS400.NET based on its ID. |
| | GetItem (see page 592) | Retrieves a country by its ID. |
| | GetList (see page 592) | Retrieves a list of countries based on filters set in the `Criteria` property. |
| | Update (see page 592) | Updates an existing country in CMS400.NET based on the CountryData (see page 1239) object. |

## CountryApi.Add Method

Adds a new country to CMS400.NET based on the CountryData (see page 1239) object. The Country.Id property will be populated with the new Country's ID.

**C#**

```
public void Add(CountryData countryData);
```

**Visual Basic**

```
Public Function Add(countryData As CountryData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| countryData | The CountryData (see page 1239) object to add. |

## CountryApi.CanDelete Method

Checks to see if a country can be deleted. A country cannot be deleted if a region or address uses the country.

**C#**

```
public bool CanDelete(int countryId, out ValidationResults result);
```

**Visual Basic**

```
Public Function CanDelete(countryId As Integer, result As out ValidationResults) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| countryId | The ID of the country to delete. |
| result | ValidationResult returned containing messages if country cannot be deleted. |

**Returns**

Returns true if country can safely be deleted.

## CountryApi.Delete Method

Deletes a country from the CMS400.NET based on its ID.

**C#**

```
public void Delete(int id);
```

**Visual Basic**

```
Public Function Delete(id As Integer) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| id | The ID of country to delete. |

### CountryApi.GetItem Method

Retrieves a country by its ID.

**C#**

```
public CountryData GetItem(int id);
```

**Visual Basic**

```
Public Function GetItem(id As Integer) As CountryData
```

**Parameters**

| Parameters | Description |
|---|---|
| id | The ID of country to retrieve. |

**Returns**

A CountryData (see page 1239) object.

### CountryApi.GetList Method

Retrieves a list of countries based on filters set in the `Criteria` property.

**C#**

```
public List<CountryData> GetList(Criteria<CountryProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<CountryProperty>) As List<CountryData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | Criteria by which to filter the list of countries being retrieved. |

**Returns**

A list of countries meeting criteria filters.

### CountryApi.Update Method

Updates an existing country in CMS400.NET based on the CountryData (see page 1239) object.

**C#**

```
public void Update(CountryData countryData);
```

**Visual Basic**

```
Public Function Update(countryData As CountryData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| countryData | The CountryData (see page 1239) object to save. |

## CouponApi Class

Api class for managing Commerce (see page 1201) Coupons in Cms (see page 7).

**Class Hierarchy**

**C#**

```
public class CouponApi : CommonApiBase;
```

**Visual Basic**

```
Public Class CouponApi
Inherits CommonApiBase
```

**File**

CouponApi.cs

## CouponApi Methods

The methods of the CouponApi class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Add (see page 593) | Saves an new couponData object. CouponData.Id will be populated with the newly created Id. |
| | AddCouponToCatalog (see page 594) | Applys a coupon to a catalog. |
| | AddCouponToProduct (see page 594) | Applys a coupon to a product. |
| | AddCouponToTaxonomy (see page 594) | Applys a coupon to a catalog. |
| | CalculateBasketCoupons (see page 595) | Runs coupon calculations on basket and returns BasketCalculatorData (see page 1220) object. |
| | Deactivate (see page 595) | Deactivate a coupondata object. |
| | Delete (see page 595) | Deletes a coupondata object. |
| | DeleteCouponProducts (see page 595) | Deletes all the products a coupon is applied to accept the products included in the ExclusionList. |
| | GetCatalogList (see page 596) | retrieves list of CouponEntryData (see page 1247) for catalogs with coupons applied to them. |
| | GetCouponsAppliedToBasket (see page 596) | Retrievies a list of coupons applied to a basket. |
| | GetItem (see page 596) | Retrieves a CouponData (see page 1243) object by Id. |
| | GetList (see page 596) | Retrieves list of couponData objects for a given customer. |
| | GetProductList (see page 597) | This is the overview for the GetProductList method overload. |
| | GetTaxonomyList (see page 597) | retrieves list of CouponEntryData (see page 1247) for taxonomies with coupons applied to them. |
| | IsCouponAppliedToProduct (see page 598) | Returns true if the coupon is applied to the catalog entry. |
| | IsCouponUsedForBasket (see page 598) | Checks if coupon has been applied to a basket. |
| | IsCouponUsedForOrder (see page 598) | Checks if coupon has been used by an order. |
| | SaveCouponApplications (see page 598) | Saves all the applications (taxonomy, catalog, product) for a given coupon. This is assumed to be an all inclusive lists. Any existing coupon applications that are not included will be deleted. |
| | Update (see page 599) | Updates the supplied CouponData (see page 1243) object. |
| | ValidateCoupon (see page 599) | This is the overview for the ValidateCoupon method overload. |

### CouponApi.Add Method

Saves an new couponData object. CouponData.Id will be populated with the newly created Id.

**C#**

```
public void Add(CouponData couponData);
```

**Visual Basic**

```
Public Function Add(couponData As CouponData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| couponData | CouponData (see page 1243) to save. |

## CouponApi.AddCouponToCatalog Method

Applys a coupon to a catalog.

**C#**

```
public void AddCouponToCatalog(long couponId, long catalogId, bool includeCatalog);
```

**Visual Basic**

```
Public Function AddCouponToCatalog(couponId As long, catalogId As long, includeCatalog As
bool) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon to apply. |
| catalogId | Id of catalog to apply coupon to. |
| includeCatalog | if true, include catalog in coupon application. Else excludes catalog from coupon. |

## CouponApi.AddCouponToProduct Method

Applys a coupon to a product.

**C#**

```
public void AddCouponToProduct(long couponId, long productId, bool includeProduct);
```

**Visual Basic**

```
Public Function AddCouponToProduct(couponId As long, productId As long, includeProduct As
bool) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon to apply. |
| productId | Id of product to apply coupon to. |
| includeProduct | if true, include product in coupon application. Else excludes product from coupon. |

## CouponApi.AddCouponToTaxonomy Method

Applys a coupon to a catalog.

**C#**

```
public void AddCouponToTaxonomy(long couponId, long taxonomyId, bool includeTaxonomy);
```

**Visual Basic**

```
Public Function AddCouponToTaxonomy(couponId As long, taxonomyId As long, includeTaxonomy As
bool) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon to apply. |
| taxonomyId | Id of taxonomy to apply coupon to. |
| includeTaxonomy | if true, include txonomy in coupon application. Else excludes taxonomy from coupon. |

### CouponApi.CalculateBasketCoupons Method

Runs coupon calculations on basket and returns BasketCalculatorData (see page 1220) object.

**C#**

```
public BasketCalculatorData CalculateBasketCoupons(long basketId);
```

**Visual Basic**

```
Public Function CalculateBasketCoupons(basketId As long) As BasketCalculatorData
```

**Parameters**

| Parameters | Description |
|---|---|
| basketId | Id of basket to calculate. |

**Returns**

BasketCalculatorData (see page 1220) with coupon calculations complete.

### CouponApi.Deactivate Method

Deactivate a coupondata object.

**C#**

```
public void Deactivate(long couponId);
```

**Visual Basic**

```
Public Function Deactivate(couponId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of couponData to deactivate |

### CouponApi.Delete Method

Deletes a coupondata object.

**C#**

```
public void Delete(long couponId);
```

**Visual Basic**

```
Public Function Delete(couponId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of couponData to delete |

### CouponApi.DeleteCouponProducts Method

Deletes all the products a coupon is applied to accept the products included in the ExclusionList.

**C#**

```
public void DeleteCouponProducts(long couponId, IEnumerable<long> exclusionIdList);
```

**Visual Basic**

```
Public Function DeleteCouponProducts(couponId As long, exclusionIdList As IEnumerable<long>)
As void
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon to remove from entries. |
| exclusionIdList | List if entry Ids to exclude. |

### CouponApi.GetCatalogList Method

retrieves list of CouponEntryData (see page 1247) for catalogs with coupons applied to them.

**C#**

```
public List<CouponEntryData> GetCatalogList(Criteria<CouponEntryProperty> criteria);
```

**Visual Basic**

```
Public Function GetCatalogList(criteria As Criteria<CouponEntryProperty>) As
List<CouponEntryData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | criteria for formulating coupon retrieval |

### CouponApi.GetCouponsAppliedToBasket Method

Retrievies a list of coupons applied to a basket.

**C#**

```
public List<CouponData> GetCouponsAppliedToBasket(long basketId);
```

**Visual Basic**

```
Public Function GetCouponsAppliedToBasket(basketId As long) As List<CouponData>
```

**Parameters**

| Parameters | Description |
|---|---|
| basketId | Id of basket to retrieve coupons for. |

**Returns**

List of CouponData (see page 1243) objects.

### CouponApi.GetItem Method

Retrieves a CouponData (see page 1243) object by Id.

**C#**

```
public CouponData GetItem(long couponId);
```

**Visual Basic**

```
Public Function GetItem(couponId As long) As CouponData
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupondata object to retrieve. |

### CouponApi.GetList Method

Retrieves list of couponData objects for a given customer.

**C#**

```
public List<CouponData> GetList(Criteria<CouponProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<CouponProperty>) As List<CouponData>
```

**Parameters**

| Parameters | Description |
|---|---|
| customerId | Id of customer to retrieve coupones for. |

## GetProductList Method

This is the overview for the GetProductList method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | CouponApi.GetProductList (Criteria<CouponEntryProperty>) (see page 597) | retrieves list of ProductCouponEntryData (see page 1349) for products with coupons available. |
| | CouponApi.GetProductList (long) (see page 597) | retrieves list of ProductCouponEntryData (see page 1349) for each product the coupon applies to. |

## CouponApi.GetProductList Method (Criteria<CouponEntryProperty>)

retrieves list of ProductCouponEntryData (see page 1349) for products with coupons available.

**C#**

```
public List<ProductCouponEntryData> GetProductList(Criteria<CouponEntryProperty> criteria);
```

**Visual Basic**

```
Public Function GetProductList(criteria As Criteria<CouponEntryProperty>) As
List<ProductCouponEntryData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | criteria for formulating product retrieval |

## CouponApi.GetProductList Method (long)

retrieves list of ProductCouponEntryData (see page 1349) for each product the coupon applies to.

**C#**

```
public List<ProductCouponEntryData> GetProductList(long couponId);
```

**Visual Basic**

```
Public Function GetProductList(couponId As long) As List<ProductCouponEntryData>
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon to search products for. |

## CouponApi.GetTaxonomyList Method

retrieves list of CouponEntryData (see page 1247) for taxonomies with coupons applied to them.

**C#**

```
public List<CouponEntryData> GetTaxonomyList(Criteria<CouponEntryProperty> criteria);
```

**Visual Basic**

```
Public Function GetTaxonomyList(criteria As Criteria<CouponEntryProperty>) As
List<CouponEntryData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | criteria for formulating coupon retrieval |

### CouponApi.IsCouponAppliedToProduct Method

Returns true if the coupon is applied to the catalog entry.

**C#**

```
public bool IsCouponAppliedToProduct(long couponId, long entryId);
```

**Visual Basic**

```
Public Function IsCouponAppliedToProduct(couponId As long, entryId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon to to check. |
| entryId | Id of entry to check for coupon. |

### CouponApi.IsCouponUsedForBasket Method

Checks if coupon has been applied to a basket.

**C#**

```
public bool IsCouponUsedForBasket(long couponId);
```

**Visual Basic**

```
Public Function IsCouponUsedForBasket(couponId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon. |

### CouponApi.IsCouponUsedForOrder Method

Checks if coupon has been used by an order.

**C#**

```
public bool IsCouponUsedForOrder(long couponId);
```

**Visual Basic**

```
Public Function IsCouponUsedForOrder(couponId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon. |

### CouponApi.SaveCouponApplications Method

Saves all the applications (taxonomy, catalog, product) for a given coupon. This is assumed to be an all inclusive lists. Any existing coupon applications that are not included will be deleted.

**C#**

```
public void SaveCouponApplications(long couponId, List<CouponEntryData> couponEntryList);
```

**Visual Basic**

```
Public Function SaveCouponApplications(couponId As long, couponEntryList As
List<CouponEntryData>) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon to update. |
| couponEntryList | List of all CouponEntry applications for the given coupon. |

## CouponApi.Update Method

Updates the supplied CouponData (see page 1243) object.

**C#**

```
public void Update(CouponData couponData);
```

**Visual Basic**

```
Public Function Update(couponData As CouponData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| couponData | CouponData (see page 1243) object to update. |

## ValidateCoupon Method

This is the overview for the ValidateCoupon method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | CouponApi.ValidateCoupon (long) (see page 599) | Validates that the coupon has not expired and is still available for use. |
| | CouponApi.ValidateCoupon (long, long) (see page 599) | Validates that the coupon has not expired and is still available for use and can be used on the supplied product. |

## CouponApi.ValidateCoupon Method (long)

Validates that the coupon has not expired and is still available for use.

**C#**

```
public bool ValidateCoupon(long couponId);
```

**Visual Basic**

```
Public Function ValidateCoupon(couponId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon to validate. |

## CouponApi.ValidateCoupon Method (long, long)

Validates that the coupon has not expired and is still available for use and can be used on the supplied product.

**C#**

```
public bool ValidateCoupon(long couponId, long productId);
```

**Visual Basic**

```
Public Function ValidateCoupon(couponId As long, productId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| couponId | Id of coupon to validate. |
| productId | Id of Product (see page 412) to validate for use with coupon. |

# CurrencyApi Class

Api class for managing commerce currencies in Cms (see page 7).

**Class Hierarchy**

**C#**

```
public class CurrencyApi : CommonApiBase;
```

**Visual Basic**

```
Public Class CurrencyApi
Inherits CommonApiBase
```

**File**

CurrencyApi.cs

## CurrencyApi Methods

The methods of the CurrencyApi class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Add (see page 600) | Adds a new Currency to Cms (see page 7). The Currency.Id property will be populated with the new Currency's Id. |
| | CanDelete (see page 601) | Checks to see if acurrency can be deleted. |
| | Delete (see page 601) | Deletes a Currency from the Cms (see page 7). |
| | GetItem (see page 601) | Retrieves a currency by currency Id. |
| | GetList (see page 601) | Retrieves a list of currencies. |
| | Update (see page 602) | Updates an existing Currency in Cms (see page 7). |

### CurrencyApi.Add Method

Adds a new Currency to Cms (see page 7). The Currency.Id property will be populated with the new Currency's Id.

**C#**

```
public void Add(CurrencyData currencyData);
```

**Visual Basic**

```
Public Function Add(currencyData As CurrencyData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| currencyData | CurrencyData (see page 1250) object to add. |

### CurrencyApi.CanDelete Method

Checks to see if acurrency can be deleted.

**C#**

```
public bool CanDelete(int currencyId, out ValidationResults result);
```

**Visual Basic**

```
Public Function CanDelete(currencyId As Integer, result As out ValidationResults) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| currencyId | Id of currency to delete. |
| result | ValidationResult returned containing errors if currency cannot be deleted. |

**Returns**

true if currency can safely be deleted.

### CurrencyApi.Delete Method

Deletes a Currency from the Cms (see page 7).

**C#**

```
public void Delete(int id);
```

**Visual Basic**

```
Public Function Delete(id As Integer) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of Currency to delete. |

### CurrencyApi.GetItem Method

Retrieves a currency by currency Id.

**C#**

```
public CurrencyData GetItem(int id);
```

**Visual Basic**

```
Public Function GetItem(id As Integer) As CurrencyData
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of currency to retrieve. |

### CurrencyApi.GetList Method

Retrieves a list of currencies.

**C#**

```
public List<CurrencyData> GetList(Criteria<CurrencyProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<CurrencyProperty>) As List<CurrencyData>
```

### Parameters

| Parameters | Description |
|---|---|
| criteria | Criteria by which to filter currency being retrieved. |

### Returns

List of currencies meeting criteria filters.

### CurrencyApi.Update Method

Updates an existing Currency in Cms (see page 7).

**C#**

```
public void Update(CurrencyData currencyData);
```

**Visual Basic**

```
Public Function Update(currencyData As CurrencyData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| currencyData | CurrencyData (see page 1250) object to save. |

## CustomerApi Class

Api class for managing Commerce (see page 1201) Customers in Cms (see page 7).

### Class Hierarchy

**C#**

```
public class CustomerApi : CommonApiBase;
```

**Visual Basic**

```
Public Class CustomerApi
Inherits CommonApiBase
```

### File

CustomerApi.cs

## CustomerApi Methods

The methods of the CustomerApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 603) | This is the overview for the Add method overload. |
| | Delete (see page 603) | Deletes a customer from the Cms (see page 7). |
| | GetItem (see page 604) | Retrieves a customer. |
| | GetList (see page 604) | Returns a list of customers based upon supplied criteria. |
| | GetOrderList (see page 604) | Returns the list of orders for a customer. |
| | GetOrderTotal (see page 605) | Returns the number of orders for a customer. |
| | Update (see page 605) | Updates an existing customer. |

## Add Method

This is the overview for the Add method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | CustomerApi.Add (CustomerData, AddressData) (see page 603) | Adds a new customer with a billing address. The Customer.Id property will be populated with the new Customer's Id. |
| | CustomerApi.Add (CustomerData, AddressData, AddressData) (see page 603) | Adds a new customer with seperate billing and shipping addresses. The Customer.Id property will be populated with the new Customer's Id. |

## CustomerApi.Add Method (CustomerData, AddressData)

Adds a new customer with a billing address. The Customer.Id property will be populated with the new Customer's Id.

### C#

```
public void Add(CustomerData customerData, AddressData billingAddress);
```

### Visual Basic

```
Public Function Add(customerData As CustomerData, billingAddress As AddressData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| customerData | customer data to save. |
| billingAddress | Billing address of customer. Also serves as shipping address. |

## CustomerApi.Add Method (CustomerData, AddressData, AddressData)

Adds a new customer with seperate billing and shipping addresses. The Customer.Id property will be populated with the new Customer's Id.

### C#

```
public void Add(CustomerData customerData, AddressData billingAddress, AddressData
shippingAddress);
```

### Visual Basic

```
Public Function Add(customerData As CustomerData, billingAddress As AddressData,
shippingAddress As AddressData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| customerData | customer data to save. |
| billingAddress | Billing address of customer. |
| shippingAddress | Shipping address of customer. |

## CustomerApi.Delete Method

Deletes a customer from the Cms (see page 7).

**C#**

```
public void Delete(long customerId);
```

**Visual Basic**

```
Public Function Delete(customerId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| customerId | Id of cusotmer to delete. |

### CustomerApi.GetItem Method

Retrieves a customer.

**C#**

```
public CustomerData GetItem(long customerId);
```

**Visual Basic**

```
Public Function GetItem(customerId As long) As CustomerData
```

**Parameters**

| Parameters | Description |
|---|---|
| customerId | Id of customer to retrieve. |

### CustomerApi.GetList Method

Returns a list of customers based upon supplied criteria.

**C#**

```
public List<CustomerData> GetList(Criteria<CustomerProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<CustomerProperty>) As List<CustomerData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | criteria by which to retrieve customer objects. |

**Returns**

List of CustomerData (see page 1255)

### CustomerApi.GetOrderList Method

Returns the list of orders for a customer.

**C#**

```
public List<OrderData> GetOrderList(long customerId);
```

**Visual Basic**

```
Public Function GetOrderList(customerId As long) As List<OrderData>
```

**Parameters**

| Parameters | Description |
|---|---|
| customerId | Id of customer to retrieve orders for. |

### CustomerApi.GetOrderTotal Method

Returns the number of orders for a customer.

**C#**

```
public long GetOrderTotal(long customerId);
```

**Visual Basic**

```
Public Function GetOrderTotal(customerId As long) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| customerId | Id of customer to get order total for. |

### CustomerApi.Update Method

Updates an existing customer.

**C#**

```
public void Update(CustomerData customerData);
```

**Visual Basic**

```
Public Function Update(customerData As CustomerData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| customerData | customer data to save. |

## ExchangeRateApi Class

Api class for managing commerce exchange rates in Cms (see page 7).

**Class Hierarchy**

**C#**

```
public class ExchangeRateApi : CommonApiBase;
```

**Visual Basic**

```
Public Class ExchangeRateApi
Inherits CommonApiBase
```

**File**

ExchangeRateApi.cs

## ExchangeRateApi Methods

The methods of the ExchangeRateApi class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Add (see page 606) | Adds a new Exchange Rate to the Cms (see page 7). The ExchangeRate.Id property will be populated with the new ExchangeRate's Id. |
| | ConvertPrice (see page 606) | Converts a price from the base currency to the exchange currency. |
| | Delete (see page 606) | Deletes an Exchange Rate from the Cms (see page 7). |

| | | |
|---|---|---|
| | GetCurrentExchangeRate (see page 606) | Retrieves the current exchange rate by exchange currency Id. |
| | GetCurrentList (see page 607) | Retrieves a list of current exchange rates. |
| | GetList (see page 607) | Retrieves a list of exchange rates. |
| | Update (see page 607) | Updates an existing Exchange Rate in Cms (see page 7). |

### ExchangeRateApi.Add Method

Adds a new Exchange Rate to the Cms (see page 7). The ExchangeRate.Id property will be populated with the new ExchangeRate's Id.

**C#**

```
public void Add(ExchangeRateData exchangeRateData);
```

**Visual Basic**

```
Public Function Add(exchangeRateData As ExchangeRateData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| exchangeRateData | ExchangeRateData (see page 1281) object to add. |

### ExchangeRateApi.ConvertPrice Method

Converts a price from the base currency to the exchange currency.

**C#**

```
public decimal ConvertPrice(decimal basePrice, int exchangeCurrencyId);
```

**Visual Basic**

```
Public Function ConvertPrice(basePrice As decimal, exchangeCurrencyId As Integer) As decimal
```

**Parameters**

| Parameters | Description |
|---|---|
| basePrice | price in the default currency. |
| exchangeCurrencyId | |

### ExchangeRateApi.Delete Method

Deletes an Exchange Rate from the Cms (see page 7).

**C#**

```
public void Delete(ExchangeRateData exchangeRateData);
```

**Visual Basic**

```
Public Function Delete(exchangeRateData As ExchangeRateData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| exchangeRateData | ExchangeRate to delete. |

### ExchangeRateApi.GetCurrentExchangeRate Method

Retrieves the current exchange rate by exchange currency Id.

**C#**

```
public ExchangeRateData GetCurrentExchangeRate(int exchangeCurrencyId);
```

**Visual Basic**

```
Public Function GetCurrentExchangeRate(exchangeCurrencyId As Integer) As ExchangeRateData
```

**Parameters**

| Parameters | Description |
|---|---|
| CurrencyId | Id of exchangeRate to retrieve. |

### ExchangeRateApi.GetCurrentList Method

Retrieves a list of current exchange rates.

**C#**

```
public List<ExchangeRateData> GetCurrentList(Criteria<ExchangeRateProperty> criteria);
```

**Visual Basic**

```
Public Function GetCurrentList(criteria As Criteria<ExchangeRateProperty>) As
List<ExchangeRateData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | Criteria by which to filter exchange rates being retrieved. |

**Returns**

List of exchange rates meeting criteria filters.

### ExchangeRateApi.GetList Method

Retrieves a list of exchange rates.

**C#**

```
public List<ExchangeRateData> GetList(Criteria<ExchangeRateProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<ExchangeRateProperty>) As List<ExchangeRateData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | Criteria by which to filter exchange rates being retrieved. |

**Returns**

List of exchange rates meeting criteria filters.

### ExchangeRateApi.Update Method

Updates an existing Exchange Rate in Cms (see page 7).

**C#**

```
public void Update(ExchangeRateData exchangeRateData);
```

**Visual Basic**

```
Public Function Update(exchangeRateData As ExchangeRateData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| exchangeRateData | ExchangeRateData (see page 1281) object to save. |

# InventoryApi Class

Api class for managing Commerce (see page 1201) Inventory in Cms (see page 7).

### Class Hierarchy

### C#

```
public class InventoryApi : CommonApiBase;
```

### Visual Basic

```
Public Class InventoryApi
Inherits CommonApiBase
```

### File

InventoryApi.cs

## InventoryApi Methods

The methods of the InventoryApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | SaveInventory (see page 608) | Saves inventory information for a catalog entry. |

### InventoryApi.SaveInventory Method

Saves inventory information for a catalog entry.

### C#

```
public void SaveInventory(InventoryData inventory);
```

### Visual Basic

```
Public Function SaveInventory(inventory As InventoryData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| inventoryData | inventory to save. |

# OrderApi Class

Api class for managing Commerce (see page 1201) Orders in Cms (see page 7).

### Class Hierarchy

### C#

```
public class OrderApi : CommonApiBase;
```

### Visual Basic

```
Public Class OrderApi
Inherits CommonApiBase
```

**File**

OrderApi.cs

## OrderApi Methods

The methods of the OrderApi class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Capture (see page 609) | This is the overview for the Capture method overload. |
| | GetCustomerOrderList (see page 610) | Retrieves a list of orders for a given customer. |
| | GetItem (see page 610) | Retrieves a single order. |
| | GetList (see page 611) | This is the overview for the GetList method overload. |
| | GetOnHoldOrderList (see page 611) | This is the overview for the GetOnHoldOrderList method overload. |
| | GetOrderMessage (see page 612) | This is the overview for the GetOrderMessage method overload. |
| | GetOrderPayment (see page 613) | Returns an order payment |
| | GetOrderStatus (see page 614) | Returns the status of an order. |
| | GetProductOrderList (see page 614) | This is the overview for the GetProductOrderList method overload. |
| | GetReport (see page 615) | Returns all the OrderReportData for given criteria. |
| | ItemsInStock (see page 615) | Returns true if all order items are in stock. |
| | MarkAsFraud (see page 615) | Marks an order as fraudulent. |
| | ProcessOrderInventory (see page 615) | Update all inventory levels for the supplied order. |
| | RollbackOrderInventory (see page 616) | Rolls back all inventory level updates for the supplied order. |
| | SetOrderStatus (see page 616) | Sets the status of an order. |
| | SetOrderToHold (see page 616) | Sets an order to OnHold. |
| | SetTrackingNumber (see page 616) | Sets the tracking number for an order part. |

### Capture Method

This is the overview for the Capture method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | OrderApi.Capture (long) (see page 609) | Captures the payment for an order. |
| | OrderApi.Capture (long, long) (see page 610) | Captures the payment for an order. |

### OrderApi.Capture Method (long)

Captures the payment for an order.

**C#**

```
public void Capture(long orderId);
```

**Visual Basic**

```
Public Function Capture(orderId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order |

### OrderApi.Capture Method (long, long)

Captures the payment for an order.

**C#**

```
public void Capture(long orderId, long paymentId);
```

**Visual Basic**

```
Public Function Capture(orderId As long, paymentId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order |

### OrderApi.GetCustomerOrderList Method

Retrieves a list of orders for a given customer.

**C#**

```
public List<OrderData> GetCustomerOrderList(long customerId, PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetCustomerOrderList(customerId As long, pagingInfo As PagingInfo) As
List<OrderData>
```

**Parameters**

| Parameters | Description |
|---|---|
| customerId | Id of customer to retrieve orders for. |
| pagingInfo | paging information for retrieval. |

**Returns**

List of OrderData (see page 1298) objects

### OrderApi.GetItem Method

Retrieves a single order.

**C#**

```
public OrderData GetItem(long orderId);
```

**Visual Basic**

```
Public Function GetItem(orderId As long) As OrderData
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order to retrieve. |

## GetList Method

This is the overview for the GetList method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | OrderApi.GetList (Criteria<OrderProperty>) (see page 611) | Retrieves a list of orders. |
| | OrderApi.GetList (DateTime, DateTime, PagingInfo) (see page 611) | Returns a list of orders for a given time period. |

## OrderApi.GetList Method (Criteria<OrderProperty>)

Retrieves a list of orders.

### C#

```
public List<OrderData> GetList(Criteria<OrderProperty> criteria);
```

### Visual Basic

```
Public Function GetList(criteria As Criteria<OrderProperty>) As List<OrderData>
```

### Parameters

| Parameters | Description |
|---|---|
| criteria | criteria by which to retrieve orders. |

### Returns

List of OrderData (see page 1298) objects

## OrderApi.GetList Method (DateTime, DateTime, PagingInfo)

Returns a list of orders for a given time period.

### C#

```
public List<OrderData> GetList(DateTime startDate, DateTime endDate, PagingInfo pagingInfo);
```

### Visual Basic

```
Public Function GetList(startDate As DateTime, endDate As DateTime, pagingInfo As PagingInfo)
As List<OrderData>
```

### Parameters

| Parameters | Description |
|---|---|
| startDate | begininning of time period. |
| endDate | end of time period. |
| pagingInfo | paging information for list or orders. |

### Returns

List of OrderData (see page 1298) objects

## GetOnHoldOrderList Method

This is the overview for the GetOnHoldOrderList method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | OrderApi.GetOnHoldOrderList (PagingInfo) (see page 612) | Retrieves a list of orders on hold. |
| | OrderApi.GetOnHoldOrderList (PagingInfo, OrderProperty, EkEnumeration.OrderByDirection) (see page 612) | Retrieves a list of orders on hold. |

## OrderApi.GetOnHoldOrderList Method (PagingInfo)

Retrieves a list of orders on hold.

**C#**

```
public List<OrderData> GetOnHoldOrderList(PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetOnHoldOrderList(pagingInfo As PagingInfo) As List<OrderData>
```

**Returns**

List of OrderData (see page 1298) objects

List of OrderData (see page 1298) objects

## OrderApi.GetOnHoldOrderList Method (PagingInfo, OrderProperty, EkEnumeration.OrderByDirection)

Retrieves a list of orders on hold.

**C#**

```
public List<OrderData> GetOnHoldOrderList(PagingInfo pagingInfo, OrderProperty orderBy,
EkEnumeration.OrderByDirection orderDirection);
```

**Visual Basic**

```
Public Function GetOnHoldOrderList(pagingInfo As PagingInfo, orderBy As OrderProperty,
orderDirection As EkEnumeration.OrderByDirection) As List<OrderData>
```

**Parameters**

| Parameters | Description |
|---|---|
| orderBy | parameter to order list by. |
| orderDirection | direction to order list in. |

**Returns**

List of OrderData (see page 1298) objects

List of OrderData (see page 1298) objects

## GetOrderMessage Method

This is the overview for the GetOrderMessage method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | OrderApi.GetOrderMessage (EkEnumeration.EmailMessageTypes, long) (see page 613) | Returns the default EmailMessage for the supplied type.. Any tokens will be replaced based upon the supplied OrderId. |
| | OrderApi.GetOrderMessage (long, long) (see page 613) | Returns the EmailMessage for the supplied messageId. Any tokens will be replaced based upon the supplied OrderId. |

## OrderApi.GetOrderMessage Method (EkEnumeration.EmailMessageTypes, long)

Returns the default EmailMessage for the supplied type.. Any tokens will be replaced based upon the supplied OrderId.

**C#**

```
public EmailMessageData GetOrderMessage(EkEnumeration.EmailMessageTypes messageType, long orderId);
```

**Visual Basic**

```
Public Function GetOrderMessage(messageType As EkEnumeration.EmailMessageTypes, orderId As long) As EmailMessageData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageType | Type of message to retrieve. i.e. OrderReceived, OrderShipped, OrderCancelled |
| orderId | Id of order to retrieve message for. |

## OrderApi.GetOrderMessage Method (long, long)

Returns the EmailMessage for the supplied messageId. Any tokens will be replaced based upon the supplied OrderId.

**C#**

```
public EmailMessageData GetOrderMessage(long messageId, long orderId);
```

**Visual Basic**

```
Public Function GetOrderMessage(messageId As long, orderId As long) As EmailMessageData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | Id of message to retrieve. |
| orderId | Id of order to retrieve message for.. |

## OrderApi.GetOrderPayment Method

Returns an order payment

**C#**

```
public OrderPaymentData GetOrderPayment(long paymentId);
```

**Visual Basic**

```
Public Function GetOrderPayment(paymentId As long) As OrderPaymentData
```

**Parameters**

| Parameters | Description |
|---|---|
| paymentId | The id of the payment to retrieve. |

### OrderApi.GetOrderStatus Method

Returns the status of an order.

**C#**

```
public EkEnumeration.OrderStatus GetOrderStatus(long orderId);
```

**Visual Basic**

```
Public Function GetOrderStatus(orderId As long) As EkEnumeration.OrderStatus
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order to retrieve status for. |

### GetProductOrderList Method

This is the overview for the GetProductOrderList method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | OrderApi.GetProductOrderList (long, Criteria<OrderProperty>) (see page 614) | Retrieves a list of orders for a given product. |
| | OrderApi.GetProductOrderList (long, PagingInfo) (see page 614) | Retrieves a list of orders for a given product. |

### OrderApi.GetProductOrderList Method (long, Criteria<OrderProperty>)

Retrieves a list of orders for a given product.

**C#**

```
public List<OrderData> GetProductOrderList(long productId, Criteria<OrderProperty> criteria);
```

**Visual Basic**

```
Public Function GetProductOrderList(productId As long, criteria As Criteria<OrderProperty>) As
List<OrderData>
```

**Parameters**

| Parameters | Description |
|---|---|
| productId | Id of proiduct to retrieve orders for. |
| criteria | criteria information for filtering and sorting retrieval. |

**Returns**

List of OrderData (see page 1298) objects

### OrderApi.GetProductOrderList Method (long, PagingInfo)

Retrieves a list of orders for a given product.

**C#**

```
public List<OrderData> GetProductOrderList(long productId, PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetProductOrderList(productId As long, pagingInfo As PagingInfo) As
List<OrderData>
```

**Parameters**

| Parameters | Description |
|---|---|
| productId | Id of proiduct to retrieve orders for. |
| pagingInfo | paging information for retrieval. |

**Returns**

List of OrderData (see page 1298) objects

### OrderApi.GetReport Method

Returns all the OrderReportData for given criteria.

**C#**

```
public Reporting.OrderReportData GetReport(Criteria<OrderProperty> criteria);
```

**Visual Basic**

```
Public Function GetReport(criteria As Criteria<OrderProperty>) As Reporting.OrderReportData
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | criteria for reporting. |

### OrderApi.ItemsInStock Method

Returns true if all order items are in stock.

**C#**

```
public bool ItemsInStock(long orderId);
```

**Visual Basic**

```
Public Function ItemsInStock(orderId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order to verify stock levels for. |

### OrderApi.MarkAsFraud Method

Marks an order as fraudulent.

**C#**

```
public void MarkAsFraud(long orderId);
```

**Visual Basic**

```
Public Function MarkAsFraud(orderId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order |

### OrderApi.ProcessOrderInventory Method

Update all inventory levels for the supplied order.

**C#**

```
public void ProcessOrderInventory(long orderId);
```

**Visual Basic**

```
Public Function ProcessOrderInventory(orderId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order to process. |

### OrderApi.RollbackOrderInventory Method

Rolls back all inventory level updates for the supplied order.

**C#**

```
public void RollbackOrderInventory(long orderId);
```

**Visual Basic**

```
Public Function RollbackOrderInventory(orderId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order to rollback. |

### OrderApi.SetOrderStatus Method

Sets the status of an order.

**C#**

```
public void SetOrderStatus(long orderId, EkEnumeration.OrderStatus status);
```

**Visual Basic**

```
Public Function SetOrderStatus(orderId As long, status As EkEnumeration.OrderStatus) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order |
| status | Status to change the order to. |

### OrderApi.SetOrderToHold Method

Sets an order to OnHold.

**C#**

```
public void SetOrderToHold(long orderId);
```

**Visual Basic**

```
Public Function SetOrderToHold(orderId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order |

### OrderApi.SetTrackingNumber Method

Sets the tracking number for an order part.

**C#**

```
public void SetTrackingNumber(long orderId, string trackingNumber, bool markAsShipped);
```

**Visual Basic**

```
Public Function SetTrackingNumber(orderId As long, trackingNumber As string, markAsShipped As
bool) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| orderId | Id of order to set tracking number. |
| trackingNumber | The tracking number |
| markAsShipped | mark the item as shipped. |

# PackageApi Class

API (see page 639) class used for managing shipping packages.

**Class Hierarchy**

**C#**

```
public class PackageApi : CommonApiBase;
```

**Visual Basic**

```
Public Class PackageApi
Inherits CommonApiBase
```

**File**

PackageApi.cs

## PackageApi Methods

The methods of the PackageApi class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Add (see page 617) | Adds a new Package to Cms (see page 7). The Package.Id property will be populated with the new Package's Id. |
| | Delete (see page 618) | Deletes a Package from the Cms (see page 7). |
| | GetItem (see page 618) | Retrieves a package by package Id. |
| | GetList (see page 618) | Retrieves a list of Packages. |
| | Update (see page 618) | Updates an existing Package in Cms (see page 7). |

### PackageApi.Add Method

Adds a new Package to Cms (see page 7). The Package.Id property will be populated with the new Package's Id.

**C#**

```
public void Add(PackageData packageData);
```

**Visual Basic**

```
Public Function Add(packageData As PackageData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| packageData | PackageData (see page 1309) object to add. |

## PackageApi.Delete Method

Deletes a Package from the Cms (see page 7).

**C#**

```
public void Delete(long id);
```

**Visual Basic**

```
Public Function Delete(id As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of Package to delete. |

## PackageApi.GetItem Method

Retrieves a package by package Id.

**C#**

```
public PackageData GetItem(long id);
```

**Visual Basic**

```
Public Function GetItem(id As long) As PackageData
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of package to retrieve. |

## PackageApi.GetList Method

Retrieves a list of Packages.

**C#**

```
public List<PackageData> GetList(Criteria<PackageProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<PackageProperty>) As List<PackageData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | Criteria by which to filter package being retrieved. |

**Returns**

List of Packages meeting criteria filters.

## PackageApi.Update Method

Updates an existing Package in Cms (see page 7).

**C#**

```
public void Update(PackageData packageData);
```

### Visual Basic

```
Public Function Update(packageData As PackageData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| packageData | PackageData (see page 1309) object to save. |

# ProductTypeApi Class

Api class for managing Commerce (see page 1201) product types in Cms (see page 7).

### Class Hierarchy

### C#

```
public class ProductTypeApi : CommonApiBase;
```

### Visual Basic

```
Public Class ProductTypeApi
Inherits CommonApiBase
```

### File

ProductTypeApi.cs

## ProductTypeApi Methods

The methods of the ProductTypeApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 619) | This is Add, a member of class ProductTypeApi. |
| | AddProductAttribute (see page 620) | This is AddProductAttribute, a member of class ProductTypeApi. |
| | Delete (see page 620) | This is Delete, a member of class ProductTypeApi. |
| | DeleteProductAttribute (see page 620) | This is the overview for the DeleteProductAttribute method overload. |
| | GetFolderProductTypeList (see page 621) | This is GetFolderProductTypeList, a member of class ProductTypeApi. |
| | GetItem (see page 621) | This is the overview for the GetItem method overload. |
| | GetProductAttribute (see page 622) | This is GetProductAttribute, a member of class ProductTypeApi. |
| | GetProductAttributeList (see page 622) | This is GetProductAttributeList, a member of class ProductTypeApi. |
| | Update (see page 622) | This is Update, a member of class ProductTypeApi. |
| | UpdateProductAttribute (see page 622) | This is UpdateProductAttribute, a member of class ProductTypeApi. |

### ProductTypeApi.Add Method

### C#

```
public void Add(ProductTypeData productType);
```

### Visual Basic

```
Public Function Add(productType As ProductTypeData) As void
```

**Description**

This is Add, a member of class ProductTypeApi.

### ProductTypeApi.AddProductAttribute Method

**C#**

```
public long AddProductAttribute(long productTypeId, ProductTypeAttributeData attribute,
System.Data.IDbTransaction transaction);
```

**Visual Basic**

```
Public Function AddProductAttribute(productTypeId As long, attribute As
ProductTypeAttributeData, transaction As System.Data.IDbTransaction) As long
```

**Description**

This is AddProductAttribute, a member of class ProductTypeApi.

### ProductTypeApi.Delete Method

**C#**

```
public void Delete(long id);
```

**Visual Basic**

```
Public Function Delete(id As long) As void
```

**Description**

This is Delete, a member of class ProductTypeApi.

### DeleteProductAttribute Method

This is the overview for the DeleteProductAttribute method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | ProductTypeApi.DeleteProductAttribute (long, bool, System.Data.IDbTransaction) (see page 621) | This is DeleteProductAttribute, a member of class ProductTypeApi. |
| | ProductTypeApi.DeleteProductAttribute (long, System.Data.IDbTransaction) (see page 620) | This is DeleteProductAttribute, a member of class ProductTypeApi. |

### ProductTypeApi.DeleteProductAttribute Method (long, System.Data.IDbTransaction)

**C#**

```
public void DeleteProductAttribute(long attributeId, System.Data.IDbTransaction transaction);
```

**Visual Basic**

```
Public Function DeleteProductAttribute(attributeId As long, transaction As
System.Data.IDbTransaction) As void
```

**Description**

This is DeleteProductAttribute, a member of class ProductTypeApi.

### ProductTypeApi.DeleteProductAttribute Method (long, bool, System.Data.IDbTransaction)

**C#**

```
public void DeleteProductAttribute(long attributeId, bool forceDelete,
System.Data.IDbTransaction transaction);
```

**Visual Basic**

```
Public Function DeleteProductAttribute(attributeId As long, forceDelete As bool, transaction
As System.Data.IDbTransaction) As void
```

**Description**

This is DeleteProductAttribute, a member of class ProductTypeApi.

### ProductTypeApi.GetFolderProductTypeList Method

**C#**

```
public List<ProductTypeData> GetFolderProductTypeList(long folderId);
```

**Visual Basic**

```
Public Function GetFolderProductTypeList(folderId As long) As List<ProductTypeData>
```

**Description**

This is GetFolderProductTypeList, a member of class ProductTypeApi.

### GetItem Method

This is the overview for the GetItem method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | ProductTypeApi.GetItem (long) (see page 621) | This is GetItem, a member of class ProductTypeApi. |
| | ProductTypeApi.GetItem (long, bool) (see page 621) | This is GetItem, a member of class ProductTypeApi. |

### ProductTypeApi.GetItem Method (long)

**C#**

```
public ProductTypeData GetItem(long productTypeId);
```

**Visual Basic**

```
Public Function GetItem(productTypeId As long) As ProductTypeData
```

**Description**

This is GetItem, a member of class ProductTypeApi.

### ProductTypeApi.GetItem Method (long, bool)

**C#**

```
public ProductTypeData GetItem(long productTypeId, bool withAttributes);
```

**Visual Basic**

```
Public Function GetItem(productTypeId As long, withAttributes As bool) As ProductTypeData
```

**Description**

This is GetItem, a member of class ProductTypeApi.

### ProductTypeApi.GetProductAttribute Method

**C#**

```
public ProductTypeAttributeData GetProductAttribute(long attributeId);
```

**Visual Basic**

```
Public Function GetProductAttribute(attributeId As long) As ProductTypeAttributeData
```

**Description**

This is GetProductAttribute, a member of class ProductTypeApi.

### ProductTypeApi.GetProductAttributeList Method

**C#**

```
public List<ProductTypeAttributeData> GetProductAttributeList(long productTypeId);
```

**Visual Basic**

```
Public Function GetProductAttributeList(productTypeId As long) As
List<ProductTypeAttributeData>
```

**Description**

This is GetProductAttributeList, a member of class ProductTypeApi.

### ProductTypeApi.Update Method

**C#**

```
public void Update(ProductTypeData productType);
```

**Visual Basic**

```
Public Function Update(productType As ProductTypeData) As void
```

**Description**

This is Update, a member of class ProductTypeApi.

### ProductTypeApi.UpdateProductAttribute Method

**C#**

```
public void UpdateProductAttribute(ProductTypeAttributeData attribute,
System.Data.IDbTransaction transaction);
```

**Visual Basic**

```
Public Function UpdateProductAttribute(attribute As ProductTypeAttributeData, transaction As
System.Data.IDbTransaction) As void
```

**Description**

This is UpdateProductAttribute, a member of class ProductTypeApi.

## RecommendationApi Class

Api class for managing entry reccomendations.

### Class Hierarchy

### C#

```
public class RecommendationApi : CommonApiBase;
```

### Visual Basic

```
Public Class RecommendationApi
Inherits CommonApiBase
```

### File

RecommendationApi.cs

## RecommendationApi Methods

The methods of the RecommendationApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 623) | Adds a new entry recommendation to cms. The supplied recommendation will be updated with the new Id. |
| | Delete (see page 623) | Deletes a recommendation. |
| | DeleteRecommendationByEntry (see page 624) | Delete (see page 623) recommendations associated with a catalog entry. |
| | GetItem (see page 624) | Retrieves a recommendation by id. |
| | Update (see page 624) | Updates an existing entry recommendation in cms. |
| | UpdateCrossSell (see page 624) | This is UpdateCrossSell, a member of class RecommendationApi. |
| | UpdateUpSell (see page 625) | This is UpdateUpSell, a member of class RecommendationApi. |

### RecommendationApi.Add Method

Adds a new entry recommendation to cms. The supplied recommendation will be updated with the new Id.

### C#

```
public void Add(RecommendationItemData recommendation);
```

### Visual Basic

```
Public Function Add(recommendation As RecommendationItemData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| recommendation | recommendation data object to add. |

### RecommendationApi.Delete Method

Deletes a recommendation.

### C#

```
public void Delete(long recommendationId);
```

### Visual Basic

```
Public Function Delete(recommendationId As long) As void
```

### Parameters

| Parameters | Description |
|---|---|
| recommendationId | id of recommendation to delete. |

### RecommendationApi.DeleteRecommendationByEntry Method

Delete (see page 623) recommendations associated with a catalog entry.

**C#**

```
public void DeleteRecommendationByEntry(long entryId);
```

**Visual Basic**

```
Public Function DeleteRecommendationByEntry(entryId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| entryId | Id of entry to delete recommendations for. |

### RecommendationApi.GetItem Method

Retrieves a recommendation by id.

**C#**

```
public RecommendationItemData GetItem(long recommendationId);
```

**Visual Basic**

```
Public Function GetItem(recommendationId As long) As RecommendationItemData
```

**Parameters**

| Parameters | Description |
|---|---|
| recommendationId | Id of recommendation to retrieve. |

### RecommendationApi.Update Method

Updates an existing entry recommendation in cms.

**C#**

```
public void Update(RecommendationItemData recommendation);
```

**Visual Basic**

```
Public Function Update(recommendation As RecommendationItemData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| recommendation | recommendation data object to update. |

### RecommendationApi.UpdateCrossSell Method

**C#**

```
public void UpdateCrossSell(long entryId, IEnumerable<RecommendationItemData> recommendations);
```

**Visual Basic**

```
Public Function UpdateCrossSell(entryId As long, recommendations As
IEnumerable<RecommendationItemData>) As void
```

**Description**

This is UpdateCrossSell, a member of class RecommendationApi.

### RecommendationApi.UpdateUpSell Method

**C#**

```
public void UpdateUpSell(long entryId, IEnumerable<RecommendationItemData> recommendations);
```

**Visual Basic**

```
Public Function UpdateUpSell(entryId As long, recommendations As
IEnumerable<RecommendationItemData>) As void
```

**Description**

This is UpdateUpSell, a member of class RecommendationApi.

## RegionApi Class

Api class for managing regions in Cms (see page 7).

**Class Hierarchy**

**C#**

```
public class RegionApi : CommonApiBase;
```

**Visual Basic**

```
Public Class RegionApi
Inherits CommonApiBase
```

**File**

RegionApi.cs

## RegionApi Methods

The methods of the RegionApi class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Add (see page 625) | Adds a new Region to the Cms (see page 7). The Region.Id property will be populated with the new regions Id. |
| | CanDelete (see page 626) | Checks to see if a Region can be deleted. |
| | Delete (see page 626) | Deletes a region from the Cms (see page 7). |
| | GetItem (see page 626) | Retrieves a region by region Id. |
| | GetList (see page 626) | Retrieves a list of regions. |
| | Update (see page 627) | Updates an existing Region in the Cms (see page 7). |

### RegionApi.Add Method

Adds a new Region to the Cms (see page 7). The Region.Id property will be populated with the new regions Id.

**C#**

```
public void Add(RegionData regionData);
```

**Visual Basic**

```
Public Function Add(regionData As RegionData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| regionData | RegionData (see page 1330) object to add. |

### RegionApi.CanDelete Method

Checks to see if a Region can be deleted.

**C#**

```
public bool CanDelete(long regionId, out ValidationResults result);
```

**Visual Basic**

```
Public Function CanDelete(regionId As long, result As out ValidationResults) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| result | ValidationResult returned containing messages if Region cannot be deleted. |
| currencyId | Id of Region to delete. |

**Returns**

true if region can safely be deleted.

### RegionApi.Delete Method

Deletes a region from the Cms (see page 7).

**C#**

```
public void Delete(long id);
```

**Visual Basic**

```
Public Function Delete(id As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of region to delete. |

### RegionApi.GetItem Method

Retrieves a region by region Id.

**C#**

```
public RegionData GetItem(long id);
```

**Visual Basic**

```
Public Function GetItem(id As long) As RegionData
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of region to retrieve. |

### RegionApi.GetList Method

Retrieves a list of regions.

**C#**

```
public List<RegionData> GetList(Criteria<RegionProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<RegionProperty>) As List<RegionData>
```

### Parameters

| Parameters | Description |
|---|---|
| criteria | Criteria by which to filter regions being retrieved. |

### Returns

List of regions meeting criteria filters.

### RegionApi.Update Method

Updates an existing Region in the Cms (see page 7).

**C#**

```
public void Update(RegionData regionData);
```

**Visual Basic**

```
Public Function Update(regionData As RegionData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| regionData | RegionData (see page 1330) object to save. |

## ShippingMethodApi Class

Api class for managing shipping methods in CMS.

### Class Hierarchy

**C#**

```
public class ShippingMethodApi : CommonApiBase;
```

**Visual Basic**

```
Public Class ShippingMethodApi
Inherits CommonApiBase
```

### File

ShippingApi.cs

## ShippingMethodApi Methods

The methods of the ShippingMethodApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 627) | This is Add, a member of class ShippingMethodApi. |
| | Update (see page 628) | This is Update, a member of class ShippingMethodApi. |

### ShippingMethodApi.Add Method

**C#**

```
public void Add(ShippingMethodData shipOption);
```

**Visual Basic**

```
Public Function Add(shipOption As ShippingMethodData) As void
```

### Description

This is Add, a member of class ShippingMethodApi.

### ShippingMethodApi.Update Method

**C#**

```
public void Update(ShippingMethodData shipOption);
```

**Visual Basic**

```
Public Function Update(shipOption As ShippingMethodData) As void
```

### Description

This is Update, a member of class ShippingMethodApi.

# TaxApi Class

Api class for managing Commerce (see page 1201) Tax rates in Cms (see page 7).

### Class Hierarchy

**C#**

```
public class TaxApi : CommonApiBase;
```

**Visual Basic**

```
Public Class TaxApi
Inherits CommonApiBase
```

### File

TaxApi.cs

## TaxApi Methods

The methods of the TaxApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 629) | Adds a new Tax rate to Cms (see page 7). The Tax.Id property will be populated with the new Tax's Id. |
| | Delete (see page 629) | Deletes a Tax rate from the Cms (see page 7). |
| | GetApplicableTaxRate (see page 629) | Returns the applicable tax rate for the supplied addresses and tax class. |
| | GetApplicableTaxRateList (see page 629) | Returns a list of applicable tax rates for the supplied addresses. There will be at most one rate for each available tax class for the the specified addresses. |
| | GetItem (see page 630) | This is the overview for the GetItem method overload. |
| | GetItemByCountryId (see page 630) | Retrieves a tax rate based upon tax class and country Id. For Example, retrieves a TaxRateType.CountrySalesTax with the country Id of 840.. |
| | GetItemByPostalCode (see page 631) | This is the overview for the GetItemByPostalCode method overload. |
| | GetItemByPostalId (see page 632) | Retrieves a tax rate based upon tax class and postal code Id. For Example, retrieves a TaxRateType.PostalSalesTax with the Postal Code Id of 40.. |
| | GetItemByRegionId (see page 632) | Retrieves a tax rate based upon tax class and Type Item Id. For Example, retrieves a TaxRateType.RegionSalesTax with the RegionId of 40.. |
| | GetList (see page 632) | Retrieves a list of tax rates. |
| | Update (see page 632) | Updates an existing tax rate in Cms (see page 7). |

### TaxApi.Add Method

Adds a new Tax rate to Cms (see page 7). The Tax.Id property will be populated with the new Tax's Id.

**C#**

```
public void Add(TaxRateData taxRateData);
```

**Visual Basic**

```
Public Function Add(taxRateData As TaxRateData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| taxRateData | TaxRateData (see page 1338) object to add. |

### TaxApi.Delete Method

Deletes a Tax rate from the Cms (see page 7).

**C#**

```
public void Delete(long id);
```

**Visual Basic**

```
Public Function Delete(id As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of Tax to delete. |

### TaxApi.GetApplicableTaxRate Method

Returns the applicable tax rate for the supplied addresses and tax class.

**C#**

```
public virtual TaxRateData GetApplicableTaxRate(AddressData shipFromAddress, AddressData
shipToAddress, long taxClassId);
```

**Visual Basic**

```
Public virtual Function GetApplicableTaxRate(shipFromAddress As AddressData, shipToAddress As
AddressData, taxClassId As long) As TaxRateData
```

**Parameters**

| Parameters | Description |
|---|---|
| shipFromAddress | Address where the products are coming from. |
| shipToAddress | Address where products are going. |
| taxClassId | The taxClass for which to retrieve a rate |

### TaxApi.GetApplicableTaxRateList Method

Returns a list of applicable tax rates for the supplied addresses. There will be at most one rate for each available tax class for the the specified addresses.

**C#**

```
public virtual List<TaxRateData> GetApplicableTaxRateList(AddressData shipFromAddress,
AddressData shipToAddress);
```

**Visual Basic**

```
Public virtual Function GetApplicableTaxRateList(shipFromAddress As AddressData, shipToAddress
As AddressData) As List<TaxRateData>
```

**Parameters**

| Parameters | Description |
|---|---|
| shipFromAddress | Address where the products are coming from. |
| shipToAddress | Address where products are going. |

## GetItem Method

This is the overview for the GetItem method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | TaxApi.GetItem (long) (see page 630) | Retrieves a tax rate by Id. |
| | TaxApi.GetItem (long, long) (see page 630) | Retrieves a tax rate based up Tax Rate Type and Type Item Id. For Example, retrieves a TaxRateType.RegionSalesTax with the RegionId of 40.. |

## TaxApi.GetItem Method (long)

Retrieves a tax rate by Id.

**C#**

```
public TaxRateData GetItem(long id);
```

**Visual Basic**

```
Public Function GetItem(id As long) As TaxRateData
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of tax rate to retrieve. |

## TaxApi.GetItem Method (long, long)

Retrieves a tax rate based up Tax Rate Type and Type Item Id. For Example, retrieves a TaxRateType.RegionSalesTax with the RegionId of 40..

**C#**

```
public TaxRateData GetItem(long taxTypeId, long typeItemId);
```

**Visual Basic**

```
Public Function GetItem(taxTypeId As long, typeItemId As long) As TaxRateData
```

**Parameters**

| Parameters | Description |
|---|---|
| taxTypeId | Type of reate to retrieve. |
| typeItemId | Id of corresponding object. For example, RegionId for TaxRateType.RegionSalesTax |

## TaxApi.GetItemByCountryId Method

Retrieves a tax rate based upon tax class and country Id. For Example, retrieves a TaxRateType.CountrySalesTax with the country Id of 840..

**C#**

```
public virtual TaxRateData GetItemByCountryId(long taxClassId, long countryId);
```

**Visual Basic**

```
Public virtual Function GetItemByCountryId(taxClassId As long, countryId As long) As
TaxRateData
```

**Parameters**

| Parameters | Description |
|---|---|
| taxClassId | Id of tax class to retrieve rate for. |
| countryId | Id of country to retrieve tax for. |

## GetItemByPostalCode Method

This is the overview for the GetItemByPostalCode method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | TaxApi.GetItemByPostalCode (long, string) (see page 631) | Retrieves a tax rate based upon tax class and postal code . |
| | TaxApi.GetItemByPostalCode (long, string, int) (see page 631) | Retrieves a tax rate based upon tax class and postal code. |

## TaxApi.GetItemByPostalCode Method (long, string)

Retrieves a tax rate based upon tax class and postal code .

**C#**

```
public TaxRateData GetItemByPostalCode(long taxClassId, string postalCode);
```

**Visual Basic**

```
Public Function GetItemByPostalCode(taxClassId As long, postalCode As string) As TaxRateData
```

**Parameters**

| Parameters | Description |
|---|---|
| taxClassId | Id of tax class to retrieve rate for. |
| postalCode | postalCode to retrieve taxrate for. |

## TaxApi.GetItemByPostalCode Method (long, string, int)

Retrieves a tax rate based upon tax class and postal code.

**C#**

```
public TaxRateData GetItemByPostalCode(long taxClassId, string postalCode, int countryId);
```

**Visual Basic**

```
Public Function GetItemByPostalCode(taxClassId As long, postalCode As string, countryId As
Integer) As TaxRateData
```

**Parameters**

| Parameters | Description |
|---|---|
| taxClassId | Id of tax class to retrieve rate for. |
| postalCode | postalCode to retrieve taxrate for. |
| countryId | Id of country associated with PostalCode |

### TaxApi.GetItemByPostalId Method

Retrieves a tax rate based upon tax class and postal code Id. For Example, retrieves a TaxRateType.PostalSalesTax with the Postal Code Id of 40..

**C#**

```
public TaxRateData GetItemByPostalId(long taxClassId, long postalCodeId);
```

**Visual Basic**

```
Public Function GetItemByPostalId(taxClassId As long, postalCodeId As long) As TaxRateData
```

**Parameters**

| Parameters | Description |
|---|---|
| taxClassId | Id of tax class to retrieve rate for. |
| postalCodeId | Id of postal code to retrieve tax for. |

### TaxApi.GetItemByRegionId Method

Retrieves a tax rate based upon tax class and Type Item Id. For Example, retrieves a TaxRateType.RegionSalesTax with the RegionId of 40..

**C#**

```
public TaxRateData GetItemByRegionId(long taxClassId, long regionId);
```

**Visual Basic**

```
Public Function GetItemByRegionId(taxClassId As long, regionId As long) As TaxRateData
```

**Parameters**

| Parameters | Description |
|---|---|
| taxClassId | Type of reate to retrieve. |
| regionId | Id of Region to retrieve rate for. |

### TaxApi.GetList Method

Retrieves a list of tax rates.

**C#**

```
public List<TaxRateData> GetList(Criteria<TaxRateProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<TaxRateProperty>) As List<TaxRateData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | Criteria by which to filter tax being retrieved. |

**Returns**

List of countries meeting criteria filters.

### TaxApi.Update Method

Updates an existing tax rate in Cms (see page 7).

**C#**

```
public void Update(TaxRateData taxRateData);
```

**Visual Basic**

```
Public Function Update(taxRateData As TaxRateData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| taxRateData | TaxRateData (see page 1338) object to save. |

# TaxClassApi Class

Api class for managing Commerce (see page 1201) Tax Classes in Cms (see page 7).

**Class Hierarchy**

**C#**

```
public class TaxClassApi : CommonApiBase;
```

**Visual Basic**

```
Public Class TaxClassApi
Inherits CommonApiBase
```

**File**

TaxClassApi.cs

## TaxClassApi Methods

The methods of the TaxClassApi class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Add (see page 633) | Adds a new Tax Class to Cms (see page 7). The TaxClass.Id property will be populated with the new TaxClass's Id. |
| | Delete (see page 633) | Deletes a Tax Class from the Cms (see page 7). |
| | GetItem (see page 634) | Retrieves a tax class by Id. |
| | GetList (see page 634) | Retrieves a list of Tax Classes. |
| | Update (see page 634) | Updates an existing Tax Class in Cms (see page 7). |

### TaxClassApi.Add Method

Adds a new Tax Class to Cms (see page 7). The TaxClass.Id property will be populated with the new TaxClass's Id.

**C#**

```
public void Add(TaxClassData taxClassData);
```

**Visual Basic**

```
Public Function Add(taxClassData As TaxClassData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| taxClassData | TaxClassData (see page 1336) object to add. |

### TaxClassApi.Delete Method

Deletes a Tax Class from the Cms (see page 7).

**C#**

```
public void Delete(long id);
```

**Visual Basic**

```
Public Function Delete(id As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of TaxClass to delete. |

### TaxClassApi.GetItem Method

Retrieves a tax class by Id.

**C#**

```
public TaxClassData GetItem(long id);
```

**Visual Basic**

```
Public Function GetItem(id As long) As TaxClassData
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of taxClass to retrieve. |

### TaxClassApi.GetList Method

Retrieves a list of Tax Classes.

**C#**

```
public List<TaxClassData> GetList(Criteria<TaxClassProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<TaxClassProperty>) As List<TaxClassData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | Criteria by which to filter taxClass being retrieved. |

**Returns**

List of TaxClass data objects.

### TaxClassApi.Update Method

Updates an existing Tax Class in Cms (see page 7).

**C#**

```
public void Update(TaxClassData taxClassData);
```

**Visual Basic**

```
Public Function Update(taxClassData As TaxClassData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| taxClassData | TaxClassData (see page 1336) object to save. |

# TaxTypeApi Class

Api class for managing Commerce (see page 1201) tax types in Cms (see page 7).

### Class Hierarchy

### C#

```
public class TaxTypeApi : CommonApiBase;
```

### Visual Basic

```
Public Class TaxTypeApi
Inherits CommonApiBase
```

### File

TaxTypeApi.cs

## TaxTypeApi Methods

The methods of the TaxTypeApi class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 635) | Adds a new Tax Type Cms (see page 7). The TaxType.Id property will be populated with the new TaxType's Id. |
| | Delete (see page 635) | Deletes a Tax Type from the Cms (see page 7). |
| | GetItem (see page 636) | Retrieves a tax type by Id. |
| | GetList (see page 636) | Retrieves a list of Tax Types. |
| | Update (see page 636) | Updates an existing Tax Type in Cms (see page 7). |

### TaxTypeApi.Add Method

Adds a new Tax Type Cms (see page 7). The TaxType.Id property will be populated with the new TaxType's Id.

### C#

```
public void Add(TaxTypeData taxTypeData);
```

### Visual Basic

```
Public Function Add(taxTypeData As TaxTypeData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| taxTypeData | TaxTypeData (see page 1341) object to add. |

### TaxTypeApi.Delete Method

Deletes a Tax Type from the Cms (see page 7).

### C#

```
public void Delete(long id);
```

### Visual Basic

```
Public Function Delete(id As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of TaxType to delete. |

### TaxTypeApi.GetItem Method

Retrieves a tax type by Id.

**C#**

```
public TaxTypeData GetItem(long id);
```

**Visual Basic**

```
Public Function GetItem(id As long) As TaxTypeData
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id of taxType to retrieve. |

### TaxTypeApi.GetList Method

Retrieves a list of Tax Types.

**C#**

```
public List<TaxTypeData> GetList(Criteria<TaxTypeProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Criteria<TaxTypeProperty>) As List<TaxTypeData>
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | Criteria by which to filter taxType being retrieved. |

**Returns**

List of TaxTypeData (see page 1341) objects.

### TaxTypeApi.Update Method

Updates an existing Tax Type in Cms (see page 7).

**C#**

```
public void Update(TaxTypeData taxTypeData);
```

**Visual Basic**

```
Public Function Update(taxTypeData As TaxTypeData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| taxTypeData | TaxTypeData (see page 1341) object to save. |

## WarehouseApi Class

**Class Hierarchy**

**C#**

```
public class WarehouseApi : CommonApiBase;
```

**Visual Basic**

```
Public Class WarehouseApi
Inherits CommonApiBase
```

**File**

WarehouseApi.cs

**Description**

This is class Ektron.Cms.Commerce.WarehouseApi.

## WarehouseApi Methods

The methods of the WarehouseApi class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Add (see page 637) | This is Add, a member of class WarehouseApi. |
| | Delete (see page 637) | This is Delete, a member of class WarehouseApi. |
| | GetDefaultWarehouse (see page 637) | This is GetDefaultWarehouse, a member of class WarehouseApi. |
| | GetItem (see page 638) | This is GetItem, a member of class WarehouseApi. |
| | GetList (see page 638) | This is GetList, a member of class WarehouseApi. |
| | MarkAsDefault (see page 638) | This is MarkAsDefault, a member of class WarehouseApi. |
| | Update (see page 638) | This is Update, a member of class WarehouseApi. |

### WarehouseApi.Add Method

**C#**

```
public void Add(WarehouseData warehouseData);
```

**Visual Basic**

```
Public Function Add(warehouseData As WarehouseData) As void
```

**Description**

This is Add, a member of class WarehouseApi.

### WarehouseApi.Delete Method

**C#**

```
public void Delete(long warehouseId);
```

**Visual Basic**

```
Public Function Delete(warehouseId As long) As void
```

**Description**

This is Delete, a member of class WarehouseApi.

### WarehouseApi.GetDefaultWarehouse Method

**C#**

```
public WarehouseData GetDefaultWarehouse();
```

**Visual Basic**

```
Public Function GetDefaultWarehouse() As WarehouseData
```

**Description**

This is GetDefaultWarehouse, a member of class WarehouseApi.

### WarehouseApi.GetItem Method

**C#**

```
public WarehouseData GetItem(long warehouseId);
```

**Visual Basic**

```
Public Function GetItem(warehouseId As long) As WarehouseData
```

**Description**

This is GetItem, a member of class WarehouseApi.

### WarehouseApi.GetList Method

**C#**

```
public List<WarehouseData> GetList(Common.Criteria<WarehouseProperty> criteria);
```

**Visual Basic**

```
Public Function GetList(criteria As Common.Criteria<WarehouseProperty>) As List<WarehouseData>
```

**Description**

This is GetList, a member of class WarehouseApi.

### WarehouseApi.MarkAsDefault Method

**C#**

```
public void MarkAsDefault(long warehouseId);
```

**Visual Basic**

```
Public Function MarkAsDefault(warehouseId As long) As void
```

**Description**

This is MarkAsDefault, a member of class WarehouseApi.

### WarehouseApi.Update Method

**C#**

```
public void Update(WarehouseData warehouseData);
```

**Visual Basic**

```
Public Function Update(warehouseData As WarehouseData) As void
```

**Description**

This is Update, a member of class WarehouseApi.

# Ektron.Cms.API Namespace

The API Namespace contains classes for programmatically accessing features typically found in the Workarea.

### Namespaces

| Name | Description |
|---|---|
| Analytics (see page 714) | This is namespace Ektron.Cms.API.Analytics. |
| Community (see page 718) | The Community API contains classes that allow you to manage aspects of your community site. |
| Content (see page 788) | This is namespace Ektron.Cms.API.Content. |
| Search (see page 864) | The Search Developer API enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 694), if you chose to install it, has excellent examples of programmatically using the SearchManager (see page 865). It is installed at http://~YourSite~/CMS400Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#.<br>Here is a list of some of the examples.<br>• Retrieve a document or HTML content that contains the word *cms*.<br>• Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer.<br>• Retrieve images that contain *RotatingGraphic1* in their title.<br>• Retrieve geo-content from CMS400.Net. |
| Task (see page 871) | This is namespace Ektron.Cms.API.Task. |
| UrlAliasing (see page 886) | This is namespace Ektron.Cms.API.UrlAliasing. |
| User (see page 913) | This is namespace Ektron.Cms.API.User. |

### Classes

| | Name | Description |
|---|---|---|
| | BusinessRule (see page 640) | Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example:<br>• **Condition** - a site visitor visits your site for the first time<br>• **Action** - add a cookie to the site visitor's system and pop up a welcome message<br>With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 640) |
| | CustomFields (see page 642) | This is class Ektron.Cms.API.CustomFields. |
| | Folder (see page 652) | This is class Ektron.Cms.API.Folder. |
| | Library (see page 664) | This is class Ektron.Cms.API.Library. |
| | Localization (see page 673) | This is class Ektron.Cms.API.Localization. |
| | Messaging (see page 682) | The Messaging class allows you to work with Private Message Objects. |
| | Metadata (see page 683) | This is class Ektron.Cms.API.Metadata. |
| | Permissions (see page 690) | This is class Ektron.Cms.API.Permissions. |
| | Site (see page 694) | This is class Ektron.Cms.API.Site. |
| | SiteMap (see page 701) | This is class Ektron.Cms.API.SiteMap. |
| | Subscription (see page 705) | This is class Ektron.Cms.API.Subscription. |
| | Utilities (see page 713) | This is class Ektron.Cms.API.Utilities. |

# Classes

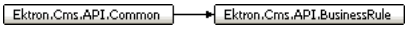
The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | BusinessRule (see page 640) | Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example:<br>• **Condition** - a site visitor visits your site for the first time<br>• **Action** - add a cookie to the site visitor's system and pop up a welcome message<br>With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 640) |
| | CustomFields (see page 642) | This is class Ektron.Cms.API.CustomFields. |
| | Folder (see page 652) | This is class Ektron.Cms.API.Folder. |
| | Library (see page 664) | This is class Ektron.Cms.API.Library. |
| | Localization (see page 673) | This is class Ektron.Cms.API.Localization. |
| | Messaging (see page 682) | The Messaging class allows you to work with Private Message Objects. |
| | Metadata (see page 683) | This is class Ektron.Cms.API.Metadata. |
| | Permissions (see page 690) | This is class Ektron.Cms.API.Permissions. |
| | Site (see page 694) | This is class Ektron.Cms.API.Site. |
| | SiteMap (see page 701) | This is class Ektron.Cms.API.SiteMap. |
| | Subscription (see page 705) | This is class Ektron.Cms.API.Subscription. |
| | Utilities (see page 713) | This is class Ektron.Cms.API.Utilities. |

# BusinessRule Class

### Class Hierarchy




### C#

```
public class BusinessRule : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class BusinessRule
Inherits Ektron.Cms.API.Common
```

### File

BusinessRule.vb

### Description

Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example:

- **Condition** - a site visitor visits your site for the first time
- **Action** - add a cookie to the site visitor's system and pop up a welcome message

With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control to your Web form, and assign the server control the Business Rules Ruleset.

### .NET Assemblies BusinessRule API vs. BusinessRules Server Control API

The .NET Assemblies BusinessRule API (see page 639) allows you to work with results of a Business Rule. The

BusinessRules (see page 66) server control API (see page 639) allows you to customize how and where a Business Rule is used on a Web page.

To learn about the BusinessRules (see page 66) server control API (see page 639), click here (see page 66).

## BusinessRule Methods

The methods of the BusinessRule class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | GetBusinessRules (see page 641) | This is the overview for the GetBusinessRules method overload. |

### GetBusinessRules Method

This is the overview for the GetBusinessRules method overload.

#### Description

Method to get the results from a business rule. This method is overloaded.

#### Overload List

| | Name | Description |
|---|---|---|
| | BusinessRule.GetBusinessRules (String, String, String, String, String, String, String, Long) (see page 641) | 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 642) | Method to get the results from a business rule by identifier. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control. |

### BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, Long)

Method to get the results from a business rule by ID. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset.

#### C#

```
public String GetBusinessRules(String httpHost, String urlRest, String cookies, String
formValues, String queryValues, String serverValues, String paramValues, Long iRuleID);
```

#### Visual Basic

```
Public Function GetBusinessRules(ByVal httpHost As String, ByVal urlRest As String, ByVal
cookies As String, ByVal formValues As String, ByVal queryValues As String, ByVal serverValues
As String, ByVal paramValues As String, ByVal iRuleID As Long) As String
```

#### Parameters

| Parameters | Description |
|---|---|
| httpHost | The HTTP host the request is made on. |
| urlRest | not used |
| cookies | Cookie Value Collection (see page 88) |
| formValues | Form (see page 836) Value Collection (see page 88) |
| queryValues | Querystring Value Collection (see page 88) |
| serverValues | Server Variables Collection (see page 88) |
| paramValues | Custom Value Collection (see page 88) |

| | |
|---|---|
| iRuleID | Numeric value representing the ID of the Ruleset |

**Returns**

A string representing the output of the evaluated ruleset.

### BusinessRule.GetBusinessRules Method (String, String, String, String, String, String, String, String)

Method to get the results from a business rule by identifier. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control.

**C#**

```
public String GetBusinessRules(String httpHost, String urlRest, String cookies, String
formValues, String queryValues, String serverValues, String paramValues, String sRuleID);
```

**Visual Basic**

```
Public Function GetBusinessRules(ByVal httpHost As String, ByVal urlRest As String, ByVal
cookies As String, ByVal formValues As String, ByVal queryValues As String, ByVal serverValues
As String, ByVal paramValues As String, ByVal sRuleID As String) As String
```

**Parameters**

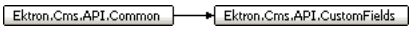
| Parameters | Description |
|---|---|
| httpHost | The HTTP host the request is made on. |
| urlRest | not used |
| cookies | Cookie Value Collection (see page 88) |
| formValues | Form (see page 836) Value Collection (see page 88) |
| queryValues | Querystring Value Collection (see page 88) |
| serverValues | Server Variables Collection (see page 88) |
| paramValues | Custom Value Collection (see page 88) |
| 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 Ektron.Cms.API.CustomFields.

## CustomFields Methods

The methods of the CustomFields class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddNewEditableCustomFieldAssignments (see page 644) | Returns the HTML string for the add/edit custom fields. |
| S | BoolToYesNo (see page 644) | Returns labels for yes or no based on locale. |
| | GetEditableCustomFieldAssignments (see page 644) | Returns the editable custom fields.<br>Break Inheritance checkbox:<br>• Visible if not at root.<br>• Checked if any fields assigned without inheritance.<br>• If unchecked & form saved, removes all assignments.<br>• If checked & form saved and no assignments chosen, creates NULL assignment (CustomFieldID -1).<br>• When checked, enables all assignment checkboxes (iterate through all of this tables elements).<br>• When unchecked, disables all assignment checkboxes (iterate through all of this tables elements).<br>• Assignment Checkboxes:<br>• When clicked: if 'Assignment' button is going unchecked, then ensure required is unchecked.<br>• When clicked: if 'Required' button, don't accept checks if assigned is unchecked.<br>• When form submitted, copy checked... more (see page 644) |
| S | GetEnhancedMetadataArea (see page 645) | Returns the HTML needed to render the EnhancedMetadata selection area. |
| S | GetEnhancedMetadataScript (see page 645) | Returns support Javascript code for enhanced metadata. |
| | GetFieldsByFolder (see page 645) | Returns all of the custom attributes by folder. |
| S | GetSearchProperties (see page 646) | This is the overview for the GetSearchProperties method overload. |
| S | Populate_AssetRequestObjectFromForm (see page 647) | Returns asset request object using form data. |
| | ProcessCustomFields (see page 648) | Process Folders' Custom-Field (and metadata) assignments:<br>When form submitted, copy checked items to hidden field.<br>Form (see page 836) processor:<br>• If break inheritance checked, and any items in hidden field, send contents to db.<br>• If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1.<br>• If break inheritance NOT checked, remove all assignments for this folder. |
| S | SearchAllAssets (see page 648) | Returns the search result based on assets. |
| S | WriteFilteredMetadataForEdit (see page 649) | Returns an HTML string for filtered metadata. |
| S | WriteFilteredMetadataForView (see page 649) | Returns an HTML string for metadata view. |
| S | WriteLibrarySearchExtended (see page 649) | Returns an HTML string for the library search custom metadata properties. |
| S | WriteMetadataDefaultForEdit (see page 650) | Returns an HTML string for the default metadata. |

| | | |
|---|---|---|
| S | WriteMetadataTypeForView (see page 650) | Returns a metadata type for view. |
| S | WritePublicSearch (see page 650) | Returns WritepublicsearchEx. |
| S | WritePublicSearchEx (see page 651) | Returns an HTML string for extended the search attributes. |

### CustomFields.AddNewEditableCustomFieldAssignments Method

Returns the HTML string for the add/edit custom fields.

**C#**

```
public String AddNewEditableCustomFieldAssignments(Long parentfolderid);
```

**Visual Basic**

```
Public Function AddNewEditableCustomFieldAssignments(ByVal parentfolderid As Long) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 836) processor;
- If break inheritance checked, and any items in hidden field, send contents to db.
- If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1.
- If break inheritance NOT checked, remove all assignments for this folder.

Note: If editMode is false, then items are visible but not changeable by user.

**C#**

```
public String GetEditableCustomFieldAssignments(Object folderID, Object editMode);
```

**Visual Basic**

```
Public Function GetEditableCustomFieldAssignments(ByVal folderID As Object, ByVal editMode As
Object) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| 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, int LanguageID);
```

**Visual Basic**

```
Public Function GetFieldsByFolder(ByVal FolderID As Long, ByVal LanguageID As Integer) As
Collection
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The folder's ID. |
| LanguageID | The language's ID. |

**Returns**

Microsoft.VisualBasic.Collection (see page 88)

## GetSearchProperties Method

This is the overview for the GetSearchProperties method overload.

**Description**

Returns the search's properties. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | CustomFields.GetSearchProperties (Ektron.Cms.Common.SearchAssetDispayRequest) (see page 646) | Returns an HTML string for the search properties display. |
| | CustomFields.GetSearchProperties (String, String, String, String, Collection, String, String, String) (see page 647) | Returns an HTML string to display search properties. |
| | CustomFields.GetSearchProperties (System.Collections.Specialized.NameValueCollection, Ektron.Cms.Common.SearchItemCustom, Ektron.Cms.Common.SearchItemCustom) (see page 647) | 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 Object GetSearchProperties(String MonikerStyle, String DataType, String Caption,
String Name, Collection Items, String AlignCaption, String szBeginFontInfo, String
szEndFontInfo);
```

**Visual Basic**

```
Public Shared Function GetSearchProperties(ByVal MonikerStyle As String, ByVal DataType As
String, ByVal Caption As String, ByVal Name As String, ByVal Items As Collection, ByVal
AlignCaption As String, ByVal szBeginFontInfo As String, ByVal szEndFontInfo As String) As
Object
```

**Parameters**

| Parameters | Description |
|---|---|
| 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,
ref Ektron.Cms.Common.SearchItemCustom CustomProps, ref Ektron.Cms.Common.SearchItemCustom
AssetProps);
```

**Visual Basic**

```
Public Shared Sub GetSearchProperties(ByVal FormReq As
System.Collections.Specialized.NameValueCollection, ByRef CustomProps() As
Ektron.Cms.Common.SearchItemCustom, ByRef AssetProps() As Ektron.Cms.Common.SearchItemCustom)
```

**Parameters**

| Parameters | Description |
|---|---|
| FormReq | Form (see page 836) field values. |
| CustomProps | Custom properties value. |
| AssetProps | Asset (see page 789) properties value. |

### CustomFields.Populate_AssetRequestObjectFromForm Method

Returns asset request object using form data.

**C#**

```
public Shared Ektron.Cms.Common.SearchAssetRequest
Populate_AssetRequestObjectFromForm(Optional ByVal
System.Collections.Specialized.NameValueCollection PostBackData);
```

**Visual Basic**

```
Public Shared Function Populate_AssetRequestObjectFromForm(Optional ByVal PostBackData As
System.Collections.Specialized.NameValueCollection = Nothing) As
Ektron.Cms.Common.SearchAssetRequest
```

**Returns**

Ektron.Cms.Common.SearchAssetRequest

## CustomFields.ProcessCustomFields Method

Process Folders' Custom-Field (and metadata) assignments:

When form submitted, copy checked items to hidden field.

Form (see page 836) processor:

- If break inheritance checked, and any items in hidden field, send contents to db.
- If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1.
- If break inheritance NOT checked, remove all assignments for this folder.

**C#**

```
public Object ProcessCustomFields(Long folderID);
```

**Visual Basic**

```
Public Function ProcessCustomFields(ByVal folderID As Long) As Object
```

**Parameters**

| Parameters | Description |
|---|---|
| folderID | The folder's ID. |

**Returns**

Object

**Remarks**

## CustomFields.SearchAllAssets Method

Returns the search result based on assets.

**C#**

```
public Shared System.Text.StringBuilder SearchAllAssets(SearchAssetDispayRequest sadReq);
```

**Visual Basic**

```
Public Shared Function SearchAllAssets(ByVal sadReq As SearchAssetDispayRequest) As
System.Text.StringBuilder
```

**Parameters**

| Parameters | Description |
|---|---|
| sadReq | Search (see page 864) parameters. (Required) |

**Returns**

System.Text.StringBuilder

### CustomFields.WriteFilteredMetadataForEdit Method

Returns an HTML string for filtered metadata.

**C#**

```
public Shared System.Text.StringBuilder
WriteFilteredMetadataForEdit(Ektron.Cms.ContentMetaData cMetadataTypes, Boolean
bSearchableOnly, String ty, Long FolderId, Optional ByRef int Count, Optional ByVal Hashtable
metaPermsObj);
```

**Visual Basic**

```
Public Shared Function WriteFilteredMetadataForEdit(ByVal cMetadataTypes() As
Ektron.Cms.ContentMetaData, ByVal bSearchableOnly As Boolean, ByVal ty As String, ByVal
FolderId As Long, Optional ByRef Count As Integer = 0, Optional ByVal metaPermsObj As
Hashtable = Nothing) As System.Text.StringBuilder
```

**Parameters**

| Parameters | Description |
|---|---|
| cMetadataTypes | Metadata (see page 683) 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 683) types. |
| FolderId | The folder's Id. |
| bSearchableOnly | True or False |

**Returns**

String

### CustomFields.WriteLibrarySearchExtended Method

Returns an HTML string for the library search custom metadata properties.

**C#**

```
public Shared System.Text.StringBuilder WriteLibrarySearchExtended(Long folderId);
```

**Visual Basic**

```
Public Shared Function WriteLibrarySearchExtended(ByVal folderId As Long) As
```

```
System.Text.StringBuilder
```

**Parameters**

| Parameters | Description |
|---|---|
| folderId | The folder's ID. |

**Returns**

System.Text.StringBuilder

### CustomFields.WriteMetadataDefaultForEdit Method

Returns an HTML string for the default metadata.

**C#**

```
public Shared System.Text.StringBuilder WriteMetadataDefaultForEdit(Ektron.Cms.ContentMetaData
cMetadataType, int DataType, Optional ByVal Hashtable metaPermsObj);
```

**Visual Basic**

```
Public Shared Function WriteMetadataDefaultForEdit(ByVal cMetadataType As
Ektron.Cms.ContentMetaData, ByVal DataType As Integer, Optional ByVal metaPermsObj As
Hashtable = Nothing) As System.Text.StringBuilder
```

**Parameters**

| Parameters | Description |
|---|---|
| cMetadataType | Metadata (see page 683) 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.

**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 (see page 864) asset data. (Required) |

### Returns

System.Text.StringBuilder

## CustomFields.WritePublicSearchEx Method

Returns an HTML string for extended the search attributes.

### C#

```
public Shared System.Text.StringBuilder WritePublicSearchEx(SearchAssetDispayRequest sadReq,
Boolean EnableAdvancedLink, Boolean EnableContentBtn, Boolean EnableContent, Boolean
EnableFormsBtn, Boolean EnableForms, Boolean EnableDMSAssetsBtn, Boolean EnableDMSAssets,
Boolean EnableLibImagesBtn, Boolean EnableLibImages, Boolean EnableLibFilesBtn, Boolean
EnableLibFiles, Boolean EnableLibHyperlinksBtn, Boolean EnableLibHyperlinks, Boolean
EnableArchivedBtn, Boolean EnableArchived, String TargetFolderDescriptionList, Optional ByVal
Boolean EnableContentFieldset, Optional ByVal String ContentFieldsetLegend, Optional ByVal
Boolean EnableLibraryFieldset, Optional ByVal String LibraryFieldsetLegend, Optional ByVal
Boolean EnableFolderFieldset, Optional ByVal String FolderFieldsetLegend, Optional ByVal
Boolean EnableCallBack, Optional ByVal Boolean RemoveTeaserHtml, Optional ByVal int
MaxTeaserLength);
```

### Visual Basic

```
Public Shared Function WritePublicSearchEx(ByVal sadReq As SearchAssetDispayRequest, ByVal
EnableAdvancedLink As Boolean, ByVal EnableContentBtn As Boolean, ByVal EnableContent As
Boolean, ByVal EnableFormsBtn As Boolean, ByVal EnableForms As Boolean, ByVal
EnableDMSAssetsBtn As Boolean, ByVal EnableDMSAssets As Boolean, ByVal EnableLibImagesBtn As
Boolean, ByVal EnableLibImages As Boolean, ByVal EnableLibFilesBtn As Boolean, ByVal
EnableLibFiles As Boolean, ByVal EnableLibHyperlinksBtn As Boolean, ByVal EnableLibHyperlinks
As Boolean, ByVal EnableArchivedBtn As Boolean, ByVal EnableArchived As Boolean, ByVal
TargetFolderDescriptionList As String, Optional ByVal EnableContentFieldset As Boolean =
False, Optional ByVal ContentFieldsetLegend As String = "", Optional ByVal
EnableLibraryFieldset As Boolean = False, Optional ByVal LibraryFieldsetLegend As String = "",
Optional ByVal EnableFolderFieldset As Boolean = False, Optional ByVal FolderFieldsetLegend As
String = "", Optional ByVal EnableCallBack As Boolean = False, Optional ByVal RemoveTeaserHtml
As Boolean = True, Optional ByVal MaxTeaserLength As Integer = 100) As
System.Text.StringBuilder
```

### Parameters

| Parameters | Description |
|---|---|
| sadReq | Search (see page 864) 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 |

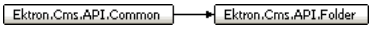
| | |
|---|---|
| EnableLibHyperlinks | True or False |
| EnableArchivedBtn | True or False |
| EnableArchived | True or False |
| TargetFolderDescriptionList | True or False |
| EnableContentFieldset | True or False |
| ContentFieldsetLegend | True or False |
| EnableLibraryFieldset | True or False |
| LibraryFieldsetLegend | True or False |
| EnableFolderFieldset | True or False |
| FolderFieldsetLegend | True or False |

### Returns

System.Text.StringBuilder

# Folder Class

### Class Hierarchy




### C#

```
public class Folder : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class Folder
Inherits Ektron.Cms.API.Common
```

### File

Folder.vb

### Description

This is class Ektron.Cms.API.Folder.

## Folder Methods

The methods of the Folder class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddCommunityFolder (see page 653) | Creates CMS Community (see page 718) folder. |
| | AddFolder (see page 653) | Creates a new folder in the CMS. By default the new folder inherits subscription properties of parent folder. To override, use the BreakInheritance property of the FolderRequest.SubscriptionProperties object (set it to True).<br>*See Also:* The Example section. |
| | CreateFolder (see page 654) | Creates a folder. |
| | DeleteFolderById (see page 655) | Delete the folder by its ID.<br>*See Also:* The Example section. |
| | GetChildFolders (see page 656) | This is the overview for the GetChildFolders method overload. |
| | GetFolder (see page 659) | This is the overview for the GetFolder method overload. |
| | GetFolderDataWithPermission (see page 661) | Retrieves the folder's details with permission information. |
| | GetFolderId (see page 661) | This is the overview for the GetFolderId method overload. |
| | GetFolderInfo (see page 662) | Returns the folder detail. |
| | GetPath (see page 662) | Gets the path of a folder in the Workarea by using the folder's ID. |

| | | |
|---|---|---|
| | IsFolderHidden (see page 662) | Returns flag indicating if the folder is hidden. |
| | RenameFolder (see page 663) | Rename the folder with new name. |
| | UpdateFolder (see page 663) | This is the overview for the UpdateFolder method overload. |
| | UpdateFolderApprovalMethod (see page 664) | Updates the ApprovalMethod of a folder. The ApprovalMethod gets or set whether or not to force all approvers to approve the content that is added to the folder. Before using this method, you must break permission inheritance for the folder by setting the FolderRequest object's BreakInheritButton property to True. |

## Folder.AddCommunityFolder Method

Creates CMS Community (see page 718) folder.

**C#**

```
public AddCommunityFolder(FolderRequest request);
```

**Visual Basic**

```
Public Sub AddCommunityFolder(ByVal request As FolderRequest)
```

**Parameters**

| Parameters | Description |
|---|---|
| request | The Folder (see page 652)'s properties as a FolderRequest object. |

## Folder.AddFolder Method

Creates a new folder in the CMS. By default the new folder inherits subscription properties of parent folder. To override, use the BreakInheritance property of the FolderRequest.SubscriptionProperties object (set it to True).

*See Also:* The Example section.

**C#**

```
public AddFolder(FolderRequest request);
```

**Visual Basic**

```
Public Sub AddFolder(ByVal request As FolderRequest)
```

**Parameters**

| Parameters | Description |
|---|---|
| request | The Folder (see page 652)'s properties as a FolderRequest object. |

**Example**

The following example shows how to create a Web page from which a folder can be added to CMS400.NET. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddFolder method. This method uses the InternalAdmin to add the folder. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
   ```
   <asp:Label ID="Label1" runat="server" Text="Folder Name: " Width="84px"></asp:Label>
   ```
2. Add a text box to enter the folder's name.
   ```
   <asp:TextBox ID="txtFolderName" runat="server" Width="96px"></asp:TextBox>
   ```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder Name box.
   ```
   <asp:RequiredFieldValidator ID="rfvFolderName" runat="server"
ControlToValidate="txtFolderName"
     ErrorMessage="Please enter folder name"></asp:RequiredFieldValidator>
   ```
4. Add a couple of breaks for spacing purposes.
   ```
   <br /><br />
   ```
5. Add a label for the description box.

```
<asp:Label ID="Label2" runat="server" Text="Description: " Width="72px"></asp:Label>
```

6. Add a text box to enter a description.
```
<asp:TextBox ID="txtDescription" runat="server" Width="358px"></asp:TextBox>
```
7. Add two more breaks for spacing purposes.
```
<br /><br />
```
8. Add a button to submit the information. We will set the button click event in the code behind.
```
<asp:Button ID="btnAddFolder" runat="server" Text="Add CMS Folder" Width="110px" />
```
9. Add two last breaks.
```
<br /><br />
```
10. Add a results label that lets a user know if the add succeeded.
```
<asp:Label ID="lblResult" runat="server" Width="545px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
        Protected Sub btnAddFolder_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnAddFolder.Click
```

2. Create an object for the Folder (see page 652) API (see page 639).

```
            Dim folderApi As New Ektron.Cms.API.Folder
```

3. Create a Try/Catch around the AddFolder method. This allows you to catch any exceptions and helps when debugging.

4. Create a FolderRequest object and assign the following properties:

```
            Try
                Dim req As New Ektron.Cms.FolderRequest
                req.FolderName = txtFolderName.Text
                req.FolderDescription = txtDescription.Text
                req.ParentId = 0  'Add this folder under the root folder in the Workarea
                req.StyleSheet = ""
                folderApi.AddFolder(req)
```

5. Display the folder name and ID and the fact that it was added.

```
                Dim newFolderID As Integer = req.FolderId
                lblResult.Text = "Folder " & txtFolderName.Text & " was added under the
root folder in the Workarea with FolderID = " & newFolderId.ToString()
```

6. If there is an error, let the user know what happened.

```
            Catch ex As Exception
                lblResult.Text = ex.Message
            End Try

    End Sub
```

### Folder.CreateFolder Method

Creates a folder.

**C#**

```
public Overridable Boolean CreateFolder(String Path);
```

**Visual Basic**

```
Public Overridable Function CreateFolder(ByVal Path As String) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| Path | The path on the IIS server. (Required) |

**Returns**

Boolean

## Folder.DeleteFolderById Method

Delete the folder by its ID.

*See Also:* The Example section.

**C#**

```
public Boolean DeleteFolderById(Long Id);
```

**Visual Basic**

```
Public Function DeleteFolderById(ByVal Id As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The folder's ID. |

**Returns**

False = successfully deleted. Otherwise, returns True.

**Example**

The following example shows how to create a Web page from which a folder can be deleted from CMS400.NET by entering its ID. This example uses some standard drag and drop controls and a small section of VB code utilizing the DeleteFolderById method. This method uses the InternalAdmin to delete the folder. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
   ```
   <asp:Label ID="Label1" runat="server" Text="Folder ID: "></asp:Label>
   ```
2. Add a text box to enter the folder's ID.
   ```
   <asp:TextBox ID="txtFolderId" runat="server" Width="44px"></asp:TextBox>
   ```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
   ```
   <asp:RequiredFieldValidator ID="rfvFolderId" runat="server" ControlToValidate="txtFolderId"
     ErrorMessage="Please enter folder ID"></asp:RequiredFieldValidator>
   ```
4. Add a couple of breaks for spacing purposes.
   ```
   <br /><br />
   ```
5. Add a delete button. We will set the button click event in the code behind.
   ```
   <asp:Button ID="btnDeleteFolder" runat="server" Text="Delete Folder" Width="95px" />
   ```
6. Add two more breaks for spacing purposes.
   ```
   <br /><br />
   ```
7. Add a label to let the user know if the delete succeeded.
   ```
   <asp:Label ID="lblResult" runat="server" Width="346px"></asp:Label>
   ```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
        Protected Sub btnDeleteFolder_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnDeleteFolder.Click
```

2. Check to make sure the folder ID entered is numeric.

```
            If Not IsNumeric(txtFolderId.Text) Then
                lblResult.Text = "Invalid Folder ID"
                Exit Sub
            End If
```

3. Create an object for the Folder (see page 652) API (see page 639).

```
            Dim folderApi As New Ektron.Cms.API.Folder
```

4. Create a Try/Catch around the DeleteFolderById method. This allows you to catch any exceptions and helps when debugging.

```
            Try
```

```
folderApi.DeleteFolderById(CInt(txtFolderId.Text))
```

5. Display whether the deletion was successful.

```
lblResult.Text = "The above folder was deleted from CMS400.NET"
```

6. If there is an error, let the user know.

```
Catch ex As Exception
    lblResult.Text = "Error deleting folder"
End Try

End Sub
```

## GetChildFolders Method

This is the overview for the GetChildFolders method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | Folder.GetChildFolders (Long, Boolean, EkEnumeration.FolderOrderBy) (see page 656) | Loads all the subfolders for a given folder ID.<br>*See Also:* The Example section. |
| | Folder.GetChildFolders (Long, Boolean, String) (see page 658) | Loads all the subfolders for a given folder ID.<br>*See Also:* The Example section. |

## Folder.GetChildFolders Method (Long, Boolean, EkEnumeration.FolderOrderBy)

Loads all the subfolders for a given folder ID.

*See Also:* The Example section.

### C#

```
public FolderData[] GetChildFolders(Long id, Boolean recursive, EkEnumeration.FolderOrderBy
orderBy);
```

### Visual Basic

```
Public Function GetChildFolders(ByVal id As Long, ByVal recursive As Boolean, ByVal orderBy As
EkEnumeration.FolderOrderBy) As FolderData()
```

### Parameters

| Parameters | Description |
|---|---|
| id | The folder's ID |
| recursive | True = recursive look up. False = none. |
| orderBy | Field to order results by. |

### Returns

Ektron.Cms.FolderData()

### Example

The following example shows how to create a Web page from which a folder ID can be entered and list of child folders returned. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetChildFolders method. This method uses the InternalAdmin to retrieve the child folder list. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:

```
<asp:Label ID="Label1" runat="server" Text="Folder ID: "></asp:Label>
```

2. Add a text box to enter the folder's ID.
```
<asp:TextBox ID="txtFolderId" runat="server" Width="44px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
```
<asp:RequiredFieldValidator ID="rfvFolderId" runat="server" ControlToValidate="txtFolderId"
  ErrorMessage="Please enter folder ID"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add a submit button. We will set the button click event in the code behind.
```
<asp:Button ID="btnGetChildFolders" runat="server" Text="Get Child Folders" Width="123px" />
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a label to display the list of child folder names.
```
<asp:Label ID="lblResult" runat="server" Width="272px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
        Protected Sub btnGetChildFolders_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnGetChildFolders.Click
```
2. Create an object for the Folder (see page 652) API (see page 639).
```
            Dim folderApi As New Ektron.Cms.API.Folder
```
3. Check to make sure the folder ID is numeric.
```
            If Not IsNumeric(txtFolderId.Text) Then
                lblResult.Text = "Invalid Folder ID"
                Exit Sub
            End If
```
4. Create an object as an array of FolderData.
```
            Dim folders As Ektron.Cms.FolderData()
```
5. Create a Try/Catch around the GetChildFolders method. This allows you to catch any exceptions and helps when debugging.
```
            Try
                folders = folderApi.GetChildFolders(CInt(txtFolderId.Text), True,
FolderOrderBy.Name)
```
6. Use an If/Else statement to display whether child folders exist.
```
                If (folders Is Nothing) Then
                    lblResult.Text &= "Child folders: None"
```
7. If child folders exist, display a total count.
```
                Else
                    lblResult.Text &= "Child folders: " & folders.Length.ToString & "<br/>"
```
8. Iterate through the array and display each child folder's name.
```
                    Dim i As Integer
                    For i = 0 To folders.Length - 1 Step 1
                        lblResult.Text &= folders(i).Name & "<br/>"
                    Next
                End If
```
9. Display a message if there is an error.
```
            Catch ex As Exception
                lblResult.Text = "Error getting child folders"
            End Try

    End Sub
```
10. Add a page load event so that lblResult.Text is blank when the page first loads.
```
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
```

```
        lblResult.Text = ""
    End Sub
```

### Folder.GetChildFolders Method (Long, Boolean, String)

Loads all the subfolders for a given folder ID.

*See Also:* The Example section.

**C#**

```
[Obsolete("Use new GetChildFolders with FolderOrderBy parameter.")]
public FolderData[] GetChildFolders(Long Id, Optional ByVal Boolean Recursive, Optional ByVal
String OrderBy);
```

**Visual Basic**

```
<Obsolete("Use new GetChildFolders with FolderOrderBy parameter.")>
Public Function GetChildFolders(ByVal Id As Long, Optional ByVal Recursive As Boolean = False,
Optional ByVal OrderBy As String = "name") As FolderData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The folder's ID |
| Recursive | True = recursive look up. False = none. |
| OrderBy | An order by clause |

**Returns**

Ektron.Cms.FolderData()

**Example**

The following example shows how to create a Web page from which a folder ID can be entered and list of child folders returned. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetChildFolders method. This method uses the InternalAdmin to retrieve the child folder list. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
   `<asp:Label ID="Label1" runat="server" Text="Folder ID: "></asp:Label>`
2. Add a text box to enter the folder's ID.
   `<asp:TextBox ID="txtFolderId" runat="server" Width="44px"></asp:TextBox>`
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
   ```
   <asp:RequiredFieldValidator ID="rfvFolderId" runat="server" ControlToValidate="txtFolderId"
     ErrorMessage="Please enter folder ID"></asp:RequiredFieldValidator>
   ```
4. Add a couple of breaks for spacing purposes.
   `<br /><br />`
5. Add a submit button. We will set the button click event in the code behind.
   `<asp:Button ID="btnGetChildFolders" runat="server" Text="Get Child Folders" Width="123px" />`
6. Add two more breaks for spacing purposes.
   `<br /><br />`
7. Add a label to display the list of child folder names.
   `<asp:Label ID="lblResult" runat="server" Width="272px"></asp:Label>`

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
        Protected Sub btnGetChildFolders_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles btnGetChildFolders.Click
```

2. Create an object for the Folder (see page 652) API (see page 639).

```
            Dim folderApi As New Ektron.Cms.API.Folder
```

3. Check to make sure the folder ID is numeric.

```
If Not IsNumeric(txtFolderId.Text) Then
    lblResult.Text = "Invalid Folder ID"
    Exit Sub
End If
```

4. Create an object as an array of FolderData.

```
Dim folders As Ektron.Cms.FolderData()
```

5. Create a Try/Catch around the GetChildFolders method. This allows you to catch any exceptions and helps when debugging.

```
Try
    folders = folderApi.GetChildFolders(CInt(txtFolderId.Text), True, "name")
```

6. Use an If/Else statement to display whether child folders exist.

```
    If (folders Is Nothing) Then
        lblResult.Text &= "Child folders: None"
```

7. If child folders exist, display a total count.

```
    Else
        lblResult.Text &= "Child folders: " & folders.Length.ToString & "<br/>"
```

8. Iterate through the array and display each child folder's name.

```
        Dim i As Integer
        For i = 0 To folders.Length - 1 Step 1
            lblResult.Text &= folders(i).Name & "<br/>"
        Next
    End If
```

9. Display a message if there is an error.

```
    Catch ex As Exception
        lblResult.Text = "Error getting child folders"
    End Try

End Sub
```

10. Add a page load event so that lblResult.Text is blank when the page first loads.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    lblResult.Text = ""
End Sub
```

## GetFolder Method

This is the overview for the GetFolder method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | Folder.GetFolder (Long) (see page 659) | Retrieves the folder's details. |
| | Folder.GetFolder (Long, Boolean) (see page 660) | Retrieves the folder's details with taxonomy information. |
| | Folder.GetFolder (Long, Boolean, Boolean) (see page 660) | Retrieves the folder's details with taxonomy information and Flag (see page 1044) information |

## Folder.GetFolder Method (Long)

Retrieves the folder's details.

**C#**

```
public FolderData GetFolder(Long Id);
```

**Visual Basic**

```
Public Function GetFolder(ByVal Id As Long) As FolderData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The Folder (see page 652)'s ID. |

**Returns**

Ektron.Cms.FolderData

**Remarks**

Read permission required

### Folder.GetFolder Method (Long, Boolean)

Retrieves the folder's details with taxonomy information.

**C#**

```
public FolderData GetFolder(Long Id, Boolean GetTaxonomy);
```

**Visual Basic**

```
Public Function GetFolder(ByVal Id As Long, ByVal GetTaxonomy As Boolean) As FolderData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The Folder (see page 652)'s ID. |
| GetTaxonomy | If true, gets the Taxonomy (see page 842) associated with the folder information. |

**Returns**

Ektron.Cms.FolderData

### Folder.GetFolder Method (Long, Boolean, Boolean)

Retrieves the folder's details with taxonomy information and Flag (see page 1044) information

**C#**

```
public FolderData GetFolder(Long Id, Boolean GetTaxonomy, Boolean GetFlags);
```

**Visual Basic**

```
Public Function GetFolder(ByVal Id As Long, ByVal GetTaxonomy As Boolean, ByVal GetFlags As
Boolean) As FolderData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The Folder (see page 652)'s ID. |
| GetTaxonomy | If true, gets the Taxonomy (see page 842) associated with the folder information. |
| GetFlags | If true, gets the Flag (see page 1044) definitions associated with the folder |

**Returns**

Ektron.Cms.FolderData

## Folder.GetFolderDataWithPermission Method

Retrieves the folder's details with permission information.

**C#**

```
public FolderData GetFolderDataWithPermission(Long Id);
```

**Visual Basic**

```
Public Function GetFolderDataWithPermission(ByVal Id As Long) As FolderData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The folder's numeric ID. |

**Returns**

Ektron.Cms.FolderData

## GetFolderId Method

This is the overview for the GetFolderId method overload.

**Description**

This method is overloaded. Finds the folder ID by using the content's ID. In addition, you could choose to find the folder ID by using the folder's name and naming the parent folder's ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Folder.GetFolderId (Long) (see page 661) | Finds the folder ID by using a content's ID. |
| | Folder.GetFolderId (String, Long) (see page 661) | Finds the folder ID by specifying its folder name and the parent folder ID. |

## Folder.GetFolderId Method (Long)

Finds the folder ID by using a content's ID.

**C#**

```
public Long GetFolderId(Long Id);
```

**Visual Basic**

```
Public Function GetFolderId(ByVal Id As Long) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The content's numeric ID. |

**Returns**

A numeric value representing the folder ID.

## Folder.GetFolderId Method (String, Long)

Finds the folder ID by specifying its folder name and the parent folder ID.

**C#**

```
public Long GetFolderId(String FolderName, Long ParentId);
```

**Visual Basic**

```
Public Function GetFolderId(ByVal FolderName As String, ByVal ParentId As Long) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderName | Name of the folder. |
| ParentId | The numeric ID of the parent folder. |

**Returns**

Returns the folder ID. Returns 0 if it does not exist.

### Folder.GetFolderInfo Method

Returns the folder detail.

**C#**

```
public Overridable String[] GetFolderInfo(String Path);
```

**Visual Basic**

```
Public Overridable Function GetFolderInfo(ByVal Path As String) As String()
```

**Parameters**

| Parameters | Description |
|---|---|
| Path | The folder's path. |

**Returns**

String()

### Folder.GetPath Method

Gets the path of a folder in the Workarea by using the folder's ID.

**C#**

```
public String GetPath(Long Id);
```

**Visual Basic**

```
Public Function GetPath(ByVal Id As Long) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The folder's ID. |

**Returns**

Returns the folder path as a string. For example: `\~SiteRoot~\Products\Medication Management`.

### Folder.IsFolderHidden Method

Returns flag indicating if the folder is hidden.

**C#**

```
public Boolean IsFolderHidden(Long folderId);
```

**Visual Basic**

```
Public Function IsFolderHidden(ByVal folderId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| folderId | The ID of the folder that you are checking. |

**Returns**

Returns True if the folder is hidden, otherwise returns False.

## Folder.RenameFolder Method

Rename the folder with new name.

**C#**

```
public Boolean RenameFolder(Long cId, String title);
```

**Visual Basic**

```
Public Function RenameFolder(ByVal cId As Long, ByVal title As String) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| cId | The folder's ID. |
| title | The new folder's name. |

**Returns**

Returns False if successfully renamed. Otherwise, it returns True.

## UpdateFolder Method

This is the overview for the UpdateFolder method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Folder.UpdateFolder (FolderData) (see page 663) | Updates the CMS folder's information. |
| | Folder.UpdateFolder (FolderRequest) (see page 664) | Updates the CMS folder's information. |

## Folder.UpdateFolder Method (FolderData)

Updates the CMS folder's information.

**C#**

```
public UpdateFolder(FolderData request);
```

**Visual Basic**

```
Public Sub UpdateFolder(ByVal request As FolderData)
```

**Parameters**

| Parameters | Description |
|---|---|
| request | The Folder (see page 652)'s properties as a FolderData object. |

### Folder.UpdateFolder Method (FolderRequest)

Updates the CMS folder's information.

**C#**

```
public UpdateFolder(FolderRequest request);
```

**Visual Basic**

```
Public Sub UpdateFolder(ByVal request As FolderRequest)
```

**Parameters**

| Parameters | Description |
|---|---|
| request | The Folder (see page 652)'s properties as a FolderRequest object. |

### Folder.UpdateFolderApprovalMethod Method

Updates the ApprovalMethod of a folder. The ApprovalMethod gets or set whether or not to force all approvers to approve the content that is added to the folder. Before using this method, you must break permission inheritance for the folder by setting the FolderRequest object's BreakInheritButton property to True.

**C#**

```
public UpdateFolderApprovalMethod(FolderData request);
```

**Visual Basic**

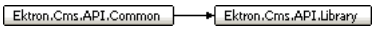
```
Public Sub UpdateFolderApprovalMethod(ByVal request As FolderData)
```

**Parameters**

| Parameters | Description |
|---|---|
| request | The Folder (see page 652)'s properties as a FolderData object. |

## Library Class

**Class Hierarchy**




**C#**

```
public class Library : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Library
Inherits Ektron.Cms.API.Common
```

**File**

Library.vb

**Description**

This is class Ektron.Cms.API.Library.

## Library Methods

The methods of the Library class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddLibraryItem (see page 665) | Adds a library item's information to the database. It does not actually upload the item to the library. |
| | AddLoadBalanceItem (see page 665) | Adds a load balance path for files and images. |
| | DeleteLibraryItem (see page 666) | Deletes the library item by extension. |
| | DeleteLibraryItemById (see page 666) | Deletes the library items from the system. |
| | DeleteLoadBalanceItem (see page 666) | Deletes the load balance item from the system. |
| | GetAllChildLibItems (see page 666) | Loads all of the library items by type. Using the OrderBy parameter allows you to sort the library items. The currentPageNum, pageSize and totalPages parameters are used with paging. Setting currentPageNum and pageSize to zero causes the API (see page 639) To retrieve all items on a single page.<br>**Note**: The caller of API (see page 639) must actually call the API (see page 639) with the correct page number to implement paging. The API (see page 639) does not provide any inherent UI for paging purposes. |
| | GetAllLoadBalancePaths (see page 668) | This is the overview for the GetAllLoadBalancePaths method overload. |
| | GetLibraryItem (see page 669) | This is the overview for the GetLibraryItem method overload. |
| | GetLibrarySettings (see page 670) | Loads the library settings. |
| | GetLibraryTypes (see page 671) | Loads the library types. |
| | GetLoadBalancePath (see page 671) | Gets the load balance path by ID. |
| | InternalLibrarySearch (see page 671) | Searches library items. |
| | UpdateLibrary (see page 672) | Updates a library item. |
| | UpdateLibrarySettings (see page 672) | Update the library settings. |
| | UpdateLoadBalanceSettings (see page 672) | Updates the load balance settings. |
| | UpdateQlinkTemplates (see page 672) | Updates the Quicklink template. |

## Library.AddLibraryItem Method

Adds a library item's information to the database. It does not actually upload the item to the library.

**C#**

```
public Long AddLibraryItem(ref LibraryData libraryData);
```

**Visual Basic**

```
Public Function AddLibraryItem(ByRef libraryData As LibraryData) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| libraryData | Details of the new library item |

## Library.AddLoadBalanceItem Method

Adds a load balance path for files and images.

**C#**

```
public AddLoadBalanceItem(String Type, Collection pagedata);
```

**Visual Basic**

```
Public Sub AddLoadBalanceItem(ByVal Type As String, ByVal pagedata As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| Type | The path type, either images or files. |
| pagedata | Load balance item values for path and relative. |

## Library.DeleteLibraryItem Method

Deletes the library item by extension.

**C#**

```
public DeleteLibraryItem(Collection pagedata);
```

**Visual Basic**

```
Public Sub DeleteLibraryItem(ByVal pagedata As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | Library (see page 664) item data |

## Library.DeleteLibraryItemById Method

Deletes the library items from the system.

**C#**

```
public DeleteLibraryItemById(Collection pagedata);
```

**Visual Basic**

```
Public Sub DeleteLibraryItemById(ByVal pagedata As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | Library (see page 664) item data |

## Library.DeleteLoadBalanceItem Method

Deletes the load balance item from the system.

**C#**

```
public DeleteLoadBalanceItem(Long Id);
```

**Visual Basic**

```
Public Sub DeleteLoadBalanceItem(ByVal Id As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The load balance item ID |

## Library.GetAllChildLibItems Method

Loads all of the library items by type. Using the OrderBy parameter allows you to sort the library items. The currentPageNum, pageSize and totalPages parameters are used with paging. Setting currentPageNum and pageSize to zero causes the API (see page 639) To retrieve all items on a single page.

**Note**: The caller of API (see page 639) must actually call the API (see page 639) with the correct page number to implement paging. The API (see page 639) does not provide any inherent UI for paging purposes.

**C#**

```
public LibraryData[] GetAllChildLibItems(String Type, Long ParentId, String OrderBy, int
currentPageNum, int pageSize, ref int totalPages);
```

**Visual Basic**

```
Public Function GetAllChildLibItems(ByVal Type As String, ByVal ParentId As Long, ByVal
OrderBy As String, ByVal currentPageNum As Integer, ByVal pageSize As Integer, ByRef
totalPages As Integer) As LibraryData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Type | The library item's type. (Required) Use one of the following:<br>• images<br>• files<br>• hyperlinks<br>• quicklinks<br>• forms |
| ParentId | The folder's ID. |
| OrderBy | The order to load the library items. You can pass:<br>• librarytitle<br>• libraryid<br>• libraryfilename<br>• date |
| currentPageNum | The current page number to display. |
| pageSize | The page size set by the caller of the API (see page 639). |
| totalPages | The total number of pages required to display all of the items. |

**Returns**

Ektron.Cms.LibraryData()

**Example**

The following example shows how to create a Web page that displays a graphical list of all images stored in the library for a specific folder in the Workarea. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetAllChildLibItems method. This method uses the InternalAdmin to retrieve the images. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
   ```
   <asp:Label ID="Label6" runat="server" Text="Folder ID:" Width="76px"></asp:Label>
   ```
2. Add a text box for the Folder ID.
   ```
   <asp:TextBox ID="txtFolderId" runat="server" Width="55px"></asp:TextBox>
   ```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
   ```
   <asp:RequiredFieldValidator ID="rfvId" runat="server" ControlToValidate="txtFolderId"
       ErrorMessage="Please enter a folder ID"></asp:RequiredFieldValidator>
   ```
4. Add a couple of breaks for spacing purposes.
   ```
   <br /><br />
   ```
5. Add a submit button. We will set the button click event in the code behind.
   ```
   <asp:Button ID="btnGet" runat="server" Text="Get Images" Width="74px" />
   ```
6. Add two more breaks for spacing purposes.
   ```
   <br /><br />
   ```
7. Add a label to display the list of images.
   ```
   <asp:Label ID="lblResult" runat="server"></asp:Label>
   ```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
        Protected Sub btnGet_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGet.Click
```

2. Check to make sure the folder ID is numeric.

```
            If Not IsNumeric(txtFolderId.Text) Then
                lblResult.Text = "Invalid Folder ID"
                Exit Sub
            End If
```

3. Create an object for the library API (see page 639).

```
            Dim libraryApi As New Ektron.Cms.API.Library
```

4. Create an array to contain the library data.

```
            Dim data As Ektron.Cms.LibraryData()
```

5. Create a Try/Catch around the GetAllChildLibItems method. This allows you to catch any exceptions and helps when debugging.

```
            Try
                data = libraryApi.GetAllChildLibItems("images", CInt(txtFolderId.Text), "",
0, 0, 0)
```

6. Check to make sure the library has images for the folder.

```
                If (Not data Is Nothing) AndAlso (data.Length > 0) Then
```

7. If so, create an object for each library item.

```
                    Dim item As Ektron.Cms.LibraryData
```

8. Create an empty string.

```
                    Dim str As String = ""
```

9. Iterate through each item and display the image with the image title as the alt text.

```
                     For Each item In data
                         str &= "<img src=""" & item.FileName & """ alt=""" & item.Title &
"""/>" & "<br/>" & "<br/>"
                         lblResult.Text = str
                     Next
```

10. If the library contains no images for the folder, display a message.

```
                 Else
                     lblResult.Text = "No images found"
                 End If
```

11. Display a message if there is an error.

```
            Catch ex As Exception
                lblResult.Text = ex.Message
            End Try

    End Sub
```

### GetAllLoadBalancePaths Method

This is the overview for the GetAllLoadBalancePaths method overload.

#### Description

Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | Library.GetAllLoadBalancePaths (Long, String) (see page 669) | Returns an array of LoadBalanceData. |
| | Library.GetAllLoadBalancePaths (String) (see page 669) | Returns all the load balance paths as a array of LoadBalanceData. |

## Library.GetAllLoadBalancePaths Method (Long, String)

Returns an array of LoadBalanceData.

**C#**

```
public LoadBalanceData[] GetAllLoadBalancePaths(Long Id, String Type);
```

**Visual Basic**

```
Public Function GetAllLoadBalancePaths(ByVal Id As Long, ByVal Type As String) As
LoadBalanceData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | Load balance ID |
| Type | Use images or files |

**Returns**

Ektron.Cms.LoadBalanceData()

## Library.GetAllLoadBalancePaths Method (String)

Returns all the load balance paths as a array of LoadBalanceData.

**C#**

```
public LoadBalanceData[] GetAllLoadBalancePaths(String Type);
```

**Visual Basic**

```
Public Function GetAllLoadBalancePaths(ByVal Type As String) As LoadBalanceData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Type | Type (images or files) |

**Returns**

Ektron.Cms.LoadBalanceData()

## GetLibraryItem Method

This is the overview for the GetLibraryItem method overload.

**Description**

Loads the library item's data. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | Library.GetLibraryItem (Long) (see page 670) | Loads the library item data. |
| | Library.GetLibraryItem (Long, Long) (see page 670) | Loads the library item's data. |

## Library.GetLibraryItem Method (Long)

Loads the library item data.

**C#**

```
public LibraryData GetLibraryItem(Long LibID);
```

**Visual Basic**

```
Public Function GetLibraryItem(ByVal LibID As Long) As LibraryData
```

**Parameters**

| Parameters | Description |
|---|---|
| LibID | The library ID |

**Returns**

Ektron.Cms.LibraryData

## Library.GetLibraryItem Method (Long, Long)

Loads the library item's data.

**C#**

```
public LibraryData GetLibraryItem(Long Id, Long ParentId);
```

**Visual Basic**

```
Public Function GetLibraryItem(ByVal Id As Long, ByVal ParentId As Long) As LibraryData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The library's ID |
| ParentId | The folder's ID |

**Returns**

Ektron.Cms.LibraryData

## Library.GetLibrarySettings Method

Loads the library settings.

**C#**

```
public LibraryConfigData GetLibrarySettings(Optional ByVal Long Id);
```

**Visual Basic**

```
Public Function GetLibrarySettings(Optional ByVal Id As Long = -1) As LibraryConfigData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The folder's ID |

**Returns**

Ektron.Cms.LibraryConfigData

### Library.GetLibraryTypes Method

Loads the library types.

**C#**

```
public LibraryTypeData[] GetLibraryTypes(Long Id, String OrderBy);
```

**Visual Basic**

```
Public Function GetLibraryTypes(ByVal Id As Long, ByVal OrderBy As String) As LibraryTypeData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The folder's ID |
| OrderBy | Order by clause |

**Returns**

Object

### Library.GetLoadBalancePath Method

Gets the load balance path by ID.

**C#**

```
public LoadBalanceData GetLoadBalancePath(Long Id);
```

**Visual Basic**

```
Public Function GetLoadBalancePath(ByVal Id As Long) As LoadBalanceData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | the load balance path's ID |

**Returns**

Ektron.Cms.LoadBalanceData

### Library.InternalLibrarySearch Method

Searches library items.

**C#**

```
public SearchResultData[] InternalLibrarySearch(Collection pagedata, Optional ByVal
Ektron.Cms.Common.SearchAssetRequest sar);
```

**Visual Basic**

```
Public Function InternalLibrarySearch(ByVal pagedata As Collection, Optional ByVal sar As
Ektron.Cms.Common.SearchAssetRequest = Nothing) As SearchResultData()
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | Search (see page 864) criteria |
| sar | Search (see page 864) asset criteria |

**Returns**

Ektron.Cms.SearchResultData()

### Library.UpdateLibrary Method

Updates a library item.

**C#**

```
public UpdateLibrary(Collection pagedata);
```

**Visual Basic**

```
Public Sub UpdateLibrary(ByVal pagedata As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | Library (see page 664) item data |

### Library.UpdateLibrarySettings Method

Update the library settings.

**C#**

```
public UpdateLibrarySettings(Collection pagedata);
```

**Visual Basic**

```
Public Sub UpdateLibrarySettings(ByVal pagedata As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | Library (see page 664) settings data |

### Library.UpdateLoadBalanceSettings Method

Updates the load balance settings.

**C#**

```
public UpdateLoadBalanceSettings(String Id, Collection pagedata);
```

**Visual Basic**

```
Public Sub UpdateLoadBalanceSettings(ByVal Id As String, ByVal pagedata As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The load balance data |
| pagedata | The load balance items |

### Library.UpdateQlinkTemplates Method

Updates the Quicklink template.

**C#**

```
public UpdateQlinkTemplates(Collection pagedata);
```

### Visual Basic

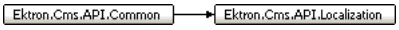
```
Public Sub UpdateQlinkTemplates(ByVal pagedata As Collection)
```

### Parameters

| Parameters | Description |
|---|---|
| pagedata | Quicklink template data |

# Localization Class

### Class Hierarchy




### C#

```
public class Localization : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class Localization
Inherits Ektron.Cms.API.Common
```

### File

Localization.vb

### Description

This is class Ektron.Cms.API.Localization.

## Localization Methods

The methods of the Localization class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | GetAddViewLanguage (see page 674) | Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content ID or whether it does not exist in that language (Type="ADD"). |
| | GetAddViewLanguageForAllContent (see page 674) | Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content block or whether no content exists in that language (Type="ADD"). |
| | GetAddViewLanguageForMenus (see page 674) | Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one menu or whether no menus exist in that language (Type="ADD"). |
| | GetContentItems (see page 675) | Returns a list of content items (i.e., XLIFF skeletons) associated with a given job. |
| | GetFile (see page 675) | Returns the specified file record. |
| | GetFiles (see page 675) | Returns a list of files associated with a given job. |
| | GetFilesInZipFile (see page 676) | Returns a list of files in a given ZIP file. |
| | GetFlagUrl (see page 676) | This is the overview for the GetFlagUrl method overload. |
| | GetInterchangeFile (see page 678) | Returns the interchange (i.e., XLIFF) file record specified by ID. |
| | GetInterchangeFiles (see page 678) | Returns a list of interchange files (i.e., XLIFF files) associated with a given content item (i.e., XLIFF Skeleton). |
| | GetInterchangeFilesInZipFile (see page 678) | Returns a list of interchange files (i.e., XLIFF files) contained within a given ZIP file. |
| | GetJobs (see page 679) | This is the overview for the GetJobs method overload. |
| | GetLocalizationUrl (see page 680) | Returns a URL to the base directory where localization files are stored. `For example, /~SiteRoot~/uploadedfiles/localization.` |
| | GetTranslationUploadDirectory (see page 680) | Returns the server directory path to upload XLIFF and ZIP files for importing. For example, `C:\inetpub\wwwroot\~SiteRoot~\workarea\uploadedfiles\localization`.<br>See Also: StartImportTranslation. |

| | | |
|---|---|---|
| | StartExportContentForTranslation (see page 680) | Starts the process of exporting the content specified by ListOfContentIDs. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text and documents are exported as XLIFF files compressed in ZIP files. The XLIFF files are suitable for sending to a translator. Once translated, the XLIFF files should be imported using the StartImportTranslation method. |
| | StartExportFolderForTranslation (see page 681) | Starts the process of exporting all the content specified by `ContentType` in the specified folder. Optionally, all the content in the subfolders may be processed as well. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text and documents are exported as XLIFF files compressed in ZIP files. The XLIFF files are suitable for sending to a translator. Once translated, the XLIFF files should be imported using the StartImportTranslation. |
| | StartExportMenusForTranslation (see page 681) | 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 682) | Starts the process of importing XLIFF files that have been translated. The files should be uploaded to server directory specified by GetTranslationUploadDirectory. Files may be XLIFF files (.xlf extension) or XLIFF files that are compressed in ZIP (.zip extension) files. |

## Localization.GetAddViewLanguage Method

Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content ID or whether it does not exist in that language (Type="ADD").

**C#**

```
public LanguageData[] GetAddViewLanguage(Long Id);
```

**Visual Basic**

```
Public Function GetAddViewLanguage(ByVal Id As Long) As LanguageData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | Content (see page 788) ID |

**Returns**

Ektron.Cms.LanguageData()

## Localization.GetAddViewLanguageForAllContent Method

Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content block or whether no content exists in that language (Type="ADD").

**C#**

```
public LanguageData[] GetAddViewLanguageForAllContent();
```

**Visual Basic**

```
Public Function GetAddViewLanguageForAllContent() As LanguageData()
```

**Returns**

LanguageData()

**Remarks**

Only Id, Name, BrowserCode and Type are populated.

## Localization.GetAddViewLanguageForMenus Method

Returns an array of all the active content languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one menu or whether no menus exist in that language (Type="ADD").

**C#**

```
public LanguageData[] GetAddViewLanguageForMenus();
```

**Visual Basic**

```
Public Function GetAddViewLanguageForMenus() As LanguageData()
```

**Returns**

LanguageData()

**Remarks**

Only Id, Name, BrowserCode and Type are populated.

### Localization.GetContentItems Method

Returns a list of content items (i.e., XLIFF skeletons) associated with a given job.

**C#**

```
public DS.LocalizationSkeletonDataTable GetContentItems(Long JobID);
```

**Visual Basic**

```
Public Function GetContentItems(ByVal JobID As Long) As DS.LocalizationSkeletonDataTable
```

**Parameters**

| Parameters | Description |
|---|---|
| JobID | Specifies the job. |

**Returns**

A LocalizationJobDataSet.LocalizationSkeletonDataTable of content (i.e., skeleton) items.

### Localization.GetFile Method

Returns the specified file record.

**C#**

```
public DS.LocalizationJobFileRow GetFile(Long FileID);
```

**Visual Basic**

```
Public Function GetFile(ByVal FileID As Long) As DS.LocalizationJobFileRow
```

**Parameters**

| Parameters | Description |
|---|---|
| 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(Long JobID);
```

**Visual Basic**

```
Public Function GetFiles(ByVal JobID As Long) As DS.LocalizationJobFileDataTable
```

**Parameters**

| Parameters | Description |
|---|---|
| JobID | Specifies the job. |

**Returns**

A LocalizationJobDataSet.LocalizationJobFileDataTable of files.

**Remarks**

Files are typically ZIP files for export jobs and ZIP and XLIFF files for import jobs.

## Localization.GetFilesInZipFile Method

Returns a list of files in a given ZIP file.

**C#**

```
public DS.LocalizationJobFileDataTable GetFilesInZipFile(Long FileID);
```

**Visual Basic**

```
Public Function GetFilesInZipFile(ByVal FileID As Long) As DS.LocalizationJobFileDataTable
```

**Parameters**

| Parameters | Description |
|---|---|
| FileID | Specifies a file (e.g., ZIP file) |

**Returns**

A LocalizationJobDataSet.LocalizationJobFileDataTable of files.

**Remarks**

Files are typically ZIP and XLIFF files for import jobs. The export process usually will not generate ZIP files with other ZIP files in them.

## GetFlagUrl Method

This is the overview for the GetFlagUrl method overload.

**Description**

Returns a URL to a small image of a national flag for the country and language specified by LanguageID or Data. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | Localization.GetFlagUrl (Integer) (see page 676) | Returns a URL to a small image of a national flag for the country and language specified by LanguageID. |
| | Localization.GetFlagUrl (LanguageData) (see page 677) | 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 677) | 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 ID.

**C#**

```
public DS.LocalizationTargetRow GetInterchangeFile(Long ID);
```

**Visual Basic**

```
Public Function GetInterchangeFile(ByVal ID As Long) 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 item (i.e., XLIFF Skeleton).

**C#**

```
public DS.LocalizationTargetDataTable GetInterchangeFiles(Long ItemID);
```

**Visual Basic**

```
Public Function GetInterchangeFiles(ByVal ItemID As Long) As DS.LocalizationTargetDataTable
```

**Parameters**

| Parameters | Description |
|---|---|
| ItemID | Specifies the content item (i.e., XLIFF skeleton). |

**Returns**

A LocalizationJobDataSet.LocalizationTargetDataTable of XLIFF files.

**Remarks**

When exporting, XLIFF files for each target language are generated from a skeleton. When importing, an XLIFF file is associated with the skeleton content item it was generated from (the original language) and a new skeleton content item generated by merging the XLIFF with the original skeleton to produce a skeleton in the target language. A skeleton consists of the content, typically the main HTML content block, a summary (HTML), title, meta data, etc.

### Localization.GetInterchangeFilesInZipFile Method

Returns a list of interchange files (i.e., XLIFF files) contained within a given ZIP file.

**C#**

```
public DS.LocalizationTargetDataTable GetInterchangeFilesInZipFile(Long FileID);
```

**Visual Basic**

```
Public Function GetInterchangeFilesInZipFile(ByVal FileID As Long) As
DS.LocalizationTargetDataTable
```

**Parameters**

| Parameters | Description |
|---|---|
| FileID | Specifies the ZIP file. |

**Returns**

A LocalizationJobDataSet.LocalizationTargetDataTable of XLIFF files.

**Remarks**

XLIFF is a standard to interchange localization information for translation of text. The XLIFF files are compressed together into one or more ZIP files for convenience.

## GetJobs Method

This is the overview for the GetJobs method overload.

**Description**

Gets a list of top level jobs or a list of the sub jobs of the given job. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | Localization.GetJobs () (see page 679) | 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.<br>This function is overloaded. See Also: GetJobs (Long) |
| | Localization.GetJobs (Long) (see page 679) | Returns a list of the sub jobs of the given job.<br>This function is overloaded. See Also: GetJobs |

## Localization.GetJobs Method ()

Returns a list of the top-level jobs, export and import jobs. The jobs listed depend on the user id, unauthenticated users (e.g., translators) see only publicly available jobs (e.g., ZIP files for download). Administrators can view all jobs, non-administrators may only view jobs they created.

This function is overloaded. See Also: GetJobs (Long)

**C#**

```
public DS.LocalizationJobDataTable GetJobs();
```

**Visual Basic**

```
Public Function GetJobs() As DS.LocalizationJobDataTable
```

**Returns**

A LocalizationJobDataSet.LocalizationJobDataTable of top-level jobs.

## Localization.GetJobs Method (Long)

Returns a list of the sub jobs of the given job.

This function is overloaded. See Also: GetJobs (see page 679)

**C#**

```
public DS.LocalizationJobDataTable GetJobs(Long ParentJobID);
```

**Visual Basic**

```
Public Function GetJobs(ByVal ParentJobID As Long) As DS.LocalizationJobDataTable
```

**Parameters**

| Parameters | Description |
|---|---|
| ParentJobID | Specifies the job. |

**Returns**

A LocalizationJobDataSet.LocalizationJobDataTable data table of jobs.

**Remarks**

### Localization.GetLocalizationUrl Method

Returns a URL to the base directory where localization files are stored. For example, /~SiteRoot~/uploadedfiles/localization.

**C#**

```
public String GetLocalizationUrl();
```

**Visual Basic**

```
Public Function GetLocalizationUrl() As String
```

**Returns**

A URL to the base localization directory.

### Localization.GetTranslationUploadDirectory Method

Returns the server directory path to upload XLIFF and ZIP files for importing. For example, C:\inetpub\wwwroot\~SiteRoot~\workarea\uploadedfiles\localization.

See Also: StartImportTranslation (see page 682).

**C#**

```
public String GetTranslationUploadDirectory();
```

**Visual Basic**

```
Public Function GetTranslationUploadDirectory() As String
```

**Returns**

The server directory path (as a String) to store XLIFF and ZIP files that are to be imported.

**Remarks**

This directory path should not be displayed on a web page.

### Localization.StartExportContentForTranslation Method

Starts the process of exporting the content specified by ListOfContentIDs. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text and documents are exported as XLIFF files compressed in ZIP files. The XLIFF files are suitable for sending to a translator. Once translated, the XLIFF files should be imported using the StartImportTranslation (see page 682) method.

**C#**

```
public StartExportContentForTranslation(String ListOfContentIDs, Optional ByVal String
ListOfLanguageIDs);
```

**Visual Basic**

```
Public Sub StartExportContentForTranslation(ByVal ListOfContentIDs As String, Optional ByVal
ListOfLanguageIDs As String = "")
```

**Parameters**

| Parameters | Description |
|---|---|
| ListOfContentIDs | Comma-delimited list of content IDs. e.g., "11,12" |
| ListOfLanguageIDs | Comma-delimited list of language IDs. e.g., "1031,1036" |

**Remarks**

To export all content in a folder, use StartExportFolderForTranslation (see page 681) (Long Integer Boolean String). See http://www.xliff.org for more information about XLIFF.

## Localization.StartExportFolderForTranslation Method

Starts the process of exporting all the content specified by ContentType in the specified folder. Optionally, all the content in the subfolders may be processed as well. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text and documents are exported as XLIFF files compressed in ZIP files. The XLIFF files are suitable for sending to a translator. Once translated, the XLIFF files should be imported using the StartImportTranslation (see page 682).

**C#**

```
public StartExportFolderForTranslation(Long FolderID, int ContentType, Boolean
IncludeSubfolders, Optional ByVal String ListOfLanguageIDs);
```

**Visual Basic**

```
Public Sub StartExportFolderForTranslation(ByVal FolderID As Long, ByVal ContentType As
Integer, ByVal IncludeSubfolders As Boolean, Optional ByVal ListOfLanguageIDs As String = "")
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | Specifies the folder containing the content to export. |
| ContentType | Specifies which type or types of content 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, use StartExportFolderForTranslation (Long Integer Boolean String)

See http://www.xliff.org for more information about XLIFF.

## Localization.StartExportMenusForTranslation Method

Starts the process of exporting all menus. The status of the process may be monitored by visiting localizationjobs.aspx, which uses methods in this class. The translatable text is exported as an XLIFF file and compressed in a ZIP file. The XLIFF file is suitable for sending to a translator. Once translated, the XLIFF file should be imported using the StartImportTranslation (see page 682) method.

**C#**

```
public StartExportMenusForTranslation(Optional ByVal 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 (see page 680). Files may be XLIFF files (.xlf extension) or XLIFF files that are compressed in ZIP (.zip extension) files.

**C#**

```
public StartImportTranslation();
```

**Visual Basic**

```
Public Sub StartImportTranslation()
```

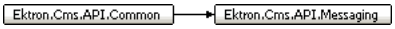
**Remarks**

See http://www.xliff.org for more information about XLIFF.

## Messaging Class

The Messaging class allows you to work with Private Message Objects.

**Class Hierarchy**




**C#**

```
public class Messaging : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Messaging
Inherits Ektron.Cms.API.Common
```

**File**

Messaging.vb

## Messaging Methods

The methods of the Messaging class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | GetUnreadMessageCountForUser (see page 682) | Gets the total unread messages for a given user ID. |

### Messaging.GetUnreadMessageCountForUser Method

Gets the total unread messages for a given user ID.

**C#**

```
public int GetUnreadMessageCountForUser(Long userId);
```

### Visual Basic

```
Public Function GetUnreadMessageCountForUser(ByVal userId As Long) As Integer
```

### Parameters

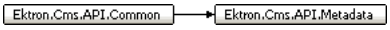
| Parameters | Description |
|---|---|
| userId | ID of the user. |

### Returns

Integer count of unread messages for a user.

# Metadata Class

### Class Hierarchy




### C#

```
public class Metadata : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class Metadata
Inherits Ektron.Cms.API.Common
```

### File

Metadata.vb

### Description

This is class Ektron.Cms.API.Metadata.

## Metadata Methods

The methods of the Metadata class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddMetaDataType (see page 683) | This is the overview for the AddMetaDataType method overload. |
| | DeleteMetadataType (see page 686) | Deletes a metadata type from the system. |
| | GetContentMetadataList (see page 686) | This is the overview for the GetContentMetadataList method overload. |
| | GetIntegratedSearchFolder (see page 688) | Loads all custom folder data. |
| | GetIntegratedSearchFolders (see page 688) | Loads all the custom folders. |
| | GetMetadataByFolder (see page 689) | Returns a collection of ContentMetadata assigned to a folder. |
| | GetMetadataType (see page 689) | Loads all content metadata. |
| | GetMetaDataTypes (see page 689) | Loads all the metadata types. |
| | UpdateMetadataType (see page 690) | Updates the metadata type. |

### AddMetaDataType Method

This is the overview for the AddMetaDataType method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Metadata.AddMetaDataType (Collection) (see page 684) | Adds a new metadata type to the system. |
| | Metadata.AddMetaDataType (ContentMetaData) (see page 684) | Adds a new metadata type to the system. |

## Metadata.AddMetaDataType Method (Collection)

Adds a new metadata type to the system.

**C#**

```
public AddMetaDataType(Collection pagedata);
```

**Visual Basic**

```
Public Sub AddMetaDataType(ByVal pagedata As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | Metadata (see page 683) items as value key pair. |

## Metadata.AddMetaDataType Method (ContentMetaData)

Adds a new metadata type to the system.

**C#**

```
public AddMetaDataType(ref ContentMetaData metaDataItem);
```

**Visual Basic**

```
Public Sub AddMetaDataType(ByRef metaDataItem As ContentMetaData)
```

**Parameters**

| Parameters | Description |
|---|---|
| metaDataItem | ContentMetadata object containing metadata definition details. |

**Example**

The following example shows how to create a Web page from which you can add a new metadata definition. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddMetaDataType(ByRef metaDataItem As ContentMetaData) method. This method uses the InternalAdmin to add the metadata definition. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Create a table with four rows and three columns.
2. In the first row, first column, add a label that shows where to enter the name of the metadata definition.
   ```
   <asp:Label ID="Label1" runat="server" Text="Name: "></asp:Label>
   ```
3. In the first row, second column, add a text box for the user to enter the metadata definition name.
   ```
   <asp:TextBox ID="txtMetaName" runat="server" Width="86px"></asp:TextBox>
   ```
4. In the first row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the name box.
   ```
   <asp:RequiredFieldValidator ID="rfvMetaName" runat="server" ControlToValidate="txtMetaName"
     ErrorMessage="Please enter metadata definition name"
   Width="237px"></asp:RequiredFieldValidator>
   ```
5. In the second row, first column, add a label that show where to select the type.
   ```
   <asp:Label ID="Label4" runat="server" Text="Type:" Width="80px"></asp:Label>
   ```
6. In the second row, second column, add a drop down list for the metadata tag type. We will populate the drop down list in the

code behind.

```
<asp:DropDownList ID="ddlMetaTagType" runat="server" Width="194px"></asp:DropDownList>
```

7. Leave the second row, third column empty.
8. In the thrid row, first column, add a label for the Editable option.

```
<asp:Label ID="Label7" runat="server" Text="Editable: " Width="76px"></asp:Label>
```

9. In the third row, second column, add a check box to decide whether a user can edit the contents of the metadata when creating or editing the metadata's content.

```
<asp:CheckBox ID="chkEditable" runat="server" Checked="True" />
```

10. Leave the third row, third column empty.
11. In the fourth row, first column, add a label for the Display in Ektron Explorer option.

```
<asp:Label ID="Label2" runat="server" Text="Display in Ektron Explorer"
Width="119px"></asp:Label>
```

12. In the fourth row, second column, add a check box to decide whether the metadata definition should be displayed in Ektron Explorer.

```
<asp:CheckBox ID="chkDisplayEE" runat="server" />
```

13. Leave the fourth row, third column empty.
14. After the table, add a break and then add a button to add the Metadata Definition.

```
<br />
<asp:Button ID="btnAdd" runat="server" Text="Add Metatype" Width="98px" />
```

15. Add two breaks for spacing purposes.

```
<br /><br />
```

16. Add a results label to display whether the Metadata Definition was added and if so, its ID.

```
<asp:Label ID="lblResult" runat="server" Width="549px" Height="25px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a page load event.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
```

2. In the page load event, populate the drop down list with the available metadata definition types. These are the only types available. Make sure you add the numeric value. This corresponds to metadata definition type in the database.

```
    If (Not IsPostBack()) Then
        ddlMetaTagType.Items.Add(New ListItem("Searchable Property", "100"))
        ddlMetaTagType.Items.Add(New ListItem("Meta Tag", "1"))
        ddlMetaTagType.Items.Add(New ListItem("HTML Tag", "0"))
        ddlMetaTagType.Items.Add(New ListItem("Collection Selector", "2"))
        ddlMetaTagType.Items.Add(New ListItem("Content Selector", "4"))
        ddlMetaTagType.Items.Add(New ListItem("File Selector", "7"))
        ddlMetaTagType.Items.Add(New ListItem("Hyperlink Selector", "6"))
        ddlMetaTagType.Items.Add(New ListItem("Image Selector", "5"))
        ddlMetaTagType.Items.Add(New ListItem("ListSummary Selector", "3"))
    End If
End Sub
```

3. Add a button click event.

```
Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnAdd.Click
```

4. Create the following objects.

```
        Dim metaDataApi As New Ektron.Cms.API.Metadata
        Dim metaItem As New Ektron.Cms.ContentMetaData
```

5. Assign the following metaItem properties.

```
        metaItem.TypeName = txtMetaName.Text
        metaItem.Editable = chkEditable.Checked
        metaItem.MetaDisplayEE = chkDisplayEE.Checked
        metaItem.TagType = ddlMetaTagType.SelectedItem.Value
```

6. Create a Try/Catch around the AddMetaDataType method. This allows you to catch any exceptions and helps when debugging.

```
        Try
```

```
                metaDataApi.AddMetaDataType(metaItem)
```

7. Check to make sure a valid TypeID was returned. If so, display the a message and the ID of the newly created metadata definition.

```
                If metaItem.TypeId <> 0 Then
                    lblResult.Text = "The above metadata type definition was created in
CMS400.NET with ID = " & metaItem.TypeId
                End If
```

8. If there is an error, let the user know what happened.

```
            Catch ex As Exception
               lblResult.Text = ex.Message
            End Try
        End Sub
```

### Metadata.DeleteMetadataType Method

Deletes a metadata type from the system.

**C#**

```
public DeleteMetadataType(Long Id);
```

**Visual Basic**

```
Public Sub DeleteMetadataType(ByVal Id As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | Metatype ID |

### GetContentMetadataList Method

This is the overview for the GetContentMetadataList method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Metadata.GetContentMetadataList (Long) (see page 686) | Loads all of the metadata for the given content. |
| | Metadata.GetContentMetadataList (Long, Boolean) (see page 688) | Loads all of the metadata for the given content. |

### Metadata.GetContentMetadataList Method (Long)

Loads all of the metadata for the given content.

**C#**

```
public CustomAttributeList GetContentMetadataList(Long ContentId);
```

**Visual Basic**

```
Public Function GetContentMetadataList(ByVal ContentId As Long) As CustomAttributeList
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | The contents ID. |

**Returns**

Ektron.Cms.CustomAttributeList

**Example**

The following example shows how to create a Web page from which all metadata for a specific content ID can be retrieved. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetContentMetadataList(ByVal ContentId as Long) method. This method uses the InternalAdmin to retrieve the metadata for the content ID. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label.
   ```
   <asp:Label ID="Label1" runat="server" Text="Content ID: "></asp:Label>
   ```
2. Add a text box to enter the content's ID.
   ```
   <asp:TextBox ID="txtContentId" runat="server" Width="63px"></asp:TextBox>
   ```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Content ID box.
   ```
   <asp:RequiredFieldValidator ID="rfvContentId" runat="server"
ControlToValidate="txtContentId"
                ErrorMessage="Please enter content ID"></asp:RequiredFieldValidator>
   ```
4. Add two breaks for spacing purposes.
   ```
   <br /><br />
   ```
5. Add a button to get the metadata. We will set the button click event in the code behind.
   ```
   <asp:Button ID="btnGetMetaData" runat="server" Text="Get MetaData" Width="91px" />
   ```
6. Add two more breaks for spacing purposes.
   ```
   <br /><br />
   ```
7. Add a label to display to display the metadata information.
   ```
   <asp:Label ID="lblResult" runat="server" Width="604px" Height="113px"></asp:Label>
   ```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

   ```
   Protected Sub btnGetMetaData_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetMetaData.Click
   ```

2. Create the following objects for the Metadata (see page 683) API (see page 639), Custom Attribute List and each Custom Attribute item.

   ```
   Dim metaDataApi As New Ektron.Cms.API.Metadata
   Dim metaDataList As New Ektron.Cms.CustomAttributeList
   Dim metaItem As New Ektron.Cms.CustomAttribute
   ```

3. Check to make sure the content ID is numeric.

   ```
   If Not IsNumeric(txtContentId.Text) Then
       lblResult.Text = "Invalid Content ID"
       Exit Sub
   End If
   ```

4. Create a Try/Catch around the GetContentMetadataList method. This allows you to catch any exceptions and helps when debugging.

   ```
   Try
       metaDataList = metaDataApi.GetContentMetadataList(CInt(txtContentId.Text))
   ```

5. Verify that items are returned and create a table to display the returned information.

   ```
   If metaDataList.AttributeList.Length > 0 Then
       lblResult.Text = "<table> <th>Metaname</th> <th align=""left"">Value</th> "
   ```

6. Populate the table with metadata information and display it.

   ```
   For Each metaItem In metaDataList.AttributeList
       If metaItem.Value.ToString().Length > 0 Then
           lblResult.Text &= "<tr>"
           lblResult.Text &= "<td>" & metaItem.Name & "</td>"
           lblResult.Text &= "<td>" & metaItem.Value & "</td>"
           lblResult.Text &= "</tr>"
       End If
   Next
   lblResult.Text &= "</table>"
   ```

8. If no metadata exists, display that information.

```
            Else
                lblResult.Text = "No metadata definitions found"
                Exit Sub
            End If
```

9. Display a message if there is an error.

```
        Catch ex As Exception
            lblResult.Text = ex.Message
        End Try
    End Sub
```

### Metadata.GetContentMetadataList Method (Long, Boolean)

Loads all of the metadata for the given content.

**C#**

```
public CustomAttributeList GetContentMetadataList(Long ContentId, Boolean preview);
```

**Visual Basic**

```
Public Function GetContentMetadataList(ByVal ContentId As Long, ByVal preview As Boolean) As
CustomAttributeList
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | The contents ID. |
| preview | True implies get metadata from staging content, else use published content. |

**Returns**

Ektron.Cms.CustomAttributeList

### Metadata.GetIntegratedSearchFolder Method

Loads all custom folder data.

**C#**

```
public IntegratedFolderData GetIntegratedSearchFolder(Long Id);
```

**Visual Basic**

```
Public Function GetIntegratedSearchFolder(ByVal Id As Long) As IntegratedFolderData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The integrated search folder ID |

**Returns**

Ektron.Cms.IntegratedSearchData

### Metadata.GetIntegratedSearchFolders Method

Loads all the custom folders.

**C#**

```
public IntegratedFolderData[] GetIntegratedSearchFolders();
```

**Visual Basic**

```
Public Function GetIntegratedSearchFolders() As IntegratedFolderData()
```

**Returns**

Ektron.Cms.IntegratedSearchData()

### Metadata.GetMetadataByFolder Method

Returns a collection of ContentMetadata assigned to a folder.

**C#**

```
public ContentMetaData[] GetMetadataByFolder(Long folderId);
```

**Visual Basic**

```
Public Function GetMetadataByFolder(ByVal folderId As Long) As ContentMetaData()
```

**Parameters**

| Parameters | Description |
|---|---|
| folderId | folder to retrieve metadata for |

**Returns**

array of Ektron.Cms.ContentMetadata objects

### Metadata.GetMetadataType Method

Loads all content metadata.

**C#**

```
public ContentMetaData GetMetadataType(Long Id);
```

**Visual Basic**

```
Public Function GetMetadataType(ByVal Id As Long) As ContentMetaData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The metadata type ID |

**Returns**

Ektron.Cms.ContentMetaData

### Metadata.GetMetaDataTypes Method

Loads all the metadata types.

**C#**

```
public ContentMetaData[] GetMetaDataTypes(String OrderBy);
```

**Visual Basic**

```
Public Function GetMetaDataTypes(ByVal OrderBy As String) As ContentMetaData()
```

**Parameters**

| Parameters | Description |
|---|---|
| OrderBy | Order by clause |

**Returns**

Ektron.Cms.ContentMetaData()

### Metadata.UpdateMetadataType Method

Updates the metadata type.

**C#**

```
public UpdateMetadataType(Collection pagedata);
```

**Visual Basic**

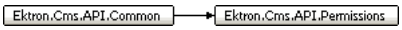
```
Public Sub UpdateMetadataType(ByVal pagedata As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | Metadata (see page 683) types |

# Permissions Class

**Class Hierarchy**




**C#**

```
public class Permissions : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Permissions
Inherits Ektron.Cms.API.Common
```

**File**

Permissions.vb

**Description**

This is class Ektron.Cms.API.Permissions.

## Permissions Methods

The methods of the Permissions class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddItemPermission (see page 691) | Adds the item permissions for a user or a usergroup. |
| | DeleteItemPermission (see page 691) | Deletes a user or a group from the item's permissions |
| | DisableContentInheritance (see page 691) | Disable permissions inheritance on the content and copies current permissions to the content. |
| | DisableFolderInheritance (see page 692) | Disable permissions inheritance on the folder and copies current permissions to the folder. |
| | EnableContentInheritance (see page 692) | Deletes current permissions from the content and enable permissions inheritance on the content. |
| | EnableFolderInheritance (see page 692) | Deletes current permissions from the folder and enable permissions inheritance on the folder. |
| | GetUserPermissions (see page 692) | This is the overview for the GetUserPermissions method overload. |
| | LoadPermissions (see page 694) | Loads the permission for given object, such as content, folder, task, etc. |
| | UpdateItemPermission (see page 694) | Updates the item permissions for a user or a usergroup. |

### Permissions.AddItemPermission Method

Adds the item permissions for a user or a usergroup.

**C#**

```
public Boolean AddItemPermission(UserPermissionData userPermData);
```

**Visual Basic**

```
Public Function AddItemPermission(ByVal userPermData As UserPermissionData) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| userPermData | User (see page 913)'s Permissions (see page 690) Data |

**Returns**

False if it's successful

### Permissions.DeleteItemPermission Method

Deletes a user or a group from the item's permissions

**C#**

```
public Boolean DeleteItemPermission(Long id, Boolean IsGroupId, Long itemId,
Ektron.Cms.Common.EkEnumeration.CMSObjectTypes itemType);
```

**Visual Basic**

```
Public Function DeleteItemPermission(ByVal id As Long, ByVal IsGroupId As Boolean, ByVal
itemId As Long, ByVal itemType As Ektron.Cms.Common.EkEnumeration.CMSObjectTypes) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| id | User (see page 913) 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 788) or EkEnumeration.CMSObjectTypes.Folder (see page 652)? |

**Returns**

False if it's successful.

### Permissions.DisableContentInheritance Method

Disable permissions inheritance on the content and copies current permissions to the content.

**C#**

```
public Boolean DisableContentInheritance(Long id);
```

**Visual Basic**

```
Public Function DisableContentInheritance(ByVal id As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Content (see page 788) Id |

**Returns**

False if it's successful.

### Permissions.DisableFolderInheritance Method

Disable permissions inheritance on the folder and copies current permissions to the folder.

**C#**

```
public Boolean DisableFolderInheritance(Long id);
```

**Visual Basic**

```
Public Function DisableFolderInheritance(ByVal id As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| id | folder Id |

**Returns**

False if it's successful.

### Permissions.EnableContentInheritance Method

Deletes current permissions from the content and enable permissions inheritance on the content.

**C#**

```
public Boolean EnableContentInheritance(Long id);
```

**Visual Basic**

```
Public Function EnableContentInheritance(ByVal id As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Content (see page 788) Id |

**Returns**

False if it's successful.

### Permissions.EnableFolderInheritance Method

Deletes current permissions from the folder and enable permissions inheritance on the folder.

**C#**

```
public Boolean EnableFolderInheritance(Long id);
```

**Visual Basic**

```
Public Function EnableFolderInheritance(ByVal id As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| id | folder Id |

**Returns**

False if it's successful.

### GetUserPermissions Method

This is the overview for the GetUserPermissions method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Permissions.GetUserPermissions (Long, String, Long, String) (see page 693) | Loads the user permissions as a array of UserPermissionData. |
| | Permissions.GetUserPermissions (Long, String, Long, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) (see page 693) | Loads the user permissions as a array of UserPermissionData. |

## Permissions.GetUserPermissions Method (Long, String, Long, String)

Loads the user permissions as a array of UserPermissionData.

**C#**

```
public UserPermissionData[] GetUserPermissions(Long Id, String ItemType, Long UserId, String
UserGroupList);
```

**Visual Basic**

```
Public Function GetUserPermissions(ByVal Id As Long, ByVal ItemType As String, ByVal UserId As
Long, ByVal UserGroupList As String) As UserPermissionData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (Long, String, Long, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType)

Loads the user permissions as a array of UserPermissionData.

**C#**

```
public UserPermissionData[] GetUserPermissions(Long Id, String ItemType, Long UserId, String
UserGroupList, ContentAPI.PermissionUserType PermissionType, ContentAPI.PermissionRequestType
PermissionRequest);
```

**Visual Basic**

```
Public Function GetUserPermissions(ByVal Id As Long, ByVal ItemType As String, ByVal UserId As
Long, ByVal UserGroupList As String, ByVal PermissionType As ContentAPI.PermissionUserType,
ByVal PermissionRequest As ContentAPI.PermissionRequestType) As UserPermissionData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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, folder, task, etc.

**C#**

```
public PermissionData LoadPermissions(Long Id, String RequestType, Optional ByVal
ContentAPI.PermissionResultType Type);
```

**Visual Basic**

```
Public Function LoadPermissions(ByVal Id As Long, ByVal RequestType As String, Optional ByVal
Type As ContentAPI.PermissionResultType = contentAPI.PermissionResultType.Common) As
PermissionData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The object ID |
| RequestType | The object type := content \| folder \| collections \| tasks |
| Type | The permission result type |

**Returns**

Ektron.Cms.PermissionData

**Remarks**

User (see page 913) login with administrator privileges are required to perform this operation.

### Permissions.UpdateItemPermission Method

Updates the item permissions for a user or a usergroup.

**C#**

```
public Boolean UpdateItemPermission(UserPermissionData userPermData);
```

**Visual Basic**

```
Public Function UpdateItemPermission(ByVal userPermData As UserPermissionData) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| userPermData | User (see page 913)'s Permissions (see page 690) Data |

**Returns**

False if it's successful

## Site Class

**Class Hierarchy**

**C#**

```
public class Site : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Site
Inherits Ektron.Cms.API.Common
```

**File**

Site.vb

**Description**

This is class Ektron.Cms.API.Site.

## Site Methods

The methods of the Site class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | BreakTaskPermissionInherit (see page 695) | Removes the task permissions for a given user. |
| | DeleteCookie (see page 696) | Removes the cookie from the site. |
| | GetAllActiveLanguages (see page 696) | Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language or Settings area. |
| | GetAllLanguages (see page 696) | This is the overview for the GetAllLanguages method overload. |
| | GetClientDayNames (see page 697) | Returns the day name of a given content language (or site language). |
| | GetClientMonthNames (see page 697) | Returns the month name of a given content language (or site language) |
| | GetCookieValue (see page 697) | Reads the cookie value. |
| | GetLanguage (see page 698) | Returns the details about the given language ID. |
| | GetLanguages (see page 698) | Gets a list of all languages that meet a given state, for example, active. |
| | GetSiteVariables (see page 698) | Returns all of the settings data. |
| | GetTaskPermission (see page 699) | Returns 1 if the user has one of the task permissions, otherwise 0. |
| | IsTaskPermissionInherited (see page 699) | Returns true if a user has already granted one of the task permissions. |
| | SetCookieValue (see page 699) | Resets the cookie value. |
| | SetTaskPermission (see page 700) | Updates the task permission(s) for the given user or group. |
| | UpdateSiteEnabledLanguages (see page 700) | Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled. |
| | UpdateSiteVariables (see page 700) | Update the site settings |

### Site.BreakTaskPermissionInherit Method

Removes the task permissions for a given user.

**C#**

```
public BreakTaskPermissionInherit(Long Id);
```

**Visual Basic**

```
Public Sub BreakTaskPermissionInherit(ByVal Id As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | User (see page 913) ID |

### Site.DeleteCookie Method

Removes the cookie from the site.

**C#**

```
public DeleteCookie();
```

**Visual Basic**

```
Public Sub DeleteCookie()
```

### Site.GetAllActiveLanguages Method

Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language or Settings area.

**C#**

```
public LanguageData[] GetAllActiveLanguages();
```

**Visual Basic**

```
Public Function GetAllActiveLanguages() As LanguageData()
```

**Returns**

Array of Ektron.Cms.LanguageData

### GetAllLanguages Method

This is the overview for the GetAllLanguages method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Site.GetAllLanguages () (see page 696) | Gets a list of all the available languages sorted by language name in ascending order. |
| | Site.GetAllLanguages (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) (see page 697) | Gets a list of all the available languages. |

### Site.GetAllLanguages Method ()

Gets a list of all the available languages sorted by language name in ascending order.

**C#**

```
public LanguageData[] GetAllLanguages();
```

**Visual Basic**

```
Public Function GetAllLanguages() As LanguageData()
```

**Returns**

Array of Ektron.Cms.LanguageData

### Site.GetAllLanguages Method (LanguageDataSet.SortBy, LanguageDataSet.SortDirection)

Gets a list of all the available languages.

**C#**

```
public LanguageData[] GetAllLanguages(LanguageDataSet.SortBy orderBy,
LanguageDataSet.SortDirection direction);
```

**Visual Basic**

```
Public Function GetAllLanguages(ByVal orderBy As LanguageDataSet.SortBy, ByVal direction As
LanguageDataSet.SortDirection) As LanguageData()
```

**Parameters**

| Parameters | Description |
|---|---|
| orderBy | Determines the sort order. |
| direction | Determines the sort direction. |

**Returns**

Array of Ektron.Cms.LanguageData

### Site.GetClientDayNames Method

Returns the day name of a given content language (or site language).

**C#**

```
public String GetClientDayNames();
```

**Visual Basic**

```
Public Function GetClientDayNames() As String
```

**Returns**

String

### Site.GetClientMonthNames Method

Returns the month name of a given content language (or site language)

**C#**

```
public String GetClientMonthNames();
```

**Visual Basic**

```
Public Function GetClientMonthNames() As String
```

**Returns**

String

### Site.GetCookieValue Method

Reads the cookie value.

**C#**

```
public String GetCookieValue(String Keyname);
```

**Visual Basic**

```
Public Function GetCookieValue(ByVal Keyname As String) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| Keyname | The Keyname. (Required) |

**Returns**

String

### Site.GetLanguage Method

Returns the details about the given language ID.

**C#**

```
public LanguageData GetLanguage(Long Id);
```

**Visual Basic**

```
Public Function GetLanguage(ByVal Id As Long) As LanguageData
```

**Parameters**

| Parameters | Description |
|---|---|
| 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,
Optional ByVal LanguageDataSet.SortBy SortBy, Optional ByVal LanguageDataSet.SortDirection
SortDirection);
```

**Visual Basic**

```
Public Function GetLanguages(ByVal State As LanguageDataSet.LanguageState, Optional ByVal
SortBy As LanguageDataSet.SortBy = LanguageDataSet.SortBy.LanguageState, Optional ByVal
SortDirection As LanguageDataSet.SortDirection = LanguageDataSet.SortDirection.Ascending) As
LanguageDataSet.LanguageDataTable
```

**Parameters**

| Parameters | Description |
|---|---|
| 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(Optional ByVal Long Id);
```

**Visual Basic**

```
Public Function GetSiteVariables(Optional ByVal Id As Long = -1) As SettingsData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | User (see page 913) ID |

**Returns**

Ektron.Cms.SettingsData

**Remarks**

If user ID is passed it returns the additional value of MaxUsersAllowed, AdInstalled and AdAllowed properties.

### Site.GetTaskPermission Method

Returns 1 if the user has one of the task permissions, otherwise 0.

**C#**

```
public Long GetTaskPermission(Long Id, int Type);
```

**Visual Basic**

```
Public Function GetTaskPermission(ByVal Id As Long, ByVal Type As Integer) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | User (see page 913) or group ID |
| Type | User (see page 913) or group (1 = user, 2 = group). |

**Returns**

Returns 1 if the user has one of the task permissions. Otherwise, it returns 0.

### Site.IsTaskPermissionInherited Method

Returns true if a user has already granted one of the task permissions.

**C#**

```
public Boolean IsTaskPermissionInherited(Long Id);
```

**Visual Basic**

```
Public Function IsTaskPermissionInherited(ByVal Id As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | User (see page 913) ID |

**Returns**

Boolean

### Site.SetCookieValue Method

Resets the cookie value.

**C#**

```
public SetCookieValue(String Keyname, String CookieValue);
```

**Visual Basic**

```
Public Sub SetCookieValue(ByVal Keyname As String, ByVal CookieValue As String)
```

**Parameters**

| Parameters | Description |
|---|---|
| Keyname | The Keyname. (Required) |
| CookieValue | Value |

## Site.SetTaskPermission Method

Updates the task permission(s) for the given user or group.

**C#**

```
public SetTaskPermission(Long Id, int UserType, int PermissionType);
```

**Visual Basic**

```
Public Sub SetTaskPermission(ByVal Id As Long, ByVal UserType As Integer, ByVal PermissionType
As Integer)
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | User (see page 913) ID or Group ID |
| UserType | User (see page 913) (1) or Group (2) |
| PermissionType | Ektron.Cms.Common.TaskPermission (1 = Create Task (see page 871), 2 = Redirect Task (see page 871), 3 = Delete Task (see page 871), 4 = Edit Alias) |

## Site.UpdateSiteEnabledLanguages Method

Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.

**C#**

```
public UpdateSiteEnabledLanguages(String LanguageIDs);
```

**Visual Basic**

```
Public Sub UpdateSiteEnabledLanguages(ByVal LanguageIDs As String)
```

**Parameters**

| Parameters | Description |
|---|---|
| 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**

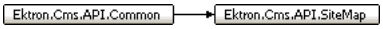
```
Public Sub UpdateSiteVariables(ByVal item As Hashtable)
```

**Parameters**

| Parameters | Description |
|---|---|
| item | Set of settings data as key, value pair. |

# SiteMap Class

### Class Hierarchy




### C#

```
public class SiteMap : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class SiteMap
Inherits Ektron.Cms.API.Common
```

### File

SiteMap.vb

### Description

This is class Ektron.Cms.API.SiteMap.

## SiteMap Methods

The methods of the SiteMap class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | GetFolderBreadcrumbPath (see page 701) | Get breadcrumb path of the folder in which content block exists. |
| | GetWebsiteMap (see page 703) | Creates the site map from folder breadcrumb. |
| | RenderFolderBreadcrumbHtml (see page 704) | Render folder breadcrumb HTML. |
| | RenderWebsiteMap (see page 705) | Create HTML to render on the page. |

### SiteMap.GetFolderBreadcrumbPath Method

Get breadcrumb path of the folder in which content block exists.

### C#

```
public Ektron.Cms.Common.SitemapPath[] GetFolderBreadcrumbPath(Long id, Optional ByVal Boolean
isFolder);
```

### Visual Basic

```
Public Function GetFolderBreadcrumbPath(ByVal id As Long, Optional ByVal isFolder As Boolean =
False) As Ektron.Cms.Common.SitemapPath()
```

### Parameters

| Parameters | Description |
|---|---|
| id | Content (see page 788) or Folder (see page 652) ID. |
| isFolder | Set to True if the ID is a folder ID. |

### Returns

Array of nodes in the path.

### Example

The following example shows how to create a Web page from which the breadcrumb path for a given folder can be retrieved. In

this This example uses some standard drag and drop controls and a small section of VB code utilizing the GetFolderBreadcrumbPath method. The GetFolderBreadcrumbPath method allows you to use a folder ID or a content's ID to retrieve the information. This method uses the InternalAdmin to retrieve the breadcrumb path. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a radio button list. We will populate the radio buttons in the code behind.
   ```
   <asp:RadioButtonList ID="rblItem" runat="server" Height="64px"></asp:RadioButtonList>
   ```
2. Add two breaks for spacing purposes.
   ```
   <br /><br />
   ```
3. Add a label to show where the ID goes.
   ```
   <asp:Label ID="Label6" runat="server" Text="ID:" Width="76px"></asp:Label>
   ```
4. Add a text box for the ID.
   ```
   <asp:TextBox ID="txtId" runat="server" Width="55px"></asp:TextBox>
   ```
5. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the ID box.
   ```
   <asp:RequiredFieldValidator ID="rfvId" runat="server" ControlToValidate="txtId"
      ErrorMessage="Please enter a content/folder ID"
Width="179px"></asp:RequiredFieldValidator>
   ```
6. Add two more breaks for spacing purposes.
   ```
   <br /><br />
   ```
7. Add a button to submit the ID to be queried. We will set the button click event in the code behind.
   ```
   <asp:Button ID="btnGet" runat="server" Text="Get Breadcrumb" Width="105px" />
   ```
8. Add two last breaks for spacing purposes.
   ```
   <br /><br />
   ```
9. Add a label to display the breadcrumb path.
   ```
   <asp:Label ID="lblResult" runat="server" Width="502px" Height="100px"></asp:Label>
   ```

Add the following information to the code behind of your aspx.vb page.

1. In the page load event, populate the radio buttons.

```
        Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
          If Not IsPostBack() Then
              rblItem.Items.Add(New ListItem("Folder", "true"))
              rblItem.Items.Add(New ListItem("Content", "false"))
              rblItem.Items(0).Selected = True
          End If
        End Sub
```

2. Add a button click event.

```
        Protected Sub btnGet_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnGet.Click
```

3. Check to make sure the ID is numeric.

```
            If Not IsNumeric(txtId.Text) Then
                lblResult.Text = "Invalid content/folder ID"
                Exit Sub
            End If
```

4. Create an object for the Site (see page 694) Map (see page 340) API (see page 639).

```
            Dim sitemapApi As New Ektron.Cms.API.SiteMap
```

5. Create an object for the site map data.

```
            Dim mapData As Ektron.Cms.Common.SitemapPath()
```

6. Create a Try/Catch around the GetFolderBreadcrumbPath method. This allows you to catch any exceptions and helps when debugging.

```
            Try
                mapData = sitemapApi.GetFolderBreadcrumbPath(CInt(txtId.Text),
CBool(rblItem.SelectedValue))
```

7. Check to make sure map data is returned. If map data is returned, display the information.

```
                If (Not mapData Is Nothing) AndAlso (mapData.Length > 0) Then
```

```
                    lblResult.Text = sitemapApi.RenderFolderBreadcrumbHtml(mapData, ">", True,
"", False, "")
```

8. If no map data exists, display a message.

```
                Else
                    lblResult.Text = "No breadcrumb path found"
                End If
```

9. Display a message if there is an error.

```
            Catch ex As Exception
               lblResult.Text = ex.Message
            End Try
        End Sub
```

### SiteMap.GetWebsiteMap Method

Creates the site map from folder breadcrumb.

**C#**

```
public WebsiteMap GetWebsiteMap(Optional ByVal Long folderId, Optional ByVal Boolean
getSameLevelNodes);
```

**Visual Basic**

```
Public Function GetWebsiteMap(Optional ByVal folderId As Long = 0, Optional ByVal
getSameLevelNodes As Boolean = False) As WebsiteMap
```

**Parameters**

| Parameters | Description |
|---|---|
| folderId | Build the web sitemap from the folder specified. |
| getSameLevelNodes | Do you want to retrieve all nodes at the same level as the one you specify?<br>**True** = include all sites at the same level as the `folderId` property.<br>**False** = Do not include the same level nodes. |

**Example**

The following example shows how to create a Web page that displays the site map for a given folder. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetWebsiteMap method. This method uses the InternalAdmin to access and return the site map information. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
   `<asp:Label ID="Label6" runat="server" Text="Folder ID:" Width="66px"></asp:Label>`
2. Add text box where the folder ID can be entered.
   `<asp:TextBox ID="txtFolderId" runat="server" Width="55px"></asp:TextBox>`
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.

```
<asp:RequiredFieldValidator ID="rfvId" runat="server" ControlToValidate="txtFolderId"
    ErrorMessage="Please enter folder ID" Width="179px"></asp:RequiredFieldValidator>
```

4. Add two breaks for spacing purposes.
   `<br /><br />`
5. Add a button to submit the folder ID and retrieve the site map information. We will set the button click event in the code behind.
   `<asp:Button ID="btnGet" runat="server" Text="Get site map" Width="91px" />`
6. Add two more breaks for spacing purposes.
   `<br /><br />`
7. Add a label to display the site map.
   `<asp:Label ID="lblResult" runat="server" Width="502px" Height="100px"></asp:Label>`

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
    Protected Sub btnGet_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
```

```
btnGet.Click
```

2. Check to make sure the folder ID is numeric.

```
            If Not IsNumeric(txtFolderId.Text) Then
                lblResult.Text = "Invalid Folder ID"
                Exit Sub
            End If
```

3. Create an object for the site map API (see page 639).

```
            Dim sitemapApi As New Ektron.Cms.API.SiteMap
```

4. Create an object for the map data.

```
            Dim mapData As Ektron.Cms.Common.WebsiteMap
```

5. Create a Try/Catch around the GetWebsiteMap method. This allows you to catch any exceptions and helps when debugging.

```
            Try
                mapData = sitemapApi.GetWebsiteMap(CInt(txtFolderId.Text), False)
```

6. Check to make sure child nodes are returned. If child nodes are returned, display the information.

```
               If (Not mapData.childrenNodes Is Nothing) Then
                   lblResult.Text = sitemapApi.RenderWebsiteMap(mapData, "")
```

7. If no child nodes are returned, display a message.

```
               Else
                   lblResult.Text = "No breadcrumb path found"
               End If
```

8. Display a message if there is an error.

```
            Catch ex As Exception
               lblResult.Text = ex.Message
            End Try
        End Sub
```

### SiteMap.RenderFolderBreadcrumbHtml Method

Render folder breadcrumb HTML.

**C#**

```
public String RenderFolderBreadcrumbHtml(Ektron.Cms.Common.SitemapPath nodes, String
pathSeparator, Boolean linkNodes, String linkTarget, Boolean displayVertical, String pageUrl);
```

**Visual Basic**

```
Public Function RenderFolderBreadcrumbHtml(ByVal nodes() As Ektron.Cms.Common.SitemapPath,
ByVal pathSeparator As String, ByVal linkNodes As Boolean, ByVal linkTarget As String, ByVal
displayVertical As Boolean, ByVal pageUrl As String) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| nodes | The items contained in the breadcrumb path. Ektron.Cms.Common.SitemapPath |
| pathSeparator | a special character that separates the items in a breadcrumb trail. |
| linkNodes | Should the items in the breadcrumb path be Hyperlinked?<br>**True** = hyperlink breadcrumb items.<br>**False** = breadcrumb items appear as plain text. |
| linkTarget | Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.<br>• **_Self** - opens link in the same window.<br>• **_Top** - opens link in a parent window.<br>• **_Blank** - opens link in a new window.<br>• **_Parent** - opens link in the parent frame. |

| | |
|---|---|
| displayVertical | Do you want the breadcrumb path to display vertically?<br>**True** = display the breadcrumb trail vertically.<br>**False** = display the breadcrumb trail horizontally. |
| pageUrl | The current page URL. Usually Request.Url. |

**Returns**

HTML

### SiteMap.RenderWebsiteMap Method

Create HTML to render on the page.

**C#**

```
public String RenderWebsiteMap(WebsiteMap webSitemap, String displayType, Optional ByVal
String className, Optional ByVal int maxLevel, Optional ByVal int startLevel, Optional ByVal
String PageUrl);
```

**Visual Basic**

```
Public Function RenderWebsiteMap(ByVal webSitemap As WebsiteMap, ByVal displayType As String,
Optional ByVal className As String = "ektSitemap", Optional ByVal maxLevel As Integer = 0,
Optional ByVal startLevel As Integer = 0, Optional ByVal PageUrl As String = "") As String
```

**Parameters**

| Parameters | Description |
|---|---|
| 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**

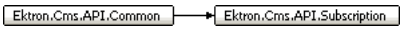
HTML/XML

**Remarks**

Supported displayType: ListWithDescription | List | RawXml

## Subscription Class

**Class Hierarchy**




**C#**

```
public class Subscription : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Subscription
Inherits Ektron.Cms.API.Common
```

**File**

Subscription.vb

**Description**

This is class Ektron.Cms.API.Subscription.

## Subscription Methods

The methods of the Subscription class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddEmailFrom (see page 706) | Adds a subscription email address. |
| | AddSubscription (see page 707) | Adds a subscription. |
| | AddSubscriptionMessage (see page 707) | Adds subscription message(s). |
| | DeleteEmailFrom (see page 707) | Deletes a subscription email address from system. |
| | DeleteSubscription (see page 707) | Deletes a subscription from the system. |
| | DeleteSubscriptionMessage (see page 708) | Deletes the subscription message from the system. |
| | GetAllActiveSubscriptions (see page 708) | Loads all of the active subscriptions. |
| | GetAllEmailFrom (see page 708) | Returns all of the email as an array of EmailFromData. |
| | GetAllSubscriptionMessages (see page 708) | Loads all the subscription messages. |
| | GetAllSubscriptions (see page 709) | Loads all the subscriptions. |
| | GetFolderInheritedFrom (see page 709) | Retrieves the inherited parent ID. |
| | GetSubscription (see page 709) | Retrieves a subscription by using its ID. |
| | GetSubscriptionMessage (see page 709) | Retrieves a subscription message by using the subscription ID. |
| | GetSubscriptionMessagesForType (see page 710) | Retrieve subscription message for type. |
| | GetSubscriptionPropertiesForContent (see page 710) | Retrieves the subscription properties for content using the content's ID. |
| | GetSubscriptionPropertiesForFolder (see page 710) | Retrieves the subscription properties for a given folder. |
| | GetSubscriptionsForContent (see page 711) | Loads the subscriptions for the given content. |
| | GetSubscriptionsForFolder (see page 711) | Loads the subscriptions for a given folder. |
| | SendWebAlert (see page 711) | Triggers the web alert subsystem. Requires that the subscription properties be set for the content first. |
| | UpdateEmailFrom (see page 712) | Update the email address. |
| | UpdateSubscription (see page 712) | Updates the subscription. |
| | UpdateSubscriptionMessage (see page 712) | Updates the subscription message. |
| | UpdateSubscriptionPropertiesForContent (see page 712) | Sets the subscription properties for a single piece of content. |
| | UpdateSubscriptionPropertiesForFolder (see page 713) | Sets the subscription properties for a single folder. |

### Subscription.AddEmailFrom Method

Adds a subscription email address.

**C#**

```
public AddEmailFrom(Collection data);
```

**Visual Basic**

```
Public Sub AddEmailFrom(ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | Email address as a collection. |

## Subscription.AddSubscription Method

Adds a subscription.

**C#**

```
public AddSubscription(Collection data);
```

**Visual Basic**

```
Public Sub AddSubscription(ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | The subscription's name. |

## Subscription.AddSubscriptionMessage Method

Adds subscription message(s).

**C#**

```
public AddSubscriptionMessage(Collection data);
```

**Visual Basic**

```
Public Sub AddSubscriptionMessage(ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | Subscription (see page 705) message data. |

## Subscription.DeleteEmailFrom Method

Deletes a subscription email address from system.

**C#**

```
public DeleteEmailFrom(Collection data);
```

**Visual Basic**

```
Public Sub DeleteEmailFrom(ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | Email data. |

## Subscription.DeleteSubscription Method

Deletes a subscription from the system.

**C#**

```
public DeleteSubscription(Collection data);
```

**Visual Basic**

```
Public Sub DeleteSubscription(ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | Subscription (see page 705) data |

### Subscription.DeleteSubscriptionMessage Method

Deletes the subscription message from the system.

**C#**

```
public DeleteSubscriptionMessage(Collection data);
```

**Visual Basic**

```
Public Sub DeleteSubscriptionMessage(ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | Subscription (see page 705) message data |

### Subscription.GetAllActiveSubscriptions Method

Loads all of the active subscriptions.

**C#**

```
public SubscriptionData[] GetAllActiveSubscriptions();
```

**Visual Basic**

```
Public Function GetAllActiveSubscriptions() As SubscriptionData()
```

**Returns**

Ektron.Cms.SubscriptionData()

### Subscription.GetAllEmailFrom Method

Returns all of the email as an array of EmailFromData.

**C#**

```
public EmailFromData[] GetAllEmailFrom();
```

**Visual Basic**

```
Public Function GetAllEmailFrom() As EmailFromData()
```

**Returns**

Ektron.Cms.EmailFromData()

### Subscription.GetAllSubscriptionMessages Method

Loads all the subscription messages.

**C#**

```
public EmailMessageData[] GetAllSubscriptionMessages();
```

**Visual Basic**

```
Public Function GetAllSubscriptionMessages() As EmailMessageData()
```

**Returns**

Ektron.Cms.EmailMessageData()

### Subscription.GetAllSubscriptions Method

Loads all the subscriptions.

**C#**

```
public SubscriptionData[] GetAllSubscriptions();
```

**Visual Basic**

```
Public Function GetAllSubscriptions() As SubscriptionData()
```

**Returns**

Ektron.Cms.SubscriptionData()

### Subscription.GetFolderInheritedFrom Method

Retrieves the inherited parent ID.

**C#**

```
public Long GetFolderInheritedFrom(Long FolderId);
```

**Visual Basic**

```
Public Function GetFolderInheritedFrom(ByVal FolderId As Long) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderId | The current folder's numeric ID. |

**Returns**

The numeric ID of parent folder from which the folder is inherited.

### Subscription.GetSubscription Method

Retrieves a subscription by using its ID.

**C#**

```
public SubscriptionData GetSubscription(Long Id);
```

**Visual Basic**

```
Public Function GetSubscription(ByVal Id As Long) As SubscriptionData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The subscription's ID. |

**Returns**

Ektron.Cms.SubscriptionData

### Subscription.GetSubscriptionMessage Method

Retrieves a subscription message by using the subscription ID.

**C#**

```
public EmailMessageData GetSubscriptionMessage(Long Id);
```

**Visual Basic**

```
Public Function GetSubscriptionMessage(ByVal Id As Long) As EmailMessageData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The subscription's ID. |

**Returns**

Ektron.Cms.EmailMessageData

### Subscription.GetSubscriptionMessagesForType Method

Retrieve subscription message for type.

**C#**

```
public EmailMessageData[]
GetSubscriptionMessagesForType(Ektron.Cms.Common.EkEnumeration.EmailMessageTypes MessageType);
```

**Visual Basic**

```
Public Function GetSubscriptionMessagesForType(ByVal MessageType As
Ektron.Cms.Common.EkEnumeration.EmailMessageTypes) As EmailMessageData()
```

**Parameters**

| Parameters | Description |
|---|---|
| MessageType | The message type. |

**Returns**

Ektron.Cms.EmailMessageData()

### Subscription.GetSubscriptionPropertiesForContent Method

Retrieves the subscription properties for content using the content's ID.

**C#**

```
public SubscriptionPropertiesData GetSubscriptionPropertiesForContent(Long ContentId);
```

**Visual Basic**

```
Public Function GetSubscriptionPropertiesForContent(ByVal ContentId As Long) As
SubscriptionPropertiesData
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | The content's ID |

**Returns**

Ektron.Cms.SubscriptionPropertiesData

### Subscription.GetSubscriptionPropertiesForFolder Method

Retrieves the subscription properties for a given folder.

**C#**

```
public SubscriptionPropertiesData GetSubscriptionPropertiesForFolder(Long FolderId);
```

**Visual Basic**

```
Public Function GetSubscriptionPropertiesForFolder(ByVal FolderId As Long) As
SubscriptionPropertiesData
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderId | The folder's ID. |

**Returns**

Ektron.Cms.SubscriptionPropertiesData

### Subscription.GetSubscriptionsForContent Method

Loads the subscriptions for the given content.

**C#**

```
public SubscriptionData[] GetSubscriptionsForContent(Long ContentId);
```

**Visual Basic**

```
Public Function GetSubscriptionsForContent(ByVal ContentId As Long) As SubscriptionData()
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | The content's ID. |

**Returns**

Ektron.Cms.SubscriptionData()

### Subscription.GetSubscriptionsForFolder Method

Loads the subscriptions for a given folder.

**C#**

```
public SubscriptionData[] GetSubscriptionsForFolder(Long FolderId);
```

**Visual Basic**

```
Public Function GetSubscriptionsForFolder(ByVal FolderId As Long) As SubscriptionData()
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderId | The folder's ID. |

**Returns**

Ektron.Cms.SubscriptionData()

### Subscription.SendWebAlert Method

Triggers the web alert subsystem. Requires that the subscription properties be set for the content first.

**C#**

```
public Boolean SendWebAlert(Long contentId);
```

**Visual Basic**

```
Public Function SendWebAlert(ByVal contentId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The ID of the content to set the subscription properties on. |

**Returns**

Boolean.

## Subscription.UpdateEmailFrom Method

Update the email address.

**C#**

```
public UpdateEmailFrom(Collection data);
```

**Visual Basic**

```
Public Sub UpdateEmailFrom(ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | email data |

## Subscription.UpdateSubscription Method

Updates the subscription.

**C#**

```
public UpdateSubscription(Collection data);
```

**Visual Basic**

```
Public Sub UpdateSubscription(ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | The subscription details. |

## Subscription.UpdateSubscriptionMessage Method

Updates the subscription message.

**C#**

```
public UpdateSubscriptionMessage(Collection data);
```

**Visual Basic**

```
Public Sub UpdateSubscriptionMessage(ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | The subscription message details. |

## Subscription.UpdateSubscriptionPropertiesForContent Method

Sets the subscription properties for a single piece of content.

**C#**

```
public Boolean UpdateSubscriptionPropertiesForContent(Long contentId,
SubscriptionPropertiesData subscriptionProperties);
```

**Visual Basic**

```
Public Function UpdateSubscriptionPropertiesForContent(ByVal contentId As Long, ByVal
subscriptionProperties As SubscriptionPropertiesData) As Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| contentId | The ID of the content to set the subscription properties on. |
| subscriptionProperties | The subscription properties class. |

### Returns

Boolean.

### Subscription.UpdateSubscriptionPropertiesForFolder Method

Sets the subscription properties for a single folder.

#### C#

```
public Boolean UpdateSubscriptionPropertiesForFolder(Long folderId, SubscriptionPropertiesData
subscriptionProperties);
```

#### Visual Basic

```
Public Function UpdateSubscriptionPropertiesForFolder(ByVal folderId As Long, ByVal
subscriptionProperties As SubscriptionPropertiesData) As Boolean
```

#### Parameters

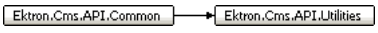
| Parameters | Description |
|---|---|
| folderId | The ID of the folder to set the subscription properties on. |
| subscriptionProperties | The subscription properties class. |

#### Returns

Boolean.

## Utilities Class

#### Class Hierarchy




#### C#

```
public class Utilities : Ektron.Cms.API.Common;
```

#### Visual Basic

```
Public Class Utilities
Inherits Ektron.Cms.API.Common
```

#### File

Utilities.vb

#### Description

This is class Ektron.Cms.API.Utilities.

## Utilities Methods

The methods of the Utilities class are listed here.

#### Public Methods

| | Name | Description |
|---|---|---|
| | LoadPermissions (see page 714) | Loads the permission for given object, such as content, folder, task, etc. |

### Utilities.LoadPermissions Method

Loads the permission for given object, such as content, folder, task, etc.

**C#**

```
[Obsolete("Use LoadPermissions from the Ektron.CMS.API.Permissions namespace.")]
public PermissionData LoadPermissions(Long Id, String RequestType, Optional ByVal
PermissionResultType Type);
```

**Visual Basic**

```
<Obsolete("Use LoadPermissions from the Ektron.CMS.API.Permissions namespace.")>
Public Function LoadPermissions(ByVal Id As Long, ByVal RequestType As String, Optional ByVal
Type As PermissionResultType = PermissionResultType.Common) As PermissionData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The object ID |
| RequestType | The object type := content \| folder \| collections \| tasks |
| Type | The permission result type |

**Returns**

Ektron.Cms.PermissionData

**Remarks**

Use LoadPermissions from the Ektron.CMS.API.Permissions (see page 690) namespace.

# Ektron.Cms.API.Analytics Namespace

This is namespace Ektron.Cms.API.Analytics.

**Classes**

| | Name | Description |
|---|---|---|
| | Analytics (see page 714) | A class for managing and querying CMS400.NET site activity information |

# Classes

The following table lists classes in this documentation.

**Classes**

| | Name | Description |
|---|---|---|
| | Analytics (see page 714) | A class for managing and querying CMS400.NET site activity information |

## Analytics Class

A class for managing and querying CMS400.NET site activity information

**Class Hierarchy**

**C#**

```
public class Analytics : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Analytics
```

```
Inherits Ektron.Cms.API.Common
```

**File**

Analytics.vb

## Analytics Methods

The methods of the Analytics class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | GetSessionRank (see page 715) | 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 715) | 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 715) | Provides a simple way of querying the database for analytics data. Returns the result as a weakly typed DataSet. Note that the connection is to the Analytics (see page 714) database and not to the standard CMS database. |
| | SoftDumpAnalyticsBuffer (see page 716) | 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 716) | This is the overview for the TrackVisit method overload. |

## Analytics.GetSessionRank Method

Gets the Session rank of the visiting user. A session rank of 0 is the first page a new user has visited. A session rank of 1 is the first page a returning user has visited. A session rank of 2 is any other page.

**C#**

```
public int GetSessionRank(System.Web.UI.Page page);
```

**Visual Basic**

```
Public Function GetSessionRank(ByVal page As System.Web.UI.Page) As Integer
```

**Parameters**

| Parameters | Description |
|---|---|
| page | |

## Analytics.HardDumpAnalyticsBuffer Method

Forces the analytics buffer to write to the database. This surpasses the checks to see if the time before write or number of records have passed.

**C#**

```
public HardDumpAnalyticsBuffer();
```

**Visual Basic**

```
Public Sub HardDumpAnalyticsBuffer()
```

## Analytics.QueryAnalytics Method

Provides a simple way of querying the database for analytics data. Returns the result as a weakly typed DataSet. Note that the connection is to the Analytics (see page 714) 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 buffer in memory to write to the database. If the time before write or number of records before write conditions have been met or exceeded, this function will force the write to the database.

### C#

```
public Boolean SoftDumpAnalyticsBuffer();
```

### Visual Basic

```
Public Function SoftDumpAnalyticsBuffer() As Boolean
```

## TrackVisit Method

This is the overview for the TrackVisit method overload.

### Description

Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer. This method is overloaded.

### Overload List

| | Name | Description |
|---|---|---|
| | Analytics.TrackVisit (Long, Long, String, Integer, String, String) (see page 716) | Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer (see page 716). |
| | Analytics.TrackVisit (System.Web.UI.Page, Long) (see page 717) | Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer (see page 716). |

## Analytics.TrackVisit Method (Long, Long, String, Integer, String, String)

Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer (see page 716).

### C#

```
public TrackVisit(Long content_id, Long user_id, String visitor_id, int session_rank, String
referring_url, String url);
```

### Visual Basic

```
Public Sub TrackVisit(ByVal content_id As Long, ByVal user_id As Long, ByVal visitor_id As
String, ByVal session_rank As Integer, ByVal referring_url As String, ByVal url As String)
```

### Parameters

| Parameters | Description |
|---|---|
| content_id | The ID of a content block that a visitor has viewed. |
| user_id | The CMS user ID of either a CMS User (see page 913) 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, Long)

Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer (see page 716).

**C#**

```
public TrackVisit(System.Web.UI.Page page, Long content_id);
```

**Visual Basic**

```
Public Sub TrackVisit(ByVal page As System.Web.UI.Page, ByVal content_id As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 block that a visitor has viewed. |

## Analytics Properties

The properties of the Analytics class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AnalyticsEnabled (see page 717) | Indicates if Analytics (see page 714) 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 18) server control. |
| | AppendToIISLog (see page 717) | 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 718) | The dataset that stores all the page/content view information in memory until it reaches its record or time threshold and writes back to the database. This is a strongly typed dataset of type 'ContentHitsData'. The underlying static member is initialized on demand. |
| | RecordsBeforeWrite (see page 718) | 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 718)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database. |
| | TimeBeforeWrite (see page 718) | 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 718)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database. |

## Analytics.AnalyticsEnabled Property

Indicates if Analytics (see page 714) 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 18) server control.

**C#**

```
public ReadOnly Boolean AnalyticsEnabled;
```

**Visual Basic**

```
Public ReadOnly Property AnalyticsEnabled() As Boolean
```

**Description**

True indicates that analytics data is being tracked. False indicates that no analytics data is being tracked.

## Analytics.AppendToIISLog Property

Indicates whether or not the web.config has been set to append user_id and content_id information to the IIS log.

**C#**

```
public ReadOnly Boolean AppendToIISLog;
```

**Visual Basic**

```
Public ReadOnly Property AppendToIISLog() As Boolean
```

**Description**

Boolean

### Analytics.InMemoryContentHitsData Property

The dataset that stores all the page/content view information in memory until it reaches its record or time threshold and writes back to the database. This is a strongly typed dataset of type 'ContentHitsData'. The underlying static member is initialized on demand.

**C#**

```
public ReadOnly ContentHitsData InMemoryContentHitsData;
```

**Visual Basic**

```
Public ReadOnly Property InMemoryContentHitsData() As ContentHitsData
```

### Analytics.RecordsBeforeWrite Property

Indicates the number of records to hold in the static memory buffer before writing to the database. This is used mainly to smooth out the number of writes to the database. This value and the 'TimeBeforeWrite (see page 718)' 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 718)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.

**C#**

```
public ReadOnly int TimeBeforeWrite;
```

**Visual Basic**

```
Public ReadOnly Property TimeBeforeWrite() As Integer
```

# Ektron.Cms.API.Community Namespace

The Community API (see page 639) contains classes that allow you to manage aspects of your community site.

**Classes**

| | Name | Description |
|---|---|---|
| | CommunityGroup (see page 719) | The CommunityGroup class allows you to work with community groups in CMS400.NET. For example, you can add a community group (AddCommunityGroup (see page 721)), get a list a list of community groups (GetAllCommunityGroups (see page 723)) or add a user to a community group (AddUserToCommunityGroup (see page 722)). |

| | | |
|---|---|---|
| | Favorites (see page 731) | The Favorites class allows you to work with a user's favorites. Favorites are content items that a user has designated as their favorite content. From this class, you can add content to a user's favorites (AddContentFavorite (see page 732)), remove content from a user's favorites (DeleteContentFavorite (see page 733)) or get a list of favorites for a given user (GetContentFavorites (see page 736)). |
| | Flagging (see page 739) | The Flagging class allows you to work with Flag (see page 1044) Objects, Flag (see page 1044) Entries and content that has been flagged. For example, you can assign a flag to content (AssignFlagToContent (see page 741)) or get a list of flags for a content item (GetContentFlagData (see page 743)). |
| | Friends (see page 745) | Friends, which are the same as Colleagues in CMS400.NET, allow users to associate themselves with other users in the community. This API (see page 639) class allows you to manage colleagues for users in the system. For example, you can add a colleague (AddFriend (see page 747)), remove a colleague (DeleteFriend (see page 748)) or get a list of current colleagues (GetFriends (see page 750)). |
| | MessageBoard (see page 755) | The MessageBoard class allows you to work with message board entries associated with a user, community group or content item. For example, you can add a message to a message board (AddMessageBoardEntry (see page 755)), delete a message (DeleteMessageBoardEntry (see page 762)) or get a list of message board items for an object (GetAllMessageBoardEntries (see page 764)). |
| | Tags (see page 771) | The Tags class allows you to manage tags for content, users, and community groups. For example, you can add a tag to the system (AddTag (see page 772)), add a tag to a user (AddTagToUser (see page 778)), get a list of tags associated with a user (GetTagsForUser (see page 787)) or delete a tag by its ID (DeleteTagById (see page 779)). |

# Classes

The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | CommunityGroup (see page 719) | The CommunityGroup class allows you to work with community groups in CMS400.NET. For example, you can add a community group (AddCommunityGroup (see page 721)), get a list a list of community groups (GetAllCommunityGroups (see page 723)) or add a user to a community group (AddUserToCommunityGroup (see page 722)). |
| | Favorites (see page 731) | The Favorites class allows you to work with a user's favorites. Favorites are content items that a user has designated as their favorite content. From this class, you can add content to a user's favorites (AddContentFavorite (see page 732)), remove content from a user's favorites (DeleteContentFavorite (see page 733)) or get a list of favorites for a given user (GetContentFavorites (see page 736)). |
| | Flagging (see page 739) | The Flagging class allows you to work with Flag (see page 1044) Objects, Flag (see page 1044) Entries and content that has been flagged. For example, you can assign a flag to content (AssignFlagToContent (see page 741)) or get a list of flags for a content item (GetContentFlagData (see page 743)). |
| | Friends (see page 745) | Friends, which are the same as Colleagues in CMS400.NET, allow users to associate themselves with other users in the community. This API (see page 639) class allows you to manage colleagues for users in the system. For example, you can add a colleague (AddFriend (see page 747)), remove a colleague (DeleteFriend (see page 748)) or get a list of current colleagues (GetFriends (see page 750)). |
| | MessageBoard (see page 755) | The MessageBoard class allows you to work with message board entries associated with a user, community group or content item. For example, you can add a message to a message board (AddMessageBoardEntry (see page 755)), delete a message (DeleteMessageBoardEntry (see page 762)) or get a list of message board items for an object (GetAllMessageBoardEntries (see page 764)). |
| | Tags (see page 771) | The Tags class allows you to manage tags for content, users, and community groups. For example, you can add a tag to the system (AddTag (see page 772)), add a tag to a user (AddTagToUser (see page 778)), get a list of tags associated with a user (GetTagsForUser (see page 787)) or delete a tag by its ID (DeleteTagById (see page 779)). |

## CommunityGroup Class

The CommunityGroup class allows you to work with community groups in CMS400.NET. For example, you can add a community group (AddCommunityGroup (see page 721)), get a list a list of community groups (GetAllCommunityGroups (see page 723)) or add a user to a community group (AddUserToCommunityGroup (see page 722)).

### Class Hierarchy

### C#

```
public class CommunityGroup : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class CommunityGroup
Inherits Ektron.Cms.API.Common
```

### File

CommunityGroup.vb

## CommunityGroup Methods

The methods of the CommunityGroup class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AcceptGroupInvite (see page 721) | Accepts a group invitation on behalf of invited user. |
| | AddCommunityGroup (see page 721) | Create a new community group. |
| | AddUserToCommunityGroup (see page 721) | This is the overview for the AddUserToCommunityGroup method overload. |
| | ApprovePendingGroupUser (see page 722) | Approves the user's pending group request. |
| | CancelJoinRequestForCommunityGroup (see page 722) | Cancels a group join request sent by the user. |
| | DeleteCommunityGroupByID (see page 723) | Deletes an existing community group. |
| | DeletePendingGroupUser (see page 723) | Delete or reject a user's pending community group request. |
| | GetAllCommunityGroups (see page 723) | Returns an array of Community (see page 718) Groups based upon supplied criteria. |
| | GetAllCommunityGroupsInDirectory (see page 724) | Retrieves a list of community groups in a Directory (see page 189). |
| | GetAllUnassignedCommunityGroupUsers (see page 724) | Returns a list of users who are not currently in a community group. |
| | GetCommunityGroupBlog (see page 724) | Returns the blog folder ID associated with a community group. |
| | GetCommunityGroupBlogIds (see page 724) | Gets a list of Blog (see page 792) IDs that belong to a community group. |
| | GetCommunityGroupByID (see page 725) | Retrieves a CommunityGroupData object by ID. |
| | GetCommunityGroupDiscussionBoardIds (see page 725) | Gets a list of discussion board IDs that belong to a community group. |
| | GetCommunityGroups (see page 725) | Gets a list of community groups a user belongs to. |
| | GetCommunityGroupUserIDs (see page 726) | Gets a list of User (see page 913) IDs that belong to a community group. |
| | GetCommunityGroupUsers (see page 726) | This is the overview for the GetCommunityGroupUsers method overload. |
| | GetFolderIdForCommunityGroup (see page 727) | Gets the Folder (see page 652) ID used by the community group in the CMS content tree. |
| | GetGroupMemberStatus (see page 727) | This method allows you to find out status of a user for a particular group. |
| | GetInvitedCommunityGroups (see page 728) | Returns a list of community groups that a user has been invited to join. |
| | GetMostPopularCommunityGroups (see page 728) | This is the overview for the GetMostPopularCommunityGroups method overload. |
| | GetPendingCommunityGroups (see page 729) | Gets a list of community groups for which the user's membership is pending the group administrator's approval. |
| | GetPendingCommunityGroupUsers (see page 729) | Returns a list of users whose community group membership is pending approval. |
| | InviteUsers (see page 730) | Invites the users to the community group. |
| | IsCommunityGroupAdmin (see page 730) | Returns true if the user is an administrator of the group. |

| | | |
|---|---|---|
| | IsUserInCommunityGroup (see page 730) | By providing the community group's ID and the user's ID, you can find out if a user belongs to a given community group. Returns true if the user is a member of the community group. |
| | RemoveUserFromCommunityGroup (see page 731) | Removes a user from a community group. |
| | UpdateCommunityGroup (see page 731) | Updates an existing community group. |

## CommunityGroup.AcceptGroupInvite Method

Accepts a group invitation on behalf of invited user.

**C#**

```
public AcceptGroupInvite(Long invitationId);
```

**Visual Basic**

```
Public Sub AcceptGroupInvite(ByVal invitationId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| invitationId | Id of group invitation to accept. |

## CommunityGroup.AddCommunityGroup Method

Create a new community group.

**C#**

```
public Long AddCommunityGroup(CommunityGroupData groupData);
```

**Visual Basic**

```
Public Function AddCommunityGroup(ByVal groupData As CommunityGroupData) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| groupData | CommunityGroupData class containing information about the new community group. |

**Returns**

ID of the new community group.

## AddUserToCommunityGroup Method

This is the overview for the AddUserToCommunityGroup method overload.

**Description**

This method is overloaded. The first method allows you to add a user to a community group by providing the community group's ID and the user's ID. The second, allows you to add a user to a community group and override the group enrollment moderation by providing the community group's ID, the user's ID and setting the overridePending parameter to true.

**Overload List**

| | Name | Description |
|---|---|---|
| | CommunityGroup.AddUserToCommunityGroup (Long, Long) (see page 722) | Adds a user to a community group. |
| | CommunityGroup.AddUserToCommunityGroup (Long, Long, Boolean) (see page 722) | Adds a user to a community group. You can override the group enrollment moderation by setting the overridePending parameter to true. |

### CommunityGroup.AddUserToCommunityGroup Method (Long, Long)

Adds a user to a community group.

**C#**

```
public AddUserToCommunityGroup(Long groupId, Long userId);
```

**Visual Basic**

```
Public Sub AddUserToCommunityGroup(ByVal groupId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of the community group to which the user will be added. |
| userId | ID of the user to add. |

**Remarks**

If the group does not have open enrollment, a group join request will be sent on behalf of the user.

### CommunityGroup.AddUserToCommunityGroup Method (Long, Long, Boolean)

Adds a user to a community group. You can override the group enrollment moderation by setting the overridePending parameter to true.

**C#**

```
public AddUserToCommunityGroup(Long groupId, Long userId, Boolean overridePending);
```

**Visual Basic**

```
Public Sub AddUserToCommunityGroup(ByVal groupId As Long, ByVal userId As Long, ByVal
overridePending As Boolean)
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of the community group to which the user will be added. |
| userId | ID of the user to be added. |
| overridePending | Set to true if you wish to bypass group enrollment moderation. |

### CommunityGroup.ApprovePendingGroupUser Method

Approves the user's pending group request.

**C#**

```
public ApprovePendingGroupUser(Long userId, Long groupId);
```

**Visual Basic**

```
Public Sub ApprovePendingGroupUser(ByVal userId As Long, ByVal groupId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of user to approve. |
| groupId | ID of the community group that the user is being approved to join. |

### CommunityGroup.CancelJoinRequestForCommunityGroup Method

Cancels a group join request sent by the user.

**C#**

```
public CancelJoinRequestForCommunityGroup(Long groupId, Long userId);
```

**Visual Basic**

```
Public Sub CancelJoinRequestForCommunityGroup(ByVal groupId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of the community group where the join request was sent. |
| userId | ID of the user whom initially sent join request. |

## CommunityGroup.DeleteCommunityGroupByID Method

Deletes an existing community group.

**C#**

```
public DeleteCommunityGroupByID(Long groupId);
```

**Visual Basic**

```
Public Sub DeleteCommunityGroupByID(ByVal groupId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | The ID of the community group to delete. |

## CommunityGroup.DeletePendingGroupUser Method

Delete or reject a user's pending community group request.

**C#**

```
public DeletePendingGroupUser(Long userId, Long groupId);
```

**Visual Basic**

```
Public Sub DeletePendingGroupUser(ByVal userId As Long, ByVal groupId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of user whose request you are deleting . |
| groupId | ID of community group from which the request is being deleted. |

## CommunityGroup.GetAllCommunityGroups Method

Returns an array of Community (see page 718) Groups based upon supplied criteria.

**C#**

```
public CommunityGroupData[] GetAllCommunityGroups(CommunityGroupRequest request);
```

**Visual Basic**

```
Public Function GetAllCommunityGroups(ByVal request As CommunityGroupRequest) As
CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| request | request criteria defining the groups to retrieve. |

**Returns**

An array of CommunityGroupData objects

### CommunityGroup.GetAllCommunityGroupsInDirectory Method

Retrieves a list of community groups in a Directory (see page 189).

**C#**

```
public CommunityGroupData[] GetAllCommunityGroupsInDirectory(ref DirectoryRequest request);
```

**Visual Basic**

```
Public Function GetAllCommunityGroupsInDirectory(ByRef request As DirectoryRequest) As
CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| request | Request criteria for community group retrieval. |

**Returns**

An array of CommunityGroupData objects.

### CommunityGroup.GetAllUnassignedCommunityGroupUsers Method

Returns a list of users who are not currently in a community group.

**C#**

```
public DirectoryUserData[] GetAllUnassignedCommunityGroupUsers(ref CommunityGroupObjectRequest
request);
```

**Visual Basic**

```
Public Function GetAllUnassignedCommunityGroupUsers(ByRef request As
CommunityGroupObjectRequest) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| request | Request criteria for user list. |

**Returns**

An array of DirectoryUserData objects.

### CommunityGroup.GetCommunityGroupBlog Method

Returns the blog folder ID associated with a community group.

**C#**

```
public Long GetCommunityGroupBlog(Long groupId);
```

**Visual Basic**

```
Public Function GetCommunityGroupBlog(ByVal groupId As Long) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | The community group ID. |

**Returns**

A Blog (see page 792)'s folder ID.

### CommunityGroup.GetCommunityGroupBlogIds Method

Gets a list of Blog (see page 792) IDs that belong to a community group.

**C#**

```
public Int64[] GetCommunityGroupBlogIds(Long communityGroupId);
```

**Visual Basic**

```
Public Function GetCommunityGroupBlogIds(ByVal communityGroupId As Long) As Int64()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | ID of community Group. |

**Returns**

An array of blog IDs (Long).

### CommunityGroup.GetCommunityGroupByID Method

Retrieves a CommunityGroupData object by ID.

**C#**

```
public CommunityGroupData GetCommunityGroupByID(Long groupId);
```

**Visual Basic**

```
Public Function GetCommunityGroupByID(ByVal groupId As Long) As CommunityGroupData
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | The ID of the community group to retrieve. |

**Returns**

CommunityGroupData if the groupId is valid, else null.

### CommunityGroup.GetCommunityGroupDiscussionBoardIds Method

Gets a list of discussion board IDs that belong to a community group.

**C#**

```
public Int64[] GetCommunityGroupDiscussionBoardIds(Long communityGroupId);
```

**Visual Basic**

```
Public Function GetCommunityGroupDiscussionBoardIds(ByVal communityGroupId As Long) As Int64()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | The ID of the community group. |

**Returns**

An array of Discussion Board IDs (Long).

### CommunityGroup.GetCommunityGroups Method

Gets a list of community groups a user belongs to.

**C#**

```
public CommunityGroupData[] GetCommunityGroups(Long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetCommunityGroups(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
```

```
CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of user to retrieve community groups for. |
| pagingInfo | Paging information for list retrieval. |

**Returns**

An array of CommunityGroupData objects.

### CommunityGroup.GetCommunityGroupUserIDs Method

Gets a list of User (see page 913) IDs that belong to a community group.

**C#**

```
public Long[] GetCommunityGroupUserIDs(Long communityGroupId);
```

**Visual Basic**

```
Public Function GetCommunityGroupUserIDs(ByVal communityGroupId As Long) As Long()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | The ID of the community group. |

**Returns**

An array of Discussion Board IDs (Long).

### GetCommunityGroupUsers Method

This is the overview for the GetCommunityGroupUsers method overload.

**Description**

This method is overloaded. The first method gets a list of users for a given community group. The second, gets a list of users for a given community group and gives you the ability to add paging to the list.

**Overload List**

| | Name | Description |
|---|---|---|
| | CommunityGroup.GetCommunityGroupUsers (Long) (see page 726) | Retrieves a list of users in a community group. |
| | CommunityGroup.GetCommunityGroupUsers (Long, PagingInfo) (see page 727) | Returns a list of users in a community group. |

### CommunityGroup.GetCommunityGroupUsers Method (Long)

Retrieves a list of users in a community group.

**C#**

```
public DirectoryUserData[] GetCommunityGroupUsers(Long communityGroupId);
```

**Visual Basic**

```
Public Function GetCommunityGroupUsers(ByVal communityGroupId As Long) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | The ID of the community group. |

**Returns**

An array of DirectoryUserData.

## CommunityGroup.GetCommunityGroupUsers Method (Long, PagingInfo)

Returns a list of users in a community group.

**C#**

```
public DirectoryUserData[] GetCommunityGroupUsers(Long communityGroupId, PagingInfo
pagingInfo);
```

**Visual Basic**

```
Public Function GetCommunityGroupUsers(ByVal communityGroupId As Long, ByVal pagingInfo As
PagingInfo) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | The ID of the community group. |
| pagingInfo | Paging information for the user list. |

**Returns**

An array of DirectoryUserData objects.

## CommunityGroup.GetFolderIdForCommunityGroup Method

Gets the Folder (see page 652) ID used by the community group in the CMS content tree.

**C#**

```
public Long GetFolderIdForCommunityGroup(Long CommunityGroupID);
```

**Visual Basic**

```
Public Function GetFolderIdForCommunityGroup(ByVal CommunityGroupID As Long) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| CommunityGroupID | ID of the community group. |

**Returns**

A numeric value representing the folder ID.

## CommunityGroup.GetGroupMemberStatus Method

This method allows you to find out status of a user for a particular group.

**C#**

```
public EkEnumeration.GroupMemberStatus GetGroupMemberStatus(Long communityGroupId, Long
userID);
```

**Visual Basic**

```
Public Function GetGroupMemberStatus(ByVal communityGroupId As Long, ByVal userID As Long) As
EkEnumeration.GroupMemberStatus
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | ID of community group to get a user's status. |
| userID | ID of the user. |

**Returns**

GroupMemberStatus as an enumeration.

## CommunityGroup.GetInvitedCommunityGroups Method

Returns a list of community groups that a user has been invited to join.

**C#**

```
public CommunityGroupData[] GetInvitedCommunityGroups(Long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetInvitedCommunityGroups(ByVal userId As Long, ByVal pagingInfo As
PagingInfo) As CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user who was invited. |
| pagingInfo | Object to supply paging parameters. |

**Returns**

An array of CommunityGroupData.

## GetMostPopularCommunityGroups Method

This is the overview for the GetMostPopularCommunityGroups method overload.

**Description**

This method is overloaded. The first method retrieves a list of the most popular community groups up to the number entered in the maxResults property. The second, retrieves all community groups ranked by popularity with the ability to have paging.

Note: The popularity of a group is determined by how many members belong to the group.

**Overload List**

| | Name | Description |
|---|---|---|
| | CommunityGroup.GetMostPopularCommunityGroups (Integer) (see page 728) | Retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a group is determined by how many members belong to the group. |
| | CommunityGroup.GetMostPopularCommunityGroups (PagingInfo) (see page 729) | Retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group. |

## CommunityGroup.GetMostPopularCommunityGroups Method (Integer)

Retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a group is determined by how many members belong to the group.

**C#**

```
public CommunityGroupData[] GetMostPopularCommunityGroups(int maxResults);
```

**Visual Basic**

```
Public Function GetMostPopularCommunityGroups(ByVal maxResults As Integer) As
CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| maxResults | The number of groups to retrieve. For example, if 10 is supplied, the top 10 most popular groups will be returned. |

**Returns**

An array of CommunityGroupData objects

### CommunityGroup.GetMostPopularCommunityGroups Method (PagingInfo)

Retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group.

**C#**

```
public CommunityGroupData[] GetMostPopularCommunityGroups(PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetMostPopularCommunityGroups(ByVal pagingInfo As PagingInfo) As
CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| pagingInfo | Paging information for paging through a large list of groups. |

**Returns**

an array of CommunityGroupData objects

### CommunityGroup.GetPendingCommunityGroups Method

Gets a list of community groups for which the user's membership is pending the group administrator's approval.

**C#**

```
public CommunityGroupData[] GetPendingCommunityGroups(Long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetPendingCommunityGroups(ByVal userId As Long, ByVal pagingInfo As
PagingInfo) As CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of user to retrieve pending groups for. |
| pagingInfo | Paging information for retrieval. |

### CommunityGroup.GetPendingCommunityGroupUsers Method

Returns a list of users whose community group membership is pending approval.

**C#**

```
public DirectoryUserData[] GetPendingCommunityGroupUsers(Long communityGroupID, PagingInfo
pagingInfo);
```

**Visual Basic**

```
Public Function GetPendingCommunityGroupUsers(ByVal communityGroupID As Long, ByVal pagingInfo
As PagingInfo) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupID | The ID of the community group. |
| pagingInfo | Paging information for the user list. |

**Returns**

An array of DirectoryUserData objects.

### CommunityGroup.InviteUsers Method

Invites the users to the community group.

**C#**

```
public System.Collections.ObjectModel.Collection(Of Ektron.Cms.Community.Invitation)
InviteUsers(InvitationSendRequestData invitationData, Long groupId);
```

**Visual Basic**

```
Public Function InviteUsers(ByVal invitationData As InvitationSendRequestData, ByVal groupId
As Long) As System.Collections.ObjectModel.Collection(Of Ektron.Cms.Community.Invitation)
```

**Parameters**

| Parameters | Description |
|---|---|
| invitationData | InvitationSendRequestData containing information about the GroupInvitation to send. |
| groupId | ID of the group to invite people to. |

**Returns**

A collection of Invitations that were sent out.

### CommunityGroup.IsCommunityGroupAdmin Method

Returns true if the user is an administrator of the group.

**C#**

```
public Boolean IsCommunityGroupAdmin(Long userId, Long groupId);
```

**Visual Basic**

```
Public Function IsCommunityGroupAdmin(ByVal userId As Long, ByVal groupId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user to be verified. |
| groupId | ID of the community group to check. |

### CommunityGroup.IsUserInCommunityGroup Method

By providing the community group's ID and the user's ID, you can find out if a user belongs to a given community group. Returns true if the user is a member of the community group.

**C#**

```
public Boolean IsUserInCommunityGroup(Long communityGroupId, Long userId);
```

**Visual Basic**

```
Public Function IsUserInCommunityGroup(ByVal communityGroupId As Long, ByVal userId As Long)
As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | ID of a community group to check membership. |
| userId | User (see page 913) ID to be checked. |

**Returns**

True if the user is a member of the community group.

### CommunityGroup.RemoveUserFromCommunityGroup Method

Removes a user from a community group.

**C#**

```
public RemoveUserFromCommunityGroup(Long groupId, Long userId);
```

**Visual Basic**

```
Public Sub RemoveUserFromCommunityGroup(ByVal groupId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of the community group from which the user is being removed. |
| userId | ID of the user to remove. |

### CommunityGroup.UpdateCommunityGroup Method

Updates an existing community group.

**C#**

```
public UpdateCommunityGroup(CommunityGroupData groupData);
```

**Visual Basic**

```
Public Sub UpdateCommunityGroup(ByVal groupData As CommunityGroupData)
```

**Parameters**

| Parameters | Description |
|---|---|
| groupData | CommunityGroupData class containing information to update for the group. |

## Favorites Class

The Favorites class allows you to work with a user's favorites. Favorites are content items that a user has designated as their favorite content. From this class, you can add content to a user's favorites (AddContentFavorite (see page 732)), remove content from a user's favorites (DeleteContentFavorite (see page 733)) or get a list of favorites for a given user (GetContentFavorites (see page 736)).

**Class Hierarchy**

**C#**

```
public class Favorites : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Favorites
Inherits Ektron.Cms.API.Common
```

**File**

Favorites.vb

## Favorites Methods

The methods of the Favorites class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddContentFavorite (see page 732) | This is the overview for the AddContentFavorite method overload. |
| | AddFavoriteLink (see page 733) | Adds a URI to a specified user's favorites. |
| | DeleteContentFavorite (see page 733) | Deletes a piece of content form a user's favorites. |
| | DeleteFavoriteLink (see page 734) | Delete a URI from a user's favorites. |
| | GetAllFavoriteLink (see page 734) | This is the overview for the GetAllFavoriteLink method overload. |
| | GetContentFavorites (see page 735) | This is the overview for the GetContentFavorites method overload. |
| | GetFavoriteId (see page 737) | This method checks to see if a piece of the URI is already in a user's favorites. If it is, it returns the ID. |
| | GetFavoriteLink (see page 738) | Retrieve the favorite by providing the favorite's ID. |
| | GetFavoritesDirectory (see page 738) | Returns user's Favorites (see page 731) taxonomy directory structure. |
| | IsContentFavorite (see page 738) | Checks if to see if a piece of content is already in a user's favorites. |
| | MoveContentFavorite (see page 739) | Moves a content favorite to a new favorites directory. |
| | MoveFavoriteLink (see page 739) | Moves a favorite from one favorite directory to another favorite directory. |
| | UpdateFavoriteLink (see page 739) | Update an existing URI favorite by providing the favorite's ID, user's ID, URI name, URI description and actual URI. |

## AddContentFavorite Method

This is the overview for the AddContentFavorite method overload.

### Description

This method is overloaded.

- AddContentFavorite (Long, Long) - adds a piece of content to user's favorites based on the content ID and user ID.
- AddContentFavorite (Long, Long, Long) - adds a piece of content to user's favorites and places it in a subdirectory based on the content ID, user ID and the ID of the subdirectory.

### Overload List

| | Name | Description |
|---|---|---|
| | Favorites.AddContentFavorite (Long, Long) (see page 732) | Adds a piece of content to user's favorites. |
| | Favorites.AddContentFavorite (Long, Long, Long) (see page 733) | Adds a piece of content to user's favorites and places it in a subdirectory. |

## Favorites.AddContentFavorite Method (Long, Long)

Adds a piece of content to user's favorites.

**C#**

```
public AddContentFavorite(Long contentId, Long userId);
```

**Visual Basic**

```
Public Sub AddContentFavorite(ByVal contentId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of content |
| userId | ID of user |

### Favorites.AddContentFavorite Method (Long, Long, Long)

Adds a piece of content to user's favorites and places it in a subdirectory.

**C#**

```
public AddContentFavorite(Long contentId, Long userId, Long directoryId);
```

**Visual Basic**

```
Public Sub AddContentFavorite(ByVal contentId As Long, ByVal userId As Long, ByVal directoryId
As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content item. |
| userId | ID of user adding the favorite. |
| directoryId | ID of the directory to add the content item to in the user's favorites. |

### Favorites.AddFavoriteLink Method

Adds a URI to a specified user's favorites.

**C#**

```
public Long AddFavoriteLink(Long userId, Long directoryId, int directoryLanguage, String
uriName, String Uri, String uriDescription);
```

**Visual Basic**

```
Public Function AddFavoriteLink(ByVal userId As Long, ByVal directoryId As Long, ByVal
directoryLanguage As Integer, ByVal uriName As String, ByVal Uri As String, ByVal
uriDescription As String) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user. |
| directoryId | A taxonomy ID. |
| directoryLanguage | A taxonomy language (for a user, this will be 0 - language neutral). |
| uriName | The name identification for the URI. |
| Uri | The actual URI. |
| uriDescription | A description of the URI. |

**Returns**

The newly created favorite ID as a Long.

### Favorites.DeleteContentFavorite Method

Deletes a piece of content form a user's favorites.

**C#**

```
public DeleteContentFavorite(Long contentId, Long userId);
```

**Visual Basic**

```
Public Sub DeleteContentFavorite(ByVal contentId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content item to be deleted from a user's favorites. |
| userId | ID of user deleting the favorite. |

### Favorites.DeleteFavoriteLink Method

Delete a URI from a user's favorites.

**C#**

```
public DeleteFavoriteLink(Long favoriteId);
```

**Visual Basic**

```
Public Sub DeleteFavoriteLink(ByVal favoriteId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| favoriteId | The ID of the favorite being deleted. |

### GetAllFavoriteLink Method

This is the overview for the GetAllFavoriteLink method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Favorites.GetAllFavoriteLink (Long) (see page 734) | Gets all the favorites for a given user. |
| | Favorites.GetAllFavoriteLink (Long, Long, Integer, Integer, Integer) (see page 735) | Gets all the favorites in user specified directory. |
| | Favorites.GetAllFavoriteLink (Long, Long, Integer, Integer, Integer, Integer) (see page 735) | Gets all the favorites in user specified directory. |

### Favorites.GetAllFavoriteLink Method (Long)

Gets all the favorites for a given user.

**C#**

```
public List(Of UriFavorite) GetAllFavoriteLink(Long userId);
```

**Visual Basic**

```
Public Function GetAllFavoriteLink(ByVal userId As Long) As List(Of UriFavorite)
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The user's ID. |

### Returns

A generic list of UriFavorite (see page 1417).

## Favorites.GetAllFavoriteLink Method (Long, Long, Integer, Integer, Integer)

Gets all the favorites in user specified directory.

### C#

```
public List(Of UriFavorite) GetAllFavoriteLink(Long userId, Long directoryId, int PageIndex,
int PageSize, ref int TotalPages);
```

### Visual Basic

```
Public Function GetAllFavoriteLink(ByVal userId As Long, ByVal directoryId As Long, ByVal
PageIndex As Integer, ByVal PageSize As Integer, ByRef TotalPages As Integer) As List(Of
UriFavorite)
```

### Parameters

| Parameters | Description |
|---|---|
| userId | The ID of the user. |
| directoryId | The ID of the directory. |

### Returns

A generic list of UriFavorite (see page 1417).

## Favorites.GetAllFavoriteLink Method (Long, Long, Integer, Integer, Integer, Integer)

Gets all the favorites in user specified directory.

### C#

```
public List(Of UriFavorite) GetAllFavoriteLink(Long userId, Long directoryId, int PageIndex,
int PageSize, ref int TotalPages, ref int linkCount);
```

### Visual Basic

```
Public Function GetAllFavoriteLink(ByVal userId As Long, ByVal directoryId As Long, ByVal
PageIndex As Integer, ByVal PageSize As Integer, ByRef TotalPages As Integer, ByRef linkCount
As Integer) As List(Of UriFavorite)
```

### Parameters

| Parameters | Description |
|---|---|
| userId | The ID of the user. |
| directoryId | The ID of the directory. |
| itemcount | total number of links in current and all its sub directories. |

### Returns

A generic list of UriFavorite (see page 1417).

## GetContentFavorites Method

This is the overview for the GetContentFavorites method overload.

### Description

This method is overloaded.

- GetContentFavorites (Long) - returns a list of the user's favorites based on a user ID.
- GetContentFavorites (Long, Long) - returns a list of the user's favorites in a directory based on the user ID and directory ID.
- GetContentFavorites (Long, Long, PagingInfo) - returns a list of the user's favorites in a given directory based on the user ID

and directory ID. This method allows for paging.

- GetContentFavorites (Long, PagingInfo) - returns a list of the user's favorites based on a user ID. This method allows for paging.

**Overload List**

| | Name | Description |
|---|---|---|
| | Favorites.GetContentFavorites (Long) (see page 736) | Returns a list of the user's favorites. |
| | Favorites.GetContentFavorites (Long, Long) (see page 736) | Returns a list of the user's favorites in a directory. |
| | Favorites.GetContentFavorites (Long, Long, PagingInfo) (see page 737) | Returns a list of the user's favorites in a given directory. |
| | Favorites.GetContentFavorites (Long, PagingInfo) (see page 737) | Returns a list of the user's favorites. |

## Favorites.GetContentFavorites Method (Long)

Returns a list of the user's favorites.

**C#**

```
public DirectoryContentData[] GetContentFavorites(Long userID);
```

**Visual Basic**

```
Public Function GetContentFavorites(ByVal userID As Long) As DirectoryContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userID | ID of user to retrieve favorites for. |

**Returns**

An array of DirectoryContentData objects.

## Favorites.GetContentFavorites Method (Long, Long)

Returns a list of the user's favorites in a directory.

**C#**

```
public DirectoryContentData[] GetContentFavorites(Long userId, Long directoryId);
```

**Visual Basic**

```
Public Function GetContentFavorites(ByVal userId As Long, ByVal directoryId As Long) As
DirectoryContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of a user to retrieve favorites for. |
| directoryId | ID of the directory to retrieve favorites from. |

**Returns**

An array of DirectoryContentData objects

### Favorites.GetContentFavorites Method (Long, Long, PagingInfo)

Returns a list of the user's favorites in a given directory.

**C#**

```
public DirectoryContentData[] GetContentFavorites(Long userId, Long directoryId, PagingInfo
pagingInfo);
```

**Visual Basic**

```
Public Function GetContentFavorites(ByVal userId As Long, ByVal directoryId As Long, ByVal
pagingInfo As PagingInfo) As DirectoryContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user to retrieve favorites for. |
| directoryId | ID of the directory to retrieve favorites from. |
| pagingInfo | Paging information for limiting the resultset. |

**Returns**

An array of DirectoryContentData objects

### Favorites.GetContentFavorites Method (Long, PagingInfo)

Returns a list of the user's favorites.

**C#**

```
public DirectoryContentData[] GetContentFavorites(Long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetContentFavorites(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
DirectoryContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user to retrieve favorites for. |
| pagingInfo | Paging information for limiting resultset. |

**Returns**

An array of DirectoryContentData objects.

### Favorites.GetFavoriteId Method

This method checks to see if a piece of the URI is already in a user's favorites. If it is, it returns the ID.

**C#**

```
public Long GetFavoriteId(String favoriteName, String favoriteUri, Long userId);
```

**Visual Basic**

```
Public Function GetFavoriteId(ByVal favoriteName As String, ByVal favoriteUri As String, ByVal
userId As Long) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| favoriteName | The favorite's name. |
| favoriteUri | The favorite's URI. |
| userId | The ID of user to whose favorites are being checked. |

**Returns**

If the favorite exists, it returns the ID; a non-negative value greater than 0.

### Favorites.GetFavoriteLink Method

Retrieve the favorite by providing the favorite's ID.

**C#**

```
public UriFavorite GetFavoriteLink(Long favoriteId);
```

**Visual Basic**

```
Public Function GetFavoriteLink(ByVal favoriteId As Long) As UriFavorite
```

**Parameters**

| Parameters | Description |
|---|---|
| favoriteId | The ID of the favorite. |

**Returns**

Returns information about the URI favorite in the format of UriFavorite (see page 1417).

### Favorites.GetFavoritesDirectory Method

Returns user's Favorites (see page 731) taxonomy directory structure.

**C#**

```
public TaxonomyData GetFavoritesDirectory(Long userId, Long directoryId, PagingInfo
pagingInfo);
```

**Visual Basic**

```
Public Function GetFavoritesDirectory(ByVal userId As Long, ByVal directoryId As Long, ByVal
pagingInfo As PagingInfo) As TaxonomyData
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user who's favorites should be retrieved. |
| directoryId | ID of the favorites directory to retrieve. |
| pagingInfo | Paging information for limiting the resultset. |

### Favorites.IsContentFavorite Method

Checks if to see if a piece of content is already in a user's favorites.

**C#**

```
public Boolean IsContentFavorite(Long contentId, Long userId);
```

**Visual Basic**

```
Public Function IsContentFavorite(ByVal contentId As Long, ByVal userId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of content to be checked. |
| userId | ID of user to check. |

**Returns**

True if the content item is in user's favorites, else false.

### Favorites.MoveContentFavorite Method

Moves a content favorite to a new favorites directory.

**C#**

```
public MoveContentFavorite(Long contentId, Long userId, Long directoryId);
```

**Visual Basic**

```
Public Sub MoveContentFavorite(ByVal contentId As Long, ByVal userId As Long, ByVal
directoryId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content item to move. |
| userId | ID of the user. |
| directoryId | ID of the directory to move content favorite it to. |

### Favorites.MoveFavoriteLink Method

Moves a favorite from one favorite directory to another favorite directory.

**C#**

```
public MoveFavoriteLink(Long favoriteId, Long sourceDir, Long destinationDir);
```

**Visual Basic**

```
Public Sub MoveFavoriteLink(ByVal favoriteId As Long, ByVal sourceDir As Long, ByVal
destinationDir As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| favoriteId | The ID of the favorite. |
| sourceDir | The ID of the current directory. |
| destinationDir | The ID of the new directory. |

### Favorites.UpdateFavoriteLink Method

Update an existing URI favorite by providing the favorite's ID, user's ID, URI name, URI description and actual URI.

**C#**

```
public UpdateFavoriteLink(Long favoriteId, Long userId, String uriName, String Uri, String
uriDescription);
```

**Visual Basic**

```
Public Sub UpdateFavoriteLink(ByVal favoriteId As Long, ByVal userId As Long, ByVal uriName As
String, ByVal Uri As String, ByVal uriDescription As String)
```

**Parameters**

| Parameters | Description |
|---|---|
| favoriteId | The ID of the favorite you are updating. |
| userId | The ID of user whose favorite is being updated. |
| uriName | The name identifiction for the new URI. |
| Uri | The actual URI. |
| uriDescription | A description of the URI. |

## Flagging Class

The Flagging class allows you to work with Flag (see page 1044) Objects, Flag (see page 1044) Entries and content that

has been flagged. For example, you can assign a flag to content (AssignFlagToContent (see page 741)) or get a list of flags for a content item (GetContentFlagData (see page 743)).

### Class Hierarchy

### C#

```
public class Flagging : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class Flagging
Inherits Ektron.Cms.API.Common
```

### File

Flagging.vb

## Flagging Methods

The methods of the Flagging class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddFlagObject (see page 740) | Saves a new Flag (see page 1044) on an Object. |
| | AssignFlagToContent (see page 741) | Assigns a flag definition to a piece of content. |
| | GetAllFlagEntries (see page 741) | This is the overview for the GetAllFlagEntries method overload. |
| | GetContentFlagData (see page 742) | This is the overview for the GetContentFlagData method overload. |
| | GetDefaultFolderFlagDef (see page 743) | Retrieves all the flag definitions applied to a folder. |
| | GetFlagDataById (see page 744) | Retrieves a flag entry on a piece of content. |
| | GetFlaggedRank (see page 744) | Retrieves a flag ranking for an object. |
| | PurgeFlagEntries (see page 744) | Purges all flags of a particular content block over a defined date/time range. |
| | RemoveObjectFlag (see page 744) | Removes a flag entry from an object. |
| | UpdateFlagObject (see page 745) | Updates an existing Flag (see page 1044) on an Object. |

## Flagging.AddFlagObject Method

Saves a new Flag (see page 1044) on an Object.

### C#

```
public Long AddFlagObject(ObjectFlagData objectFlag);
```

### Visual Basic

```
Public Function AddFlagObject(ByVal objectFlag As ObjectFlagData) As Long
```

### Parameters

| Parameters | Description |
|---|---|
| objectFlag | Object Flag (see page 1044) Data to save. |

### Returns

New Flag (see page 1044) Entry ID if successful, else 0.

## Flagging.AssignFlagToContent Method

Assigns a flag definition to a piece of content.

**C#**

```
public AssignFlagToContent(Long contentId, Long flagId);
```

**Visual Basic**

```
Public Sub AssignFlagToContent(ByVal contentId As Long, ByVal flagId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content to be assigned the flag. |
| flagId | ID of the flag definition. |

## GetAllFlagEntries Method

This is the overview for the GetAllFlagEntries method overload.

**Description**

This method is overloaded.

- GetAllFlagEntries (Integer, DateTime, DateTime, Integer) - gets a list of flag entries over a given time period based on the maximum number of results to retrieve, start and end dates and total flags.
- GetAllFlagEntries (Integer, Integer, DateTime, DateTime, Integer) - gets a list of flag entries over a given time period based on the content ID, maximum number of results to retrieve, start and end dates and total flags.

**Overload List**

| | Name | Description |
|---|---|---|
| | Flagging.GetAllFlagEntries (Integer, DateTime, DateTime, Integer) (see page 741) | Gets a list of flag entries over a given time period. |
| | Flagging.GetAllFlagEntries (Long, Integer, DateTime, DateTime, Integer) (see page 742) | Gets a list of flag entries for a piece of content over a given time period. |

## Flagging.GetAllFlagEntries Method (Integer, DateTime, DateTime, Integer)

Gets a list of flag entries over a given time period.

**C#**

```
public ContentFlagData[] GetAllFlagEntries(int maxResults, DateTime startDate, DateTime
endDate, ref int totalFlags);
```

**Visual Basic**

```
Public Function GetAllFlagEntries(ByVal maxResults As Integer, ByVal startDate As DateTime,
ByVal endDate As DateTime, ByRef totalFlags As Integer) As ContentFlagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| maxResults | Max number of results to retrieve. |
| startDate | Date to begin retrieving flags. |

| | |
|---|---|
| endDate | Date to end retrieving flags. |
| totalFlags | Returns the total number of flags found. |

**Returns**

Array of content flag data.

## Flagging.GetAllFlagEntries Method (Long, Integer, DateTime, DateTime, Integer)

Gets a list of flag entries for a piece of content over a given time period.

**C#**

```
public ContentFlagData[] GetAllFlagEntries(Long contentId, int maxResults, DateTime startDate,
DateTime endDate, ref int totalFlags);
```

**Visual Basic**

```
Public Function GetAllFlagEntries(ByVal contentId As Long, ByVal maxResults As Integer, ByVal
startDate As DateTime, ByVal endDate As DateTime, ByRef totalFlags As Integer) As
ContentFlagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of content item for which to retrieve flags. |
| maxResults | Max number of results to retrieve. |
| startDate | Date to begin retrieving flags. |
| endDate | Date to end retrieving flags. |
| totalFlags | Returns the total number of flags found. |

**Returns**

Array of content flag data.

## GetContentFlagData Method

This is the overview for the GetContentFlagData method overload.

**Description**

This method is overloaded.

- GetContentFlagData (Long, Long) - returns flagData for a piece of content for a given user based the content's ID and user's ID.
- GetContentFlagData (Long, String) - returns flagData for a piece of content for a given user based the content's ID and non-cms user's visitor ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Flagging.GetContentFlagData (Long, Long) (see page 743) | Returns flagData for a piece of content for a given user. |
| | Flagging.GetContentFlagData (Long, String) (see page 743) | Returns flagData for a piece of content for a given user. |

### Flagging.GetContentFlagData Method (Long, Long)

Returns flagData for a piece of content for a given user.

**C#**

```
public ContentFlagData GetContentFlagData(Long contentId, Long userId);
```

**Visual Basic**

```
Public Function GetContentFlagData(ByVal contentId As Long, ByVal userId As Long) As
ContentFlagData
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content to retrieve flag. |
| userId | ID of the CMS400.NET user who flagged content. |

**Returns**

ContentFlagData object.

### Flagging.GetContentFlagData Method (Long, String)

Returns flagData for a piece of content for a given user.

**C#**

```
public ContentFlagData GetContentFlagData(Long contentId, String visitorId);
```

**Visual Basic**

```
Public Function GetContentFlagData(ByVal contentId As Long, ByVal visitorId As String) As
ContentFlagData
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of content to retrieve flag for. |
| visitorId | Visitor ID of non-cms user. |

**Returns**

ContentFlagData object.

### Flagging.GetDefaultFolderFlagDef Method

Retrieves all the flag definitions applied to a folder.

**C#**

```
public FolderFlagDefData GetDefaultFolderFlagDef(Long folderId);
```

**Visual Basic**

```
Public Function GetDefaultFolderFlagDef(ByVal folderId As Long) As FolderFlagDefData
```

**Parameters**

| Parameters | Description |
|---|---|
| folderId | ID of the folder to retrieve flag definitions. |

**Returns**

array of FolderFlagDefData

### Flagging.GetFlagDataById Method

Retrieves a flag entry on a piece of content.

**C#**

```
public ContentFlagData GetFlagDataById(Long flagId);
```

**Visual Basic**

```
Public Function GetFlagDataById(ByVal flagId As Long) As ContentFlagData
```

**Parameters**

| Parameters | Description |
|---|---|
| flagId | ID of the flag entry. |

**Returns**

ContentFlagData

### Flagging.GetFlaggedRank Method

Retrieves a flag ranking for an object.

**C#**

```
public FlaggedRankData[] GetFlaggedRank(Long ObjectId);
```

**Visual Basic**

```
Public Function GetFlaggedRank(ByVal ObjectId As Long) As FlaggedRankData()
```

**Parameters**

| Parameters | Description |
|---|---|
| ObjectId | ID of the object for which to retrieve a flag ranking. |

**Returns**

FlaggedRankData

### Flagging.PurgeFlagEntries Method

Purges all flags of a particular content block over a defined date/time range.

**C#**

```
public PurgeFlagEntries(Long objectId, DateTime startDate, DateTime endDate);
```

**Visual Basic**

```
Public Sub PurgeFlagEntries(ByVal objectId As Long, ByVal startDate As DateTime, ByVal endDate
As DateTime)
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the content that is having its flags purged. |
| startDate | The beginning of the date range of content to be purged. |
| endDate | The end of the date range of content to be purged. |

### Flagging.RemoveObjectFlag Method

Removes a flag entry from an object.

**C#**

```
public RemoveObjectFlag(Long flagId);
```

**Visual Basic**

```
Public Sub RemoveObjectFlag(ByVal flagId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| flagId | ID of flag entry to remove. |

### Flagging.UpdateFlagObject Method

Updates an existing Flag (see page 1044) on an Object.

**C#**

```
public UpdateFlagObject(ObjectFlagData objectFlag);
```

**Visual Basic**

```
Public Sub UpdateFlagObject(ByVal objectFlag As ObjectFlagData)
```

**Parameters**

| Parameters | Description |
|---|---|
| objectFlag | Object flag data to update based upon the FlagEntryId. |

## Friends Class

Friends, which are the same as Colleagues in CMS400.NET, allow users to associate themselves with other users in the community. This API (see page 639) class allows you to manage colleagues for users in the system. For example, you can add a colleague (AddFriend (see page 747)), remove a colleague (DeleteFriend (see page 748)) or get a list of current colleagues (GetFriends (see page 750)).

**Class Hierarchy**

**C#**

```
public class Friends : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Friends
Inherits Ektron.Cms.API.Common
```

**File**

Friends.vb

### Friends Methods

The methods of the Friends class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AcceptFriendInvite (see page 746) | Accepts a friend invitation on behalf of invited user. |
| | AcceptPendingFriend (see page 746) | Accepts a colleague request from one user to another. |
| | AddFriend (see page 746) | This is the overview for the AddFriend method overload. |
| | AddPendingFriend (see page 747) | Adds a pending colleague. Sends the other user a colleague invitation. |
| | AddSelectedFriend (see page 748) | Adds a colleague to a user's selected colleague list by providing the selected colleague's user ID and the ID of the user who wants to add the selected colleague. |

| | | |
|---|---|---|
| | DeleteFriend (see page 748) | Delete's a colleague from a user's colleague list. |
| | DeletePendingFriendRequest (see page 748) | This is the overview for the DeletePendingFriendRequest method overload. |
| | DeleteSelectedFriend (see page 749) | Deletes a colleague from a user's selected colleague list by providing the selected colleague's user ID and the ID of the user who wants to delete the selected colleague. |
| | GetFriends (see page 749) | This is the overview for the GetFriends method overload. |
| | GetFriendsDirectory (see page 751) | Retrieves a user's colleague directory. |
| | GetFriendStatus (see page 752) | Get a user's colleague relation status to another user. For instance, user **A** may be a colleague to user **B**. User (see page 913) **C** may be a pending colleague to user **D**. |
| | GetInvitedUsers (see page 752) | Gets a list of users that a user has sent colleague invitations to. |
| | GetPendingFriends (see page 752) | Gets a list of users that have sent another user colleague invitations. This only returns users whose invitations have not yet been accepted and are still pending. |
| | GetSelectedFriends (see page 753) | Get a list of selected colleagues for a user by providing the user's ID. |
| | GetSentFriendInvitations (see page 753) | Gets a list of pending colleague invitations sent out by a user. |
| | InviteUsers (see page 753) | Send a friend invitation to a list of users and\or email addresses. |
| | IsFriend (see page 754) | Checks to see if a user is the colleague of another user by providing both colleague's ID information. In the friendUserID, enter the colleague's ID who you want to check. In the userID, enter the user you are checking against. Returns True when the users are colleagues. For example, if you want to find out whether Jim is a colleague of Tom, enter Jim's user ID in the friendUserId parameter and Tom's user ID in the userId parameter. If Jim and Tom are colleagues, this method returns true. |
| | IsSelectedFriend (see page 754) | Checks whether a user is a selected colleague of user. |
| | MoveFriend (see page 754) | Moves a colleague into another colleague directory. |

## Friends.AcceptFriendInvite Method

Accepts a friend invitation on behalf of invited user.

**C#**

```
public AcceptFriendInvite(Long invitationId);
```

**Visual Basic**

```
Public Sub AcceptFriendInvite(ByVal invitationId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| invitationId | Id of friend invitation to accept. |

## Friends.AcceptPendingFriend Method

Accepts a colleague request from one user to another.

**C#**

```
public AcceptPendingFriend(Long fromUserId, Long toUserId);
```

**Visual Basic**

```
Public Sub AcceptPendingFriend(ByVal fromUserId As Long, ByVal toUserId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| fromUserId | The ID of the user who sent the colleague request. |
| toUserId | The ID of the user who received colleague request and is accepting it. |

## AddFriend Method

This is the overview for the AddFriend method overload.

**Description**

This function is overloaded. The first method allows you to add one user to another's colleague list. The second, allows you to add a user to a specified folder in another user's colleague list.

**Overload List**

| | Name | Description |
|---|---|---|
| | Friends.AddFriend (Long, Long) (see page 747) | Adds a user to another user's colleague list. |
| | Friends.AddFriend (Long, Long, Long) (see page 747) | Adds a user to another user's friends list in a specific sub directory. |

## Friends.AddFriend Method (Long, Long)

Adds a user to another user's colleague list.

**C#**

```
public AddFriend(Long friendUserId, Long userId);
```

**Visual Basic**

```
Public Sub AddFriend(ByVal friendUserId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | The ID of the user to be added to a user's colleague list. |
| userId | The ID of the user adding the colleague. |

## Friends.AddFriend Method (Long, Long, Long)

Adds a user to another user's friends list in a specific sub directory.

**C#**

```
public AddFriend(Long friendUserId, Long userId, Long directoryId);
```

**Visual Basic**

```
Public Sub AddFriend(ByVal friendUserId As Long, ByVal userId As Long, ByVal directoryId As
Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | The ID of the user to be added to the user's colleague list. |
| userId | The ID of the user adding the colleague. |
| directoryId | The ID of the directory where the new colleague will be added. |

## Friends.AddPendingFriend Method

Adds a pending colleague. Sends the other user a colleague invitation.

**C#**

```
public AddPendingFriend(Long friendUserId, Long userId);
```

**Visual Basic**

```
Public Sub AddPendingFriend(ByVal friendUserId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | User (see page 913) receiving colleague request. |
| userId | User (see page 913) sending colleague request. |

## Friends.AddSelectedFriend Method

Adds a colleague to a user's selected colleague list by providing the selected colleague's user ID and the ID of the user who wants to add the selected colleague.

**C#**

```
public AddSelectedFriend(Long selectedFriendUserId, Long userId);
```

**Visual Basic**

```
Public Sub AddSelectedFriend(ByVal selectedFriendUserId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| selectedFriendUserId | User (see page 913) ID of a colleague to add to a user's selected colleague list. |
| userId | ID of the user adding the selected colleague. |

## Friends.DeleteFriend Method

Delete's a colleague from a user's colleague list.

**C#**

```
public DeleteFriend(Long friendUserId, Long userId);
```

**Visual Basic**

```
Public Sub DeleteFriend(ByVal friendUserId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | The user ID of the colleague to delete from the user's colleague list. |
| userId | The ID of the user deleting the colleague. |

## DeletePendingFriendRequest Method

This is the overview for the DeletePendingFriendRequest method overload.

**Description**

This method is overloaded. The first method deletes a pending colleague request from a user's pending colleague list by providing fromUserId and toUserId information. The second, deletes a pending colleague from a user's pending colleague list by providing fromUserId and toEmailAddress information.

**Overload List**

| | Name | Description |
|---|---|---|
| | Friends.DeletePendingFriendRequest (Long, Long) (see page 749) | Deletes a pending colleague request. |
| | Friends.DeletePendingFriendRequest (Long, String) (see page 749) | Deletes a pending colleague request. |

### Friends.DeletePendingFriendRequest Method (Long, Long)

Deletes a pending colleague request.

**C#**

```
public DeletePendingFriendRequest(Long fromUserId, Long toUserId);
```

**Visual Basic**

```
Public Sub DeletePendingFriendRequest(ByVal fromUserId As Long, ByVal toUserId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| fromUserId | The ID of the user who sent the original colleague request. |
| toUserId | The ID of the user who received the colleague request. |

### Friends.DeletePendingFriendRequest Method (Long, String)

Deletes a pending colleague request.

**C#**

```
public DeletePendingFriendRequest(Long fromUserId, String toEmailAddress);
```

**Visual Basic**

```
Public Sub DeletePendingFriendRequest(ByVal fromUserId As Long, ByVal toEmailAddress As String)
```

**Parameters**

| Parameters | Description |
|---|---|
| fromUserId | The ID of the user who sent original colleague request. |
| toEmailAddress | The Email address of the user who received the colleague request. |

### Friends.DeleteSelectedFriend Method

Deletes a colleague from a user's selected colleague list by providing the selected colleague's user ID and the ID of the user who wants to delete the selected colleague.

**C#**

```
public DeleteSelectedFriend(Long selectedFriendUserId, Long userId);
```

**Visual Basic**

```
Public Sub DeleteSelectedFriend(ByVal selectedFriendUserId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| selectedFriendUserId | User (see page 913) ID of a colleague to delete from user's selected colleagues list. |
| userId | ID of user deleting the colleague. |

### GetFriends Method

This is the overview for the GetFriends method overload.

**Description**

This function is overloaded.

- GetFriends (Long) - returns a list of colleagues based on a user's ID.
- GetFriends (Long, Long) - returns a list of colleagues in a specified directory based on a user's ID and the directory's ID.
- GetFriends (Long, PagingInfo) - returns a list of colleagues based on a user's ID. This method contains the ability to page results.

- GetFriends (Long, PagingInfo, Long) - returns a list of colleagues in a specified directory based on a user's ID and the directory's ID. This method contains the ability to page results.

**Overload List**

| | Name | Description |
|---|---|---|
| | Friends.GetFriends (Long) (see page 750) | Returns a list of colleagues for a user. |
| | Friends.GetFriends (Long, Long) (see page 750) | Returns a list of colleagues in a specified directory for a given user. |
| | Friends.GetFriends (Long, PagingInfo) (see page 751) | Returns a list of colleagues for a user. This method contains the ability to page results. |
| | Friends.GetFriends (Long, PagingInfo, Long) (see page 751) | Returns a list of colleagues for a user in a specific directory. This method contains the ability to page the results. |

## Friends.GetFriends Method (Long)

Returns a list of colleagues for a user.

**C#**

```
public DirectoryUserData[] GetFriends(Long userId);
```

**Visual Basic**

```
Public Function GetFriends(ByVal userId As Long) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user whose colleague list is being returned. |

**Returns**

An array of DirectoryUserData.

## Friends.GetFriends Method (Long, Long)

Returns a list of colleagues in a specified directory for a given user.

**C#**

```
public DirectoryUserData[] GetFriends(Long userId, Long directoryID);
```

**Visual Basic**

```
Public Function GetFriends(ByVal userId As Long, ByVal directoryID As Long) As
DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user whose colleague list is being returned. |
| directoryID | The ID of the directory from which to retrieve a colleague list. |

**Returns**

An array of DirectoryUserData.

### Friends.GetFriends Method (Long, PagingInfo)

Returns a list of colleagues for a user. This method contains the ability to page results.

**C#**

```
public DirectoryUserData[] GetFriends(Long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetFriends(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user whose colleague list is being returned. |
| pagingInfo | Supply paging information to control retrieval. |

**Returns**

An array of DirectoryUserData.

### Friends.GetFriends Method (Long, PagingInfo, Long)

Returns a list of colleagues for a user in a specific directory. This method contains the ability to page the results.

**C#**

```
public DirectoryUserData[] GetFriends(Long userId, PagingInfo pagingInfo, Long directoryID);
```

**Visual Basic**

```
Public Function GetFriends(ByVal userId As Long, ByVal pagingInfo As PagingInfo, ByVal
directoryID As Long) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user whose colleague list is being returned. |
| pagingInfo | supply paging information to control retrieval. |
| directoryID | The ID of the directory from which to retrieve a colleague list. |

**Returns**

array of DirectoryUserData.

### Friends.GetFriendsDirectory Method

Retrieves a user's colleague directory.

**C#**

```
public DirectoryAdvancedUserData GetFriendsDirectory(Long userId, Long directoryId, PagingInfo
pagingInfo);
```

**Visual Basic**

```
Public Function GetFriendsDirectory(ByVal userId As Long, ByVal directoryId As Long, ByVal
pagingInfo As PagingInfo) As DirectoryAdvancedUserData
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user whose colleague directory is being retrieved. |
| directoryId | ID of the colleague directory to retrieve. |
| pagingInfo | Supply paging information to control retrieval. |

**Returns**

DirectoryAdvancedUserData

### Friends.GetFriendStatus Method

Get a user's colleague relation status to another user. For instance, user **A** may be a colleague to user **B**. User (see page 913) **C** may be a pending colleague to user **D**.

**C#**

```
public EkEnumeration.FriendStatus GetFriendStatus(Long friendUserId, Long userId);
```

**Visual Basic**

```
Public Function GetFriendStatus(ByVal friendUserId As Long, ByVal userId As Long) As
EkEnumeration.FriendStatus
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | The ID of the user whose status is being checked. |
| userId | The ID of the user seeking status. |

**Returns**

FriendStatus enumeration.

### Friends.GetInvitedUsers Method

Gets a list of users that a user has sent colleague invitations to.

**C#**

```
public DirectoryUserData[] GetInvitedUsers(Long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetInvitedUsers(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of user who sent colleague invitations. |
| pagingInfo | Supply paging information to control retrieval. |

**Returns**

An array of DirectoryUserData.

### Friends.GetPendingFriends Method

Gets a list of users that have sent another user colleague invitations. This only returns users whose invitations have not yet been accepted and are still pending.

**C#**

```
public DirectoryUserData[] GetPendingFriends(Long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetPendingFriends(ByVal userId As Long, ByVal pagingInfo As PagingInfo) As
DirectoryUserData()
```

### Parameters

| Parameters | Description |
|---|---|
| userId | ID of user who has received invitations. |
| pagingInfo | Supply paging information to control retrieval. |

### Returns

array of DirectoryUserData

## Friends.GetSelectedFriends Method

Get a list of selected colleagues for a user by providing the user's ID.

### C#

```
public DirectoryUserData[] GetSelectedFriends(Long userId);
```

### Visual Basic

```
Public Function GetSelectedFriends(ByVal userId As Long) As DirectoryUserData()
```

### Parameters

| Parameters | Description |
|---|---|
| userId | ID of the user to retrieve selected colleagues. |

### Returns

An array of DirectoryUserData.

## Friends.GetSentFriendInvitations Method

Gets a list of pending colleague invitations sent out by a user.

### C#

```
public FriendInvitation[] GetSentFriendInvitations(Long userId, PagingInfo pagingInfo);
```

### Visual Basic

```
Public Function GetSentFriendInvitations(ByVal userId As Long, ByVal pagingInfo As PagingInfo)
As FriendInvitation()
```

### Parameters

| Parameters | Description |
|---|---|
| userId | ID of the user who sent invitations. |
| pagingInfo | Supply paging information to control retrieval. |

### Returns

AN array of FriendInvitations.

## Friends.InviteUsers Method

Send a friend invitation to a list of users and\or email addresses.

### C#

```
public ObjectModel.Collection(Of Ektron.Cms.Community.Invitation)
InviteUsers(InvitationSendRequestData invitationData);
```

### Visual Basic

```
Public Function InviteUsers(ByVal invitationData As InvitationSendRequestData) As
ObjectModel.Collection(Of Ektron.Cms.Community.Invitation)
```

**Parameters**

| Parameters | Description |
|---|---|
| invitationData | Invitation request data detailing invitation information. |

**Returns**

A collection of Invitation objects that have been sent out.

### Friends.IsFriend Method

Checks to see if a user is the colleague of another user by providing both colleague's ID information. In the friendUserID, enter the colleague's ID who you want to check. In the userID, enter the user you are checking against. Returns True when the users are colleagues. For example, if you want to find out whether Jim is a colleague of Tom, enter Jim's user ID in the friendUserId parameter and Tom's user ID in the userId parameter. If Jim and Tom are colleagues, this method returns true.

**C#**

```
public Boolean IsFriend(Long friendUserId, Long userId);
```

**Visual Basic**

```
Public Function IsFriend(ByVal friendUserId As Long, ByVal userId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | first userId |
| userId | second userId |

**Returns**

Returns true if FriendUserId and UserID are colleagues, else false.

### Friends.IsSelectedFriend Method

Checks whether a user is a selected colleague of user.

**C#**

```
public Boolean IsSelectedFriend(Long friendUserId, Long userId);
```

**Visual Basic**

```
Public Function IsSelectedFriend(ByVal friendUserId As Long, ByVal userId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | User (see page 913) ID to check in selected colleagues list. |
| userId | ID of the user who is checking for a selected colleague. |

**Returns**

Returns true if the colleague's user ID is in the selected colleagues list of for a user ID, else false.

### Friends.MoveFriend Method

Moves a colleague into another colleague directory.

**C#**

```
public MoveFriend(Long friendUserId, Long userId, Long directoryId);
```

**Visual Basic**

```
Public Sub MoveFriend(ByVal friendUserId As Long, ByVal userId As Long, ByVal directoryId As
```

```
Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | The user ID of the colleague to be moved. |
| userId | The ID of the user moving the colleague. |
| directoryId | The ID of the directory where the colleague will be moved. |

## MessageBoard Class

The MessageBoard class allows you to work with message board entries associated with a user, community group or content item. For example, you can add a message to a message board (AddMessageBoardEntry (see page 755)), delete a message (DeleteMessageBoardEntry (see page 762)) or get a list of message board items for an object (GetAllMessageBoardEntries (see page 764)).

**Class Hierarchy**

**C#**

```
public class MessageBoard : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class MessageBoard
Inherits Ektron.Cms.API.Common
```

**File**

MessageBoard.vb

### MessageBoard Methods

The methods of the MessageBoard class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddMessageBoardEntry (see page 755) | Adds a Message Board Entry |
| | ApproveMessage (see page 756) | This is the overview for the ApproveMessage method overload. |
| | DeleteMessageBoardEntries (see page 758) | This is the overview for the DeleteMessageBoardEntries method overload. |
| | DeleteMessageBoardEntriesByUser (see page 759) | This is the overview for the DeleteMessageBoardEntriesByUser method overload. |
| | DeleteMessageBoardEntry (see page 761) | This is the overview for the DeleteMessageBoardEntry method overload. |
| | GetAllMessageBoardEntries (see page 763) | This is the overview for the GetAllMessageBoardEntries method overload. |
| | GetAllMessageBoardEntriesByUser (see page 765) | This is the overview for the GetAllMessageBoardEntriesByUser method overload. |
| | GetMessageBoardEntry (see page 767) | This is the overview for the GetMessageBoardEntry method overload. |
| | IsUserMessageBoardAdmin (see page 769) | This function determines if the user has administrative privileges for the message board. |
| | UpdateMessageText (see page 769) | This is the overview for the UpdateMessageText method overload. |

### MessageBoard.AddMessageBoardEntry Method

Adds a Message Board Entry

**C#**

```
public MessageBoardData AddMessageBoardEntry(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, String messageText, Boolean isModerated, Long
callerUserId);
```

**Visual Basic**

```
Public Function AddMessageBoardEntry(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal messageText As String, ByVal isModerated As
Boolean, ByVal callerUserId As Long) As MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object that is associated with the message board. |
| objectType | The type of object. |
| messageText | the text of the message being added to the board. |
| isModerated | The moderation switch. |
| callerUserId | The user ID of the member requesting data. |

## ApproveMessage Method

This is the overview for the ApproveMessage method overload.

**Description**

This method is overloaded.

- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType) - change the flag of the message from pending to approved by providing the message ID, object ID and type.
- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - change the flag of the message from pending to approved by providing the message ID, object ID and type, and the ID of the user who changing the flag.
- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) - change the flag of the message from pending to approved by providing the message ID, object ID and type, the ID of the user who changing the flag and whether the message board is moderated.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType) (see page 756) | This function changes the flag of the message from pending to approve. |
| | MessageBoard.ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long) (see page 757) | This function changes the flag of the message from pending to approve. |
| | MessageBoard.ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) (see page 757) | This function changes the flag of the message from pending to approve. |

## MessageBoard.ApproveMessage Method (Long, Long, EkEnumeration.MessageBoardObjectType)

This function changes the flag of the message from pending to approve.

**C#**

```
public MessageBoardData ApproveMessage(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

**Visual Basic**

```
Public Function ApproveMessage(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType) As MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object. |
| objectType | The type of the object. |

**Returns**

Returns the updated row.

### MessageBoard.ApproveMessage Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long)

This function changes the flag of the message from pending to approve.

**C#**

```
public MessageBoardData ApproveMessage(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId);
```

**Visual Basic**

```
Public Function ApproveMessage(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long) As
MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object associated with the message board. |
| objectType | The type of the object associated with the message board. |
| callerUserId | The user ID of the member doing the update. |

**Returns**

Returns the updated row.

### MessageBoard.ApproveMessage Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean)

This function changes the flag of the message from pending to approve.

**C#**

```
public MessageBoardData ApproveMessage(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId, Boolean isModerated);
```

**Visual Basic**

```
Public Function ApproveMessage(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long, ByVal
isModerated As Boolean) As MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |

| | |
|---|---|
| objectId | The ID of the object associated with the message board. |
| objectType | The type of the object associated with the message board. |
| callerUserId | The user ID of the member doing the update. |
| isModerated | The moderation switch. |

**Returns**

Returns the updated row.

## DeleteMessageBoardEntries Method

This is the overview for the DeleteMessageBoardEntries method overload.

**Description**

This method is overloaded.

- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType) - deletes all the message board records for a particular object based on the object ID and type.
- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long) - deletes all the message board records for a particular object based on the object ID and type, and the user ID of the member deleting the messages.
- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) - deletes all the message board records for a particular object based on the object ID and type, the user ID of the member deleting the messages, and whether the message board is moderated.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType) (see page 758) | Deletes all the message board records for a particular object. |
| | MessageBoard.DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long) (see page 759) | Deletes all the message board records for a particular object. |
| | MessageBoard.DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) (see page 759) | Deletes all the message board records for a particular object. |

## MessageBoard.DeleteMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType)

Deletes all the message board records for a particular object.

**C#**

```
public Boolean DeleteMessageBoardEntries(Long objectId, EkEnumeration.MessageBoardObjectType
objectType);
```

**Visual Basic**

```
Public Function DeleteMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object that has the message board. |

| | |
|---|---|
| objectType | The type of object. |

**Returns**

Returns true when all the messages are deleted successfully.

### MessageBoard.DeleteMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, Long)

Deletes all the message board records for a particular object.

**C#**

```
public Boolean DeleteMessageBoardEntries(Long objectId, EkEnumeration.MessageBoardObjectType
objectType, Long callerUserId);
```

**Visual Basic**

```
Public Function DeleteMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The user ID of the member deleting the messages. |

**Returns**

Returns true when all the messages are deleted successfully.

### MessageBoard.DeleteMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, Long, Boolean)

Deletes all the message board records for a particular object.

**C#**

```
public Boolean DeleteMessageBoardEntries(Long objectId, EkEnumeration.MessageBoardObjectType
objectType, Long callerUserId, Boolean isModerated);
```

**Visual Basic**

```
Public Function DeleteMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long, ByVal isModerated As
Boolean) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The user ID of the member deleting the messages. |
| isModerated | The moderation switch. |

**Returns**

Returns true when all the messages are deleted successfully

### DeleteMessageBoardEntriesByUser Method

This is the overview for the DeleteMessageBoardEntriesByUser method overload.

**Description**

This method is overloaded.

- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long) - deletes all of a user's messages on a given message board based on the object ID and type, and the user's ID.
- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long) - deletes all of a user's messages on a given message board based on the object ID and type, the ID of the user whose messages are being removed and the ID of the user removing the messages.
- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Boolean) - deletes all of a user's messages on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and whether the message board is moderated.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long) (see page 760) | Deletes all the message board records for a particular User (see page 913). |
| | MessageBoard.DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long) (see page 760) | Deletes all the message board records for a particular User (see page 913). |
| | MessageBoard.DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Boolean) (see page 761) | Deletes all the message board records for a particular User (see page 913). |

## MessageBoard.DeleteMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long)

Deletes all the message board records for a particular User (see page 913).

**C#**

```
public Boolean DeleteMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId);
```

**Visual Basic**

```
Public Function DeleteMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object that has the message board. |
| objectType | The type of object. |
| userId | The ID of the user whose messages are going to be deleted. |

## MessageBoard.DeleteMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, Long)

Deletes all the message board records for a particular User (see page 913).

**C#**

```
public Boolean DeleteMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, Long callerUserId);
```

**Visual Basic**

```
Public Function DeleteMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal callerUserId As Long) As
Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| objectId | The ID of the object that has the message board. |
| objectType | The type of object. |
| userId | The ID of the user whose messages are going to be deleted. |
| callerUserId | The user ID of the member deleting the messages. |

## MessageBoard.DeleteMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Boolean)

Deletes all the message board records for a particular User (see page 913).

### C#

```
public Boolean DeleteMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, Long callerUserId, Boolean
isModerated);
```

### Visual Basic

```
Public Function DeleteMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal callerUserId As Long, ByVal
isModerated As Boolean) As Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| objectId | The ID of the object that has the message board. |
| objectType | The type of object. |
| userId | The ID of the user whose messages are going to be deleted. |
| callerUserId | The user ID of the member deleting the messages. |
| isModerated | The moderation switch. |

## DeleteMessageBoardEntry Method

This is the overview for the DeleteMessageBoardEntry method overload.

### Description

This method is overloaded.

- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) - deletes single message board record based on the message ID, object ID and object type.
- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - deletes single message board record based on the message ID, object ID, object type and user ID requesting the data.
- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) - deletes single message board record based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

### Overload List

| | Name | Description |
|---|---|---|
| | MessageBoard.DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) (see page 762) | Deletes single message board record based on the message ID, object ID and object type. |
| | MessageBoard.DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) (see page 762) | Deletes single message board record based on the message ID, object ID, object type and user ID requesting the data. |

| | | |
|---|---|---|
| | MessageBoard.DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) (see page 762) | Deletes single message board record based on the message ID, object ID, object type, the moderation switch and user ID requesting the data. |

### MessageBoard.DeleteMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType)

Deletes single message board record based on the message ID, object ID and object type.

**C#**

```
public Boolean DeleteMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

**Visual Basic**

```
Public Function DeleteMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object that is associated with the message board. |
| objectType | The type of object. |

**Returns**

Returns true when the message is deleted successfully.

### MessageBoard.DeleteMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long)

Deletes single message board record based on the message ID, object ID, object type and user ID requesting the data.

**C#**

```
public Boolean DeleteMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId);
```

**Visual Basic**

```
Public Function DeleteMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object that is associated with the message board. |
| objectType | The type of object. |
| callerUserId | The user ID of the member deleting the message. |

**Returns**

Returns true when the message is deleted successfully.

### MessageBoard.DeleteMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean)

Deletes single message board record based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

**C#**

```
public Boolean DeleteMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId, Boolean isModerated);
```

**Visual Basic**

```
Public Function DeleteMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long, ByVal
isModerated As Boolean) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The user ID of the member deleting the message. |
| isModerated | The moderation switch. |

**Returns**

Returns true when the message is deleted successfully.

## GetAllMessageBoardEntries Method

This is the overview for the GetAllMessageBoardEntries method overload.

**Description**

This method is overloaded.

- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo) - gets all messages for a particular object based on the object ID, object type and paging information.
- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long) - gets all messages for a particular object based on the object ID, object type, ID of the user requesting the list and paging information.
- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long, Boolean) - gets all messages for a particular object based on the object ID, object type, ID of the user requesting the list, paging information and the moderation switch.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo) (see page 764) | Gets all messages for a particular object. |
| | MessageBoard.GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long) (see page 764) | Gets all messages for a particular object. |
| | MessageBoard.GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long, Boolean) (see page 764) | Gets all messages for a particular object. |

### MessageBoard.GetAllMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, PagingInfo)

Gets all messages for a particular object.

**C#**

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntries(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging);
```

**Visual Basic**

```
Public Function GetAllMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal paging As PagingInfo) As Collection(Of
MessageBoardData)
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object. |
| objectType | The type of the object. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

### MessageBoard.GetAllMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long)

Gets all messages for a particular object.

**C#**

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntries(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging, Long callerUserId);
```

**Visual Basic**

```
Public Function GetAllMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal paging As PagingInfo, ByVal callerUserId As Long)
As Collection(Of MessageBoardData)
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object. |
| objectType | The type of the object. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |
| callerUserId | The user ID of the member requesting data. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

### MessageBoard.GetAllMessageBoardEntries Method (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long, Boolean)

Gets all messages for a particular object.

**C#**

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntries(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging, Long callerUserId, Boolean
isModerated);
```

**Visual Basic**

```
Public Function GetAllMessageBoardEntries(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal paging As PagingInfo, ByVal callerUserId As Long,
ByVal isModerated As Boolean) As Collection(Of MessageBoardData)
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object. |
| objectType | The type of the object. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |
| callerUserId | The user ID of the member requesting data. |
| isModerated | The moderation switch. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

## GetAllMessageBoardEntriesByUser Method

This is the overview for the GetAllMessageBoardEntriesByUser method overload.

**Description**

This method is overloaded.

- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Integer, PagingInfo) - gets all messages entered by a user on a given message board based on the object ID and type, the user's ID and paging information.
- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long) - gets all messages entered by a user on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and paging information.
- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long, Boolean) - gets all messages entered by a user on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and whether the message board is moderated.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo) (see page 765) | Gets all messages entered by a user. |
| | MessageBoard.GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long) (see page 766) | Gets all messages entered by a user. |
| | MessageBoard.GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long, Boolean) (see page 766) | Gets all messages entered by a user. |

## MessageBoard.GetAllMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo)

Gets all messages entered by a user.

**C#**

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, PagingInfo paging);
```

**Visual Basic**

```
Public Function GetAllMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal paging As PagingInfo) As
Collection(Of MessageBoardData)
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object. |
| objectType | The type of the object. |
| userId | The user ID for which to retrieve data. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

## MessageBoard.GetAllMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long)

Gets all messages entered by a user.

**C#**

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, PagingInfo paging, Long
callerUserId);
```

**Visual Basic**

```
Public Function GetAllMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal paging As PagingInfo, ByVal
callerUserId As Long) As Collection(Of MessageBoardData)
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object. |
| objectType | The type of the object. |
| userId | The user ID for which to retrieve data. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |
| callerUserId | The user ID of the member requesting data. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

## MessageBoard.GetAllMessageBoardEntriesByUser Method (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long, Boolean)

Gets all messages entered by a user.

**C#**

```
public Collection(Of MessageBoardData) GetAllMessageBoardEntriesByUser(Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long userId, PagingInfo paging, Long
callerUserId, Boolean isModerated);
```

**Visual Basic**

```
Public Function GetAllMessageBoardEntriesByUser(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long, ByVal paging As PagingInfo, ByVal
callerUserId As Long, ByVal isModerated As Boolean) As Collection(Of MessageBoardData)
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object. |
| objectType | The type of the object. |
| userId | The user ID for which to retrieve data. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |
| callerUserId | The user ID of the member requesting data. |
| isModerated | The moderation switch. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

## GetMessageBoardEntry Method

This is the overview for the GetMessageBoardEntry method overload.

**Description**

This method is overload.

GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) - gets a specific message from a message board based on the message ID, object ID and object type.

GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - gets a specific message from a message board based on the message ID, object ID, object type and user ID requesting the data.

GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) - gets a specific message from a message board based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) (see page 767) | Gets a specific message from a message board based on the message ID, object ID and object type. |
| | MessageBoard.GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) (see page 768) | Gets a specific message from a message board based on the message ID, object ID, object type and user ID requesting the data. |
| | MessageBoard.GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean) (see page 768) | Gets a specific message from a message board based on the message ID, object ID, object type, the moderation switch and user ID requesting the data. |

## MessageBoard.GetMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType)

Gets a specific message from a message board based on the message ID, object ID and object type.

**C#**

```
public MessageBoardData GetMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

**Visual Basic**

```
Public Function GetMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType) As MessageBoardData
```

### Parameters

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object that has the message board. |
| objectType | The type of object. |

### Returns

The message board data object with the data from the database.

## MessageBoard.GetMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long)

Gets a specific message from a message board based on the message ID, object ID, object type and user ID requesting the data.

### C#

```
public MessageBoardData GetMessageBoardEntry(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId);
```

### Visual Basic

```
Public Function GetMessageBoardEntry(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long) As
MessageBoardData
```

### Parameters

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The user ID of the member requesting data. |

### Returns

The message board data object with the data from the database.

## MessageBoard.GetMessageBoardEntry Method (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Boolean)

Gets a specific message from a message board based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

### C#

```
public MessageBoardData GetMessageBoardEntry(Long MessageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, Long callerUserId, Boolean isModerated);
```

### Visual Basic

```
Public Function GetMessageBoardEntry(ByVal MessageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal callerUserId As Long, ByVal
isModerated As Boolean) As MessageBoardData
```

### Parameters

| Parameters | Description |
|---|---|
| MessageId | The unique ID of the message. |
| objectId | The ID of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The user ID of the member requesting data. |
| isModerated | The moderation switch. |

### Returns

The message board data object with the data from the database.

## MessageBoard.IsUserMessageBoardAdmin Method

This function determines if the user has administrative privileges for the message board.

### C#

```
public Boolean IsUserMessageBoardAdmin(Long objectId, EkEnumeration.MessageBoardObjectType
objectType, Long userId);
```

### Visual Basic

```
Public Function IsUserMessageBoardAdmin(ByVal objectId As Long, ByVal objectType As
EkEnumeration.MessageBoardObjectType, ByVal userId As Long) As Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| userId | The User (see page 913) ID of the user |

### Returns

Returns true if the user has administrative privileges.

## UpdateMessageText Method

This is the overview for the UpdateMessageText method overload.

### Description

This method is overloaded.

- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String) - updates the text of a message based on the message ID, object ID and type, and the new text.
- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long) - updates the text of a message based on the message ID, object ID and type, the new text and the ID of the user changing the text.
- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long, Boolean) - updates the text of a message based on the message ID, object ID and type, the new text, the ID of the user changing the text and whether the board is moderated.

### Overload List

| | Name | Description |
|---|---|---|
| | MessageBoard.UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String) (see page 770) | Updates the Text of a message. |
| | MessageBoard.UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long) (see page 770) | Updates the Text of a message. |
| | MessageBoard.UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long, Boolean) (see page 770) | Updates the Text of a message. |

### MessageBoard.UpdateMessageText Method (Long, Long, EkEnumeration.MessageBoardObjectType, String)

Updates the Text of a message.

**C#**

```
public MessageBoardData UpdateMessageText(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, String messageText);
```

**Visual Basic**

```
Public Function UpdateMessageText(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal messageText As String) As
MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object associated with the message board. |
| objectType | The type of the object associated with the message board. |
| messageText | The new message text. |

**Returns**

Returns the updated row.

### MessageBoard.UpdateMessageText Method (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long)

Updates the Text of a message.

**C#**

```
public MessageBoardData UpdateMessageText(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, String messageText, Long callerUserId);
```

**Visual Basic**

```
Public Function UpdateMessageText(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal messageText As String, ByVal
callerUserId As Long) As MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object associated with the message board. |
| objectType | The type of the object associated with the message board. |
| messageText | The new message text. |
| callerUserId | The user ID of the member performing the update. |

**Returns**

Returns the updated row.

**Description**

'''

### MessageBoard.UpdateMessageText Method (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long, Boolean)

Updates the Text of a message.

**C#**

```
public MessageBoardData UpdateMessageText(Long messageId, Long objectId,
EkEnumeration.MessageBoardObjectType objectType, String messageText, Long callerUserId,
Boolean isModerated);
```

**Visual Basic**

```
Public Function UpdateMessageText(ByVal messageId As Long, ByVal objectId As Long, ByVal
objectType As EkEnumeration.MessageBoardObjectType, ByVal messageText As String, ByVal
callerUserId As Long, ByVal isModerated As Boolean) As MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique ID of the message. |
| objectId | The ID of the object associated with the message board. |
| objectType | The type of the object associated with the message board. |
| messageText | The new message text. |
| callerUserId | The user ID of the member doing the update. |
| isModerated | The moderation switch. |

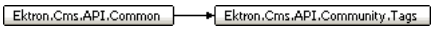
**Returns**

Returns the updated row.

## Tags Class

The Tags class allows you to manage tags for content, users, and community groups. For example, you can add a tag to the system (AddTag (see page 772)), add a tag to a user (AddTagToUser (see page 778)), get a list of tags associated with a user (GetTagsForUser (see page 787)) or delete a tag by its ID (DeleteTagById (see page 779)).

**Class Hierarchy**




**C#**

```
public class Tags : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Tags
Inherits Ektron.Cms.API.Common
```

**File**

Tags.vb

### Tags Methods

The methods of the Tags class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddTag (see page 772) | Saves a tag and returns an ID. If the tag already exists, the existing tag's ID is returned. |
| | AddTagToCommunityGroup (see page 772) | This is the overview for the AddTagToCommunityGroup method overload. |
| | AddTagToContent (see page 774) | This is the overview for the AddTagToContent method overload. |
| | AddTagToObject (see page 775) | This is the overview for the AddTagToObject method overload. |
| | AddTagToUser (see page 777) | This is the overview for the AddTagToUser method overload. |

| | | |
|---|---|---|
| | DeleteTagById (see page 779) | Permanently deletes tag and all associations. Even deletes system tags so this should only be called if user has permission to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser (see page 779). |
| | DeleteTagOnObject (see page 779) | Deletes a user's tag applied to an object. For example, User (see page 913) **A** has tagged content ID 101 with "CMS" and he now wishes to delete that object tag. |
| | DeleteTagsByUser (see page 779) | Deletes all tags applied by a user. To delete tags applied to a user, use DeleteTagsOnUser (see page 781). |
| | DeleteTagsOnCommunityGroup (see page 780) | Deletes all tags applied to a community group. |
| | DeleteTagsOnContent (see page 780) | Deletes all tags applied to a piece of content. |
| | DeleteTagsOnObject (see page 780) | Deletes all tags applied to an object. |
| | DeleteTagsOnUser (see page 781) | Deletes all tags applied to a user. To delete tags applied by a given user, use DeleteTagsByUser (see page 779). |
| | EditTag (see page 781) | Updates an existing Tag. |
| | GetAllTagRelatedContent (see page 781) | Returns a list of content related to the supplied content via a common tag. For example, both Content (see page 788) One and Content (see page 788) Two are both tagged with "CMS" and are considered related. |
| | GetAllTagRelatedUsers (see page 781) | Returns a list of users related to the supplied user via a common tag. For example, both User (see page 913) One and User (see page 913) Two are both tagged with "Engineer" and are considered related. |
| | GetAllTags (see page 782) | This is the overview for the GetAllTags method overload. |
| | GetDefaultTags (see page 783) | This is the overview for the GetDefaultTags method overload. |
| | GetTagByID (see page 784) | Returns a specific tag by providing it's ID. |
| | GetTagByName (see page 784) | Returns Tag with specified text. |
| | GetTagCloud (see page 785) | Returns a tag cloud based upon the request. |
| | GetTagsForContent (see page 785) | Returns an array of tags for a piece of content. |
| | GetTagsForObject (see page 785) | This is the overview for the GetTagsForObject method overload. |
| | GetTagsForUser (see page 786) | This is the overview for the GetTagsForUser method overload. |
| | RemoveTagAsDefault (see page 788) | Removes a tag from the object's default tag list. |
| | SaveTagAsDefault (see page 788) | Saves a tag to the the object's default list. |

## Tags.AddTag Method

Saves a tag and returns an ID. If the tag already exists, the existing tag's ID is returned.

### C#

```
public Long AddTag(ref TagData tag);
```

### Visual Basic

```
Public Function AddTag(ByRef tag As TagData) As Long
```

### Parameters

| Parameters | Description |
|---|---|
| tag | Tag data to save. |

### Returns

The ID of newly created tag or existing tag.

## AddTagToCommunityGroup Method

This is the overview for the AddTagToCommunityGroup method overload.

### Description

This method is overloaded.

- AddTagToCommunityGroup (String, Long) - applies a tag to a community group by providing the tag's text and the community group's ID.
- AddTagToCommunityGroup (String, Long, Integer) - applies a tag in a specified language to a community group by providing the tag's text, the community group's ID, and the language ID.
- AddTagToCommunityGroup (String, Long, Long, Integer) - applies a user created tag in a specified language to a community group by providing the tag's text, the community group's ID, the user ID and the language ID.

### Overload List

| | Name | Description |
|---|---|---|
| | Tags.AddTagToCommunityGroup (String, Long) (see page 773) | Applies a tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language. |
| | Tags.AddTagToCommunityGroup (String, Long, Integer) (see page 773) | Applies a tag to a community group. If the tag text does not already exist, it is created. Uses the currently logged in user. The newly created ID is accessible via the returned TagData. |
| | Tags.AddTagToCommunityGroup (String, Long, Long, Integer) (see page 774) | Applies a tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. |

## Tags.AddTagToCommunityGroup Method (String, Long)

Applies a tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

**C#**

```
public TagData AddTagToCommunityGroup(String tagText, Long groupId);
```

**Visual Basic**

```
Public Function AddTagToCommunityGroup(ByVal tagText As String, ByVal groupId As Long) As
TagData
```

### Parameters

| Parameters | Description |
|---|---|
| tagText | Text of tag to apply to object. |
| groupId | ID of the community that is being tagged. |

### Returns

TagData structure representing the applied tag.

## Tags.AddTagToCommunityGroup Method (String, Long, Integer)

Applies a tag to a community group. If the tag text does not already exist, it is created. Uses the currently logged in user. The newly created ID is accessible via the returned TagData.

**C#**

```
public TagData AddTagToCommunityGroup(String tagText, Long groupId, int languageId);
```

**Visual Basic**

```
Public Function AddTagToCommunityGroup(ByVal tagText As String, ByVal groupId As Long, ByVal languageId As Integer) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of tag to apply to object. |
| groupId | ID of the community group that is being tagged. |
| languageId | ID of the language the tagText should be saved as. Use -1 for all languages. |

**Returns**

TagData structure representing the applied tag.

## Tags.AddTagToCommunityGroup Method (String, Long, Long, Integer)

Applies a tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

**C#**

```
public TagData AddTagToCommunityGroup(String tagText, Long groupId, Long userId, int languageId);
```

**Visual Basic**

```
Public Function AddTagToCommunityGroup(ByVal tagText As String, ByVal groupId As Long, ByVal userId As Long, ByVal languageId As Integer) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of the tag to apply to an object. |
| groupId | ID of the community that is being tagged. |
| userId | ID of the user applying the tag. |
| languageId | ID of the language the tagText should be saved as. Use -1 for all languages. |

**Returns**

TagData structure representing the applied tag.

## AddTagToContent Method

This is the overview for the AddTagToContent method overload.

**Description**

This method is overloaded.

- AddTagToContent (String, Long) - applies a tag to a piece of content based on the tag's text and the content's ID.
- AddTagToContent (String, Long, Long, Integer) - applies a user created tag in a specified language to a content item by providing the tag's text, the content's ID, the user ID and the language ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.AddTagToContent (String, Long) (see page 775) | Applies a tag to a piece of content. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the tag is saved under the current site language. |

| | | |
|---|---|---|
| | Tags.AddTagToContent (String, Long, Long, Integer) (see page 775) | Applies a tag to a piece of content. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. |

### Tags.AddTagToContent Method (String, Long)

Applies a tag to a piece of content. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the tag is saved under the current site language.

**C#**

```
public TagData AddTagToContent(String tagText, Long contentId);
```

**Visual Basic**

```
Public Function AddTagToContent(ByVal tagText As String, ByVal contentId As Long) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of the tag to apply to an object. |
| contentId | ID of the piece of content that is being tagged. |

**Returns**

TagData structure representing the applied tag.

### Tags.AddTagToContent Method (String, Long, Long, Integer)

Applies a tag to a piece of content. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

**C#**

```
public TagData AddTagToContent(String tagText, Long contentId, Long userId, int languageId);
```

**Visual Basic**

```
Public Function AddTagToContent(ByVal tagText As String, ByVal contentId As Long, ByVal userId
As Long, ByVal languageId As Integer) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of tag to apply to object. |
| contentId | ID of the piece of content that is being tagged. |
| userId | ID of the user applying the tag. |
| languageId | ID of the language the tagText should be saved as. Use -1 for all languages. |

**Returns**

TagData structure representing the applied tag.

### AddTagToObject Method

This is the overview for the AddTagToObject method overload.

**Description**

This method is overloaded.

- AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes) - applies a tag to an object based on the tag's text and the object's type and ID.

- AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes, Integer, Integer) - applies a user created tag in a specified language to an object by providing the tag's text, the object's type and ID, and the language ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.AddTagToObject (Long, Long, EkEnumeration.CMSObjectTypes, Long, Integer) (see page 776) | Applies a tag to an object. |
| | Tags.AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes) (see page 776) | Applies a tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language. |
| | Tags.AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes, Long, Integer) (see page 777) | Applies a tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. |

## Tags.AddTagToObject Method (Long, Long, EkEnumeration.CMSObjectTypes, Long, Integer)

Applies a tag to an object.

**C#**

```
public AddTagToObject(Long tagId, Long objectId, EkEnumeration.CMSObjectTypes objectType, Long
userId, int languageId);
```

**Visual Basic**

```
Public Sub AddTagToObject(ByVal tagId As Long, ByVal objectId As Long, ByVal objectType As
EkEnumeration.CMSObjectTypes, ByVal userId As Long, ByVal languageId As Integer)
```

**Parameters**

| Parameters | Description |
|---|---|
| tagId | Id of the tag to apply to an object. |
| objectId | ID of object that is being tagged. |
| objectType | Type of object represented by objectId. |
| userId | ID of the user applying the tag. |
| languageId | ID of the language the tagText should be saved as. Use -1 for all languages. |

## Tags.AddTagToObject Method (String, Long, EkEnumeration.CMSObjectTypes)

Applies a tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

**C#**

```
public TagData AddTagToObject(String tagText, Long objectId, EkEnumeration.CMSObjectTypes
objectType);
```

**Visual Basic**

```
Public Function AddTagToObject(ByVal tagText As String, ByVal objectId As Long, ByVal
objectType As EkEnumeration.CMSObjectTypes) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of the tag to apply to an object. |

| | |
|---|---|
| objectId | ID of object that is being tagged. |
| objectType | Type of object represented by objectId. |

**Returns**

TagData structure representing the applied tag.

### Tags.AddTagToObject Method (String, Long, EkEnumeration.CMSObjectTypes, Long, Integer)

Applies a tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

**C#**

```
public TagData AddTagToObject(String tagText, Long objectId, EkEnumeration.CMSObjectTypes
objectType, Long userId, int languageId);
```

**Visual Basic**

```
Public Function AddTagToObject(ByVal tagText As String, ByVal objectId As Long, ByVal
objectType As EkEnumeration.CMSObjectTypes, ByVal userId As Long, ByVal languageId As Integer)
As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of the tag to apply to an object. |
| objectId | ID of object that is being tagged. |
| objectType | Type of object represented by objectId. |
| userId | ID of the user applying the tag. |
| languageId | ID of the language the tagText should be saved as. Use -1 for all languages. |

**Returns**

TagData structure representing the applied tag.

### AddTagToUser Method

This is the overview for the AddTagToUser method overload.

**Description**

This method is overloaded.

- AddTagToUser (Long, Long) - applies a tag to a user based on a tag's ID and the user's ID. Use this method when a tag already exists.
- AddTagToUser (String, Long) - applies a tag to a user based on the tag's text and the user's ID. Use this method when you are creating a new tag for a user.
- AddTagToUser (String, Long, Integer) - applies a tag in a specified language to a user by providing the tag's text, the user ID and the language ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.AddTagToUser (Long, Long) (see page 778) | Applies a Tag to a user based on a tag's ID and the user's ID. |
| | Tags.AddTagToUser (String, Long) (see page 778) | Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. The Tag is saved under the current site language. |

| | | |
|---|---|---|
| | Tags.AddTagToUser (String, Long, Integer) (see page 778) | Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. |

## Tags.AddTagToUser Method (Long, Long)

Applies a Tag to a user based on a tag's ID and the user's ID.

**C#**

```
public AddTagToUser(Long tagId, Long userId);
```

**Visual Basic**

```
Public Sub AddTagToUser(ByVal tagId As Long, ByVal userId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| tagId | ID of tag to apply to object. |
| userId | ID of the user that is being tagged. |

## Tags.AddTagToUser Method (String, Long)

Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. The Tag is saved under the current site language.

**C#**

```
public TagData AddTagToUser(String tagText, Long userId);
```

**Visual Basic**

```
Public Function AddTagToUser(ByVal tagText As String, ByVal userId As Long) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of the tag to apply to an object. |
| userId | ID of the user that is being tagged. |

**Returns**

TagData structure representing the applied tag.

## Tags.AddTagToUser Method (String, Long, Integer)

Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

**C#**

```
public TagData AddTagToUser(String tagText, Long userId, int languageId);
```

**Visual Basic**

```
Public Function AddTagToUser(ByVal tagText As String, ByVal userId As Long, ByVal languageId
As Integer) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of the tag to apply to an object. |
| userId | ID of the user that is being tagged. |
| languageId | ID of language the tag should be saved as. |

### Returns

TagData structure representing the applied tag.

## Tags.DeleteTagById Method

Permanently deletes tag and all associations. Even deletes system tags so this should only be called if user has permission to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser (see page 779).

### C#

```
public Boolean DeleteTagById(Long tagId);
```

### Visual Basic

```
Public Function DeleteTagById(ByVal tagId As Long) As Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| tagId | ID of tag to be deleted. |

## Tags.DeleteTagOnObject Method

Deletes a user's tag applied to an object. For example, User (see page 913) **A** has tagged content ID 101 with "CMS" and he now wishes to delete that object tag.

### C#

```
public Boolean DeleteTagOnObject(Long tagID, Long ObjectID, EkEnumeration.CMSObjectTypes
ObjectType, Long userId);
```

### Visual Basic

```
Public Function DeleteTagOnObject(ByVal tagID As Long, ByVal ObjectID As Long, ByVal
ObjectType As EkEnumeration.CMSObjectTypes, ByVal userId As Long) As Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| tagID | ID of tag to delete. |
| ObjectID | ID of the object to delete tags. |
| ObjectType | The type of object associated with the ObjectId. |
| userId | The user who originally applied the tag and wants to delete it. |

### Returns

True if operation was successful.

## Tags.DeleteTagsByUser Method

Deletes all tags applied by a user. To delete tags applied to a user, use DeleteTagsOnUser (see page 781).

### C#

```
public Boolean DeleteTagsByUser(Long userId);
```

### Visual Basic

```
Public Function DeleteTagsByUser(ByVal userId As Long) As Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| userId | ID of user whose tags are being deleted. |

**Returns**

True if operation was successful.

### Tags.DeleteTagsOnCommunityGroup Method

Deletes all tags applied to a community group.

**C#**

```
public Boolean DeleteTagsOnCommunityGroup(Long groupId);
```

**Visual Basic**

```
Public Function DeleteTagsOnCommunityGroup(ByVal groupId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of community group that has associated tags you wish to delete. |

**Returns**

True if operation was successful.

### Tags.DeleteTagsOnContent Method

Deletes all tags applied to a piece of content.

**C#**

```
public Boolean DeleteTagsOnContent(Long contentId);
```

**Visual Basic**

```
Public Function DeleteTagsOnContent(ByVal contentId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of a content item to delete associated tags. |

**Returns**

True if operation was successful.

### Tags.DeleteTagsOnObject Method

Deletes all tags applied to an object.

**C#**

```
public Boolean DeleteTagsOnObject(Long objectId, EkEnumeration.CMSObjectTypes objectType);
```

**Visual Basic**

```
Public Function DeleteTagsOnObject(ByVal objectId As Long, ByVal objectType As
EkEnumeration.CMSObjectTypes) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | ID of the object to delete associated tags. |
| objectType | The type of object associated with the ObjectId. |

**Returns**

True if operation was successful.

### Tags.DeleteTagsOnUser Method

Deletes all tags applied to a user. To delete tags applied by a given user, use DeleteTagsByUser (see page 779).

**C#**

```
public Boolean DeleteTagsOnUser(Long userId);
```

**Visual Basic**

```
Public Function DeleteTagsOnUser(ByVal userId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user whose associated tags are being deleted. |

**Returns**

True if operation was successful.

### Tags.EditTag Method

Updates an existing Tag.

**C#**

```
public TagData EditTag(ref TagData tag);
```

**Visual Basic**

```
Public Function EditTag(ByRef tag As TagData) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tag | Tag data to save. |

**Returns**

ID of newly created tag or existing tag.

### Tags.GetAllTagRelatedContent Method

Returns a list of content related to the supplied content via a common tag. For example, both Content (see page 788) One and Content (see page 788) Two are both tagged with "CMS" and are considered related.

**C#**

```
public ContentData[] GetAllTagRelatedContent(Long contentId);
```

**Visual Basic**

```
Public Function GetAllTagRelatedContent(ByVal contentId As Long) As ContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | This is the piece of content for which to retrieve related content. |

**Returns**

Returns array of partially populated ContentData (see page 1378) objects.

### Tags.GetAllTagRelatedUsers Method

Returns a list of users related to the supplied user via a common tag. For example, both User (see page 913) One and User (see page 913) Two are both tagged with "Engineer" and are considered related.

**C#**

```
public UserData[] GetAllTagRelatedUsers(Long userId);
```

**Visual Basic**

```
Public Function GetAllTagRelatedUsers(ByVal userId As Long) As UserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | This is the user for which to retrieve related users. |

**Returns**

Array of partially populated UserData (see page 1419) objects.

## GetAllTags Method

This is the overview for the GetAllTags method overload.

**Description**

This method is overloaded.

- GetAllTags (PagingInfo) - retrieves all tags in the system. This method allows for paging.
- GetAllTags (TagRequestData, Integer) - retrieves a list of tags in the system based on the requested information and the total number of tags requested.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.GetAllTags (PagingInfo) (see page 782) | Returns a list of all tags in the system. |
| | Tags.GetAllTags (TagRequestData, Integer) (see page 783) | Returns a list of all tags in the system. |

## Tags.GetAllTags Method (PagingInfo)

Returns a list of all tags in the system.

**C#**

```
public TagData[] GetAllTags(PagingInfo pagingInfo);
```

**Visual Basic**

```
Public Function GetAllTags(ByVal pagingInfo As PagingInfo) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| pagingInfo | Paging information for call. |

**Returns**

An array of TagData objects.

## Tags.GetAllTags Method (TagRequestData, Integer)

Returns a list of all tags in the system.

**C#**

```
public TagData[] GetAllTags(TagRequestData request, ref int totalTags);
```

**Visual Basic**

```
Public Function GetAllTags(ByVal request As TagRequestData, ByRef totalTags As Integer) As
TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| request | Contains the parameters for the tags to retrieve. |
| totalTags | Returns the total number of tags available for the requested parameters. |

**Returns**

Array of TagData objects

## GetDefaultTags Method

This is the overview for the GetDefaultTags method overload.

**Description**

This method is overloaded.

- GetDefaultTags (CMSObjectTypes) - gets default tags for a specified object.
- GetDefaultTags (CMSObjectTypes, Integer) - gets default tags in a specified language for an object.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.GetDefaultTags (CMSObjectTypes) (see page 783) | Returns a list of all system defined tags in the system. |
| | Tags.GetDefaultTags (CMSObjectTypes, Integer) (see page 784) | Returns a list of all system defined tags in the system. |

## Tags.GetDefaultTags Method (CMSObjectTypes)

Returns a list of all system defined tags in the system.

**C#**

```
public TagData[] GetDefaultTags(CMSObjectTypes objectType);
```

**Visual Basic**

```
Public Function GetDefaultTags(ByVal objectType As CMSObjectTypes) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| objectType | The type of object for which to retrieve tags. For example, retrieve all the default tags for CommunityGroups (CMSObjectTypes.CommunityGroup (see page 719)). |

**Returns**

Collection (see page 88) of TagData objects

## Tags.GetDefaultTags Method (CMSObjectTypes, Integer)

Returns a list of all system defined tags in the system.

**C#**

```
public TagData[] GetDefaultTags(CMSObjectTypes objectType, int languageId);
```

**Visual Basic**

```
Public Function GetDefaultTags(ByVal objectType As CMSObjectTypes, ByVal languageId As
Integer) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| objectType | The type of object for which to retrieve tags. For example, retrieve all the default tags for CommunityGroups (CMSObjectTypes.CommunityGroup (see page 719)). |
| languageId | Language of tags to retrieve. |

**Returns**

Collection (see page 88) of TagData objects

## Tags.GetTagByID Method

Returns a specific tag by providing it's ID.

**C#**

```
public TagData GetTagByID(Long tagId);
```

**Visual Basic**

```
Public Function GetTagByID(ByVal tagId As Long) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagId | ID of the tag to retrieve. |

**Returns**

Returns a tag found with supplied ID or new tag if no tag is found.

## Tags.GetTagByName Method

Returns Tag with specified text.

**C#**

```
public TagData GetTagByName(String tagText);
```

**Visual Basic**

```
Public Function GetTagByName(ByVal tagText As String) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of the tag to retrieve. |

**Returns**

Returns tag found with supplied ID or new tag if no tag is found.

## Tags.GetTagCloud Method

Returns a tag cloud based upon the request.

**C#**

```
public TagCloud GetTagCloud(TagCloudRequestData request, ref int totalTags);
```

**Visual Basic**

```
Public Function GetTagCloud(ByVal request As TagCloudRequestData, ByRef totalTags As Integer)
As TagCloud
```

**Parameters**

| Parameters | Description |
|---|---|
| request | Contains the parameters for the tag cloud to retrieve. |
| totalTags | Returns the total number of tags available in the requested cloud. |

**Returns**

A TagCloud (see page 493) object representing the request.

## Tags.GetTagsForContent Method

Returns an array of tags for a piece of content.

**C#**

```
public TagData[] GetTagsForContent(Long contentId);
```

**Visual Basic**

```
Public Function GetTagsForContent(ByVal contentId As Long) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The ID of the piece of content for which to retrieve tags. |

**Returns**

An array of TagData objects that have been applied to the content.

## GetTagsForObject Method

This is the overview for the GetTagsForObject method overload.

**Description**

This method is overloaded.

- GetTagsForObject (Long, CMSObjectTypes, Integer) - returns an array of tags for a given object based on the object's ID and type, and the language ID.
- GetTagsForObject (Long, EkEnumeration.CMSObjectTypes) - returns an array of tags for a given object based on the object's ID and type.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.GetTagsForObject (Long, CMSObjectTypes, Integer) (see page 786) | Returns an array of tags for a given object. |
| | Tags.GetTagsForObject (Long, EkEnumeration.CMSObjectTypes) (see page 786) | Returns an array of tags for a given object. |

## Tags.GetTagsForObject Method (Long, CMSObjectTypes, Integer)

Returns an array of tags for a given object.

**C#**

```
public TagData[] GetTagsForObject(Long objectId, CMSObjectTypes objectType, int languageId);
```

**Visual Basic**

```
Public Function GetTagsForObject(ByVal objectId As Long, ByVal objectType As CMSObjectTypes,
ByVal languageId As Integer) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The object ID to retrieve tags for. |
| objectType | The object type to retrieve tags for. |
| languageId | The language ID of the tags to retrieve. |

**Returns**

An array of TagData objects that have been applied to an object.

## Tags.GetTagsForObject Method (Long, EkEnumeration.CMSObjectTypes)

Returns an array of tags for a given object.

**C#**

```
public TagData[] GetTagsForObject(Long objectId, EkEnumeration.CMSObjectTypes objectType);
```

**Visual Basic**

```
Public Function GetTagsForObject(ByVal objectId As Long, ByVal objectType As
EkEnumeration.CMSObjectTypes) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the object to retrieve tags for. |
| objectType | The Object type to retrieve tags for. |

**Returns**

An array of TagData objects that have been applied to an object.

## GetTagsForUser Method

This is the overview for the GetTagsForUser method overload.

**Description**

This method is overloaded.

- GetTagsForUser (Long) - returns an array of tags for a user based on the user's ID.
- GetTagsForUser (Long, Integer) - returns an array of tags for a specific language associated with a user based on the user's ID and language ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.GetTagsForUser (Long) (see page 787) | Returns an array of tags for a user. |
| | Tags.GetTagsForUser (Long, Integer) (see page 787) | Returns an array of tags for a user. |

## Tags.GetTagsForUser Method (Long)

Returns an array of tags for a user.

**C#**

```
public TagData[] GetTagsForUser(Long userId);
```

**Visual Basic**

```
Public Function GetTagsForUser(ByVal userId As Long) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user to retrieve tags for. |

**Returns**

An array of TagData objects that have been applied to the user.

## Tags.GetTagsForUser Method (Long, Integer)

Returns an array of tags for a user.

**C#**

```
public TagData[] GetTagsForUser(Long userId, int languageId);
```

**Visual Basic**

```
Public Function GetTagsForUser(ByVal userId As Long, ByVal languageId As Integer) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user to retrieve tags for. |
| languageId | Language of tags to retrieve. |

**Returns**

An array of TagData objects that have been applied to the user.

### Tags.RemoveTagAsDefault Method

Removes a tag from the object's default tag list.

**C#**

```
public RemoveTagAsDefault(String tagText, int languageId, EkEnumeration.CMSObjectTypes
objectType);
```

**Visual Basic**

```
Public Sub RemoveTagAsDefault(ByVal tagText As String, ByVal languageId As Integer, ByVal
objectType As EkEnumeration.CMSObjectTypes)
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | The text of the tag. |
| languageId | The ID of the tag being removed. |
| objectType | The type of object. |

### Tags.SaveTagAsDefault Method

Saves a tag to the the object's default list.

**C#**

```
public TagData SaveTagAsDefault(String tagText, int languageId, EkEnumeration.CMSObjectTypes
objectType);
```

**Visual Basic**

```
Public Function SaveTagAsDefault(ByVal tagText As String, ByVal languageId As Integer, ByVal
objectType As EkEnumeration.CMSObjectTypes) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | The text of the tag. |
| languageId | The ID of the language to which the tag will be assigned. |
| objectType | The type of object. |

# Ektron.Cms.API.Content Namespace

This is namespace Ektron.Cms.API.Content.

**Classes**

| | Name | Description |
|---|---|---|
| | Asset (see page 789) | This is class Ektron.Cms.API.Content.Asset. |
| | Blog (see page 792) | This is class Ektron.Cms.API.Content.Blog. |
| | Content (see page 806) | This is class Ektron.Cms.API.Content.Content. |
| | ContentRating (see page 831) | This is class Ektron.Cms.API.Content.ContentRating. |
| | Form (see page 836) | This is class Ektron.Cms.API.Content.Form. |
| | Taxonomy (see page 842) | This is class Ektron.Cms.API.Content.Taxonomy. |
| | ThreadedDiscussion (see page 848) | The ThreadedDiscussion class allows you to work with threaded discussion by adding and removing discussion boards, forums, and topics. It also allows you to subscribe or unsubscribe to a board, forum or topic. In addition, you can retrieve a specific topic or search a topic's replies.<br>The Threaded Discussion feature provides a discussion board where a forum can be created and topics can be discussed on your Web site. A site or membership user with Add Topic permission creates a topic and posts a starter question. Other site visitors can then reply to the topic. |

# Classes

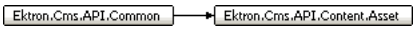
The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | Asset (see page 789) | This is class Ektron.Cms.API.Content.Asset. |
| | Blog (see page 792) | This is class Ektron.Cms.API.Content.Blog. |
| | Content (see page 806) | This is class Ektron.Cms.API.Content.Content. |
| | ContentRating (see page 831) | This is class Ektron.Cms.API.Content.ContentRating. |
| | Form (see page 836) | This is class Ektron.Cms.API.Content.Form. |
| | Taxonomy (see page 842) | This is class Ektron.Cms.API.Content.Taxonomy. |
| | ThreadedDiscussion (see page 848) | The ThreadedDiscussion class allows you to work with threaded discussion by adding and removing discussion boards, forums, and topics. It also allows you to subscribe or unsubscribe to a board, forum or topic. In addition, you can retrieve a specific topic or search a topic's replies.<br>The Threaded Discussion feature provides a discussion board where a forum can be created and topics can be discussed on your Web site. A site or membership user with Add Topic permission creates a topic and posts a starter question. Other site visitors can then reply to the topic. |

## Asset Class

### Class Hierarchy




### C#

```
public class Asset : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class Asset
Inherits Ektron.Cms.API.Common
```

### File

Asset.vb

### Description

This is class Ektron.Cms.API.Content.Asset.

## Asset Methods

The methods of the Asset class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | GetAssetMgtConfigInfo (see page 790) | Loads all of the registered plug-in configuration details. This method is a member of Ektron.Cms.API.Content.Asset (see page 789) |
| | GetAssetMgtInfo (see page 790) | Loads all of the registered plug-in type details. This method is a member of Ektron.Cms.API.Content.Asset (see page 789) |
| | GetAssetSupertypes (see page 790) | Loads all of the asset supertypes supported by the documents add-in. This method is a member of Ektron.Cms.API.Content.Asset (see page 789) |
| | GetEditSnippet (see page 790) | Loads the edit snippet for the given document. Used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 789) |
| | GetMyAssetMgtID (see page 791) | Retrieves the Document Management functionality client ID. This method is a member of Ektron.Cms.API.Content.Asset (see page 789) |
| | GetViewUrl (see page 791) | Returns a URL associated with the current asset. This method is a member of Ektron.Cms.API.Content.Asset (see page 789) |

| | | |
|---|---|---|
| | MakeCopyDMSAsset (see page 791) | Create a copy of a DMS Asset (see page 789). This method is a member of Ektron.Cms.API.Content.Asset (see page 789) |
| | SetAssetMgtConfigInfo (see page 792) | Updates the Document Management configuration information. This method is a member of Ektron.Cms.API.Content.Asset (see page 789) |

### Asset.GetAssetMgtConfigInfo Method

Loads all of the registered plug-in configuration details. This method is a member of Ektron.Cms.API.Content.Asset (see page 789)

**C#**

```
public AssetConfigInfo[] GetAssetMgtConfigInfo();
```

**Visual Basic**

```
Public Function GetAssetMgtConfigInfo() As AssetConfigInfo()
```

**Returns**

Ektron.Cms.AssetConfigInfo()

### Asset.GetAssetMgtInfo Method

Loads all of the registered plug-in type details. This method is a member of Ektron.Cms.API.Content.Asset (see page 789)

**C#**

```
public AssetInfoData[] GetAssetMgtInfo();
```

**Visual Basic**

```
Public Function GetAssetMgtInfo() As AssetInfoData()
```

**Returns**

Ektron.Cms.AssetInfoData()

### Asset.GetAssetSupertypes Method

Loads all of the asset supertypes supported by the documents add-in. This method is a member of Ektron.Cms.API.Content.Asset (see page 789)

**C#**

```
public AssetInfoData[] GetAssetSupertypes();
```

**Visual Basic**

```
Public Function GetAssetSupertypes() As AssetInfoData()
```

**Returns**

Ektron.Cms.AssetInfoData()

### Asset.GetEditSnippet Method

Loads the edit snippet for the given document. Used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 789)

**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, such as office documents or managed files or * (all). |

**Returns**

String

**Remarks**

The Document Management functionality with a valid license is required to perform this operation.

### Asset.GetMyAssetMgtID Method

Retrieves the Document Management functionality client ID. This method is a member of Ektron.Cms.API.Content.Asset (see page 789)

**C#**

```
public String GetMyAssetMgtID();
```

**Visual Basic**

```
Public Function GetMyAssetMgtID() As String
```

**Returns**

String

### Asset.GetViewUrl Method

Returns a URL associated with the current asset. This method is a member of Ektron.Cms.API.Content.Asset (see page 789)

**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, such as office documents or managed files or * (all). |

**Returns**

String

**Remarks**

The Document Management functionality with a valid license is required to perform this operation.

### Asset.MakeCopyDMSAsset Method

Create a copy of a DMS Asset (see page 789). This method is a member of Ektron.Cms.API.Content.Asset (see page 789)

**C#**

```
public Boolean MakeCopyDMSAsset(ref String strAssetID, ref String strAssetVersion);
```

**Visual Basic**

```
Public Function MakeCopyDMSAsset(ByRef strAssetID As String, ByRef strAssetVersion As String)
```

```
As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| strAssetID | ID associated with the asset |
| strAssetVersion | Version of the asset |

**Returns**

boolean

### Asset.SetAssetMgtConfigInfo Method

Updates the Document Management configuration information. This method is a member of Ektron.Cms.API.Content.Asset (see page 789)

**C#**

```
public Boolean SetAssetMgtConfigInfo(AssetConfigInfo asset_config);
```

**Visual Basic**

```
Public Function SetAssetMgtConfigInfo(ByVal asset_config() As AssetConfigInfo) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| asset_config | The Document Management configuration data |

**Returns**

Boolean

## Blog Class

**Class Hierarchy**

**C#**

```
public class Blog : Content;
```

**Visual Basic**

```
Public Class Blog
Inherits Content
```

**File**

Blog.vb

**Description**

This is class Ektron.Cms.API.Content.Blog.

### Blog Methods

The methods of the Blog class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddBlog (see page 793) | Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog. |
| | AddCommentForPost (see page 794) | 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 794) | Adds a post to a given blog. Accepts a ContentData (see page 1378) class as the post content. Return new post ID.<br>*See Also:* The Example section. |
| | DeleteBlog (see page 796) | Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0. |
| | DeleteBlogPostComment (see page 796) | Delete Blog (see page 792) post comment by providing the comment's ID. |
| | DeletePost (see page 796) | 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 796) | Returns a blank BlogPostData class. Used Primarily in the Workarea. |
| | GetBlog (see page 797) | This is the overview for the GetBlog method overload. |
| | GetBlogData (see page 800) | This is the overview for the GetBlogData method overload. |
| | GetBlogDataForPost (see page 803) | 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 803) | 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 804) | This is the overview for the GetBlogRoll method overload. |
| | GetPost (see page 805) | Gets a blog post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData (see page 1378) Class. |
| | GetPostbyID (see page 805) | Gets a blog post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class. |
| | GetUserBlog (see page 806) | gets the user blog id |
| | UpdateBlogProperties (see page 806) | Updates the properties of a blog. Accepts standard blogdata object. Returns boolean. |

## Blog.AddBlog Method

Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.

### C#

```
public Long AddBlog(Long ParentFolderID, String BlogName, String BlogTitle, String
BlogDescription, Ektron.Cms.Common.EkEnumeration.BlogVisibility Visibility, Boolean
EnableComments, Boolean ModerateComments, Boolean RequireAuthentication, String Categories,
BlogRoll RollLinks);
```

### Visual Basic

```
Public Function AddBlog(ByVal ParentFolderID As Long, ByVal BlogName As String, ByVal
BlogTitle As String, ByVal BlogDescription As String, ByVal Visibility As
Ektron.Cms.Common.EkEnumeration.BlogVisibility, ByVal EnableComments As Boolean, ByVal
ModerateComments As Boolean, ByVal RequireAuthentication As Boolean, ByVal Categories() As
String, ByVal RollLinks As BlogRoll) As Long
```

### Parameters

| Parameters | Description |
|---|---|
| ParentFolderID | |
| BlogName | Name of the Blog (see page 792). |
| BlogTitle | Title of the Blog (see page 792). |
| BlogDescription | Description of the Blog (see page 792). |
| Visibility | If the blog is public, enter Public. If private, enter Private. |
| EnableComments | Enter True to allow commenting, otherwise enter false. |
| ModerateComments | If commenting is allowed, enter True to moderate comments. |
| RequireAuthentication | If commenting is allowed, require authentication to add comment. |
| Categories | String Array representing the categories for this blog. |
| RollLinks | The Blog (see page 792)'s roll links. |

### Returns

Numeric value of the blog ID that is greater than 0 (zero).

## Blog.AddCommentForPost Method

Adds a post comment to a blog post. Accepts a BlogComment class as the input. Will return the result of that add action. False indicates the action was successful.

**C#**

```
public Boolean AddCommentForPost(BlogComment bcComment);
```

**Visual Basic**

```
Public Function AddCommentForPost(ByVal bcComment As BlogComment) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 1378) class as the post content. Return new post ID.

*See Also:* The Example section.

**C#**

```
public Long AddPost(Long blogID, ContentData PostContent, String Categories, Boolean Pingback,
String Tags, String TrackbackURL);
```

**Visual Basic**

```
Public Function AddPost(ByVal blogID As Long, ByVal PostContent As ContentData, ByVal
Categories() As String, ByVal Pingback As Boolean, ByVal Tags As String, ByVal TrackbackURL As
String) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| blogID | Numeric ID of the blog to which the post is added. |
| PostContent | ContentData (see page 1378) class representing the HTML, title, etc. |
| Categories | String Array representing the categories to which this post applies. |
| Pingback | Pingback URL(s) in this post? |
| Tags | String list of tags that applies to this post. |
| TrackbackURL | Trackback URL(s) to notify. |

**Returns**

Numeric ID of the post created. (Long)

**Example**

The following example shows how to create a Web page from which a user can add a blog post. A user enters the blog's ID, post title and content for the post. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddPost method. This method uses the InternalAdmin to add the blog post. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, create a table with three rows and three columns.
2. In the first row, first column, add the following label for the blog ID.
   ```
   <asp:Label ID="Label6" runat="server" Text="Blog ID:" Width="76px"></asp:Label>
   ```
3. In the first row, second column, add a text box to enter the blog's ID.
   ```
   <asp:TextBox ID="txtBlogId" runat="server" Width="55px"></asp:TextBox>
   ```
4. In the first row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that

something appears in the blog ID box.

```
<asp:RequiredFieldValidator ID="rfvBlogId" runat="server" ControlToValidate="txtBlogId"
  ErrorMessage="Please enter user name">Please enter blog ID</asp:RequiredFieldValidator>
```

5. In the second row, first column, add the following label for the area to post content.

```
<asp:Label ID="Label4" runat="server" Text="Post content:" Width="82px"></asp:Label>
```

6. In the second row, second column, add text box where the user can enter the blog post's content.

```
<asp:TextBox ID="txtPostContent" runat="server" Width="144px"></asp:TextBox>
```

7. In the second row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the blog post's content box.

```
<asp:RequiredFieldValidator ID="rfvPostContent" runat="server"
ControlToValidate="txtPostContent"
  ErrorMessage="Please enter first name">Please enter content to
post</asp:RequiredFieldValidator>
```

8. In the third row, first column, add the following label for the blog post's title.

```
<asp:Label ID="Label7" runat="server" Text="Post Title: " Width="76px"></asp:Label>
```

9. In the third row, second column, add text box where the user can enter the blog post's title.

```
<asp:TextBox ID="txtPostTitle" runat="server" Width="144px">Post Title</asp:TextBox>
```

10. In the third row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the blog post's title box.

```
<asp:RequiredFieldValidator ID="rfvPostTitle" runat="server"
ControlToValidate="txtPostTitle"
  ErrorMessage="Please enter first name">Please enter title for the content to
post</asp:RequiredFieldValidator>
```

11. After the table, add a break and then add a button to submit the blog post.

```
<br />
<asp:Button ID="btnAdd" runat="server" Text="Add Post" Width="70px" />
```

12. Add a some breaks for spacing purposes.

```
<br /><br />
```

13. Add a results label that lets a user know that the blog post has been added and the blog post's ID.

```
<asp:Label ID="lblResult" runat="server" Width="432px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnAdd.Click
```

2. Check to make sure the blog ID is numeric.

```
If Not IsNumeric(txtblogId.Text) Then
    lblResult.Text = "Invalid Blog ID"
    Exit Sub
End If
```

3. Create a Try/Catch around the AddPost method. This allows you to catch any exceptions and helps when debugging.

4. Create objects for the following items.

```
Try
   Dim blogApi As New Ektron.Cms.API.Content.Blog
   Dim postContentData As New Ektron.Cms.ContentData
   Dim categories() As String = New String() {}
```

5. Populate postContentData with the Title and the Content (see page 806) of the blog post.

```
   postContentData.Title = txtPostTitle.text
   postContentData.Html = txtPostContent.Text
```

6. Create a place holder for blog post's ID using the AddPost method.

```
   Dim postId As Integer
   postId = blogApi.AddPost(CInt(txtBlogId.Text), postContentData, categories, False,
"", "")
```

7. Let the user know the post ID and that it has been added.

```
   lblResult.Text = "A post was made to the blog with post ID " & postId
```

8. If there is an error, let the user know what happened.

```
            Catch ex As Exception
               lblResult.Text = ex.Message
            End Try

    End Sub
```

### Blog.DeleteBlog Method

Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0.

**C#**

```
public DeleteBlog(Long BlogID);
```

**Visual Basic**

```
Public Sub DeleteBlog(ByVal BlogID As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| BlogID | The numeric ID of the blog. This corresponds to the folder ID of the blog. |

### Blog.DeleteBlogPostComment Method

Delete Blog (see page 792) post comment by providing the comment's ID.

**C#**

```
public DeleteBlogPostComment(Long postCommentId);
```

**Visual Basic**

```
Public Sub DeleteBlogPostComment(ByVal postCommentId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| postCommentId | Post comment ID. |

### Blog.DeletePost Method

Deletes a post by the post's ID. This method will delete all comments associated with the post as well. Returns boolean value.

**C#**

```
public Boolean DeletePost(Long post_id);
```

**Visual Basic**

```
Public Function DeletePost(ByVal post_id As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| post_id | The numeric value of the post ID to delete. |

**Returns**

Boolean

### Blog.GetBlankBlogPostData Method

Returns a blank BlogPostData class. Used Primarily in the Workarea.

**C#**

```
public BlogPostData GetBlankBlogPostData();
```

**Visual Basic**

```
Public Function GetBlankBlogPostData() As BlogPostData
```

**Returns**

A blank, but not null, BlogPostData class.

## GetBlog Method

This is the overview for the GetBlog method overload.

**Description**

Returns a BlogData class for a given blog, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This method is overloaded. One GetBlog method allows you to use the Blog's ID. The other, allows you to use a Blog's path.

**Overload List**

| | Name | Description |
|---|---|---|
| | Blog.GetBlog (Long, String, Integer, Integer) (see page 797) | Returns a BlogData class for a given blog ID, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. |
| | Blog.GetBlog (String, String, Integer, Integer) (see page 800) | Returns a BlogData class for a given path, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. |

## Blog.GetBlog Method (Long, String, Integer, Integer)

Returns a BlogData class for a given blog ID, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

**C#**

```
public BlogData GetBlog(Long ID, String StartDate, int MaxEntries, int LanguageID);
```

**Visual Basic**

```
Public Function GetBlog(ByVal ID As Long, ByVal StartDate As String, ByVal MaxEntries As
Integer, ByVal LanguageID As Integer) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| ID | The ID of the blog. This is a numeric value that corresponds to the folder ID of the blog. |
| StartDate | The date to start getting entries. 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.<br>If set to 0, the `postsvisible` property of the blog is used.<br>If set to -1, the content is not returned, only the post info for the content. |
| LanguageID | The language ID for the content to get. An integer value. |

**Returns**

For a given blog ID, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

**Remarks**

Developers are advised to check for null prior to using.

**Example**

The following example shows how to create a Web page from which a user can retrieve blog information by specifying a blog ID, a date and how to handle post information. The sample returns BlogData for the Title, Tagline, Categories, Blog (see page 792) roll links and the number of posts. To create this example, we use some standard drag and drop controls and a small section of VB code utilizing the GetBlog method. This method uses the InternalAdmin to retrieve the blog information. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Create a table with three rows and two columns.
2. In the first row, first column, add a label that shows where to enter the blog's ID.
```
<asp:Label ID="Label1" runat="server" Text="Blog ID: "></asp:Label>
```
3. In the first row, second column, add a text box for the user to enter the blog ID and a RequiredFieldValidator to validate that something appears in the blog ID box.
```
<asp:TextBox ID="txtBlogId" runat="server" Width="63px">41</asp:TextBox>
<asp:RequiredFieldValidator ID="rfvBlogId" runat="server" ControlToValidate="txtBlogId"
    ErrorMessage="Please enter blog ID"></asp:RequiredFieldValidator>
```
4. In the second row, first column, add a label that shows where to enter the date.
```
<asp:Label ID="Label2" runat="server" Text="Date: "></asp:Label>
```
5. In the second row, second column, add a text box for the user to enter the date and a RequiredFieldValidator to validate that something appears in the date box.
```
<asp:TextBox ID="txtDate" runat="server" Width="63px"></asp:TextBox>
<asp:RequiredFieldValidator ID="rfvDate" runat="server" ControlToValidate="txtDate"
   ErrorMessage="Please enter the post date"></asp:RequiredFieldValidator>
```
6. In the third row, first column, add a label for the posts.
```
<asp:Label ID="Label4" runat="server" Text="Posts:"></asp:Label>
```
7. In the third row, second column, add the following radio buttons with breaks between each one.
```
<asp:RadioButton ID="rb_forday" runat="server" Checked="True" GroupName="rb_num" Text="All
posts for that day only" /><br />
<asp:RadioButton ID="rb_dayprior" runat="server" GroupName="rb_num" Text="All posts for
that day and before" /><br />
<asp:RadioButton ID="rb_month" runat="server" GroupName="rb_num" Text="All posts for the
month that day falls on." /><br />
<asp:RadioButton ID="rb_Custom" runat="server" GroupName="rb_num" Text="Starting from the
date, this number of posts:" />
```
8. Add a text box at the end of the last radio button so a user can enter the quantity of posts to return.
```
<asp:TextBox ID="txt_num" runat="server" Width="63px">1</asp:TextBox>
```
9. After the table, add a break.
```
<br />
```
10. Add a button to retrieve the blog information.
```
<asp:Button ID="btnGetBlog" runat="server" Text="Get Blog" Width="72px" />
```
11. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
12. Add a results label that displays the blog information. It will also display a message if there is an error.
```
<asp:Label ID="lblResult" runat="server" Width="604px" Height="113px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
    Protected Sub btnGetBlog_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetBlog.Click
```

2. Create the following objects.

```
    Dim iPostNum As Integer = -1
    Dim blogDataItem As New Ektron.Cms.BlogData
    Dim blogApi As New Ektron.Cms.API.Content.Blog
    Dim i As Integer
```

3. Check to make sure txtBlogId is numeric.

```
If Not IsNumeric(txtBlogId.Text) Then
    lblResult.Text = "Invalid Blog ID"
    Exit Sub
End If
```

4. Check to make sure txtDate is a date.

```
If Not IsDate(txtDate.Text) Then
    lblResult.Text = "Invalid Date"
    Exit Sub
End If
```

5. Create a Try/Catch.

6. Add a check to see which radio button is active and set the number of posts.

```
Try
   If Me.rb_forday.Checked = True Then
       iPostNum = -1
   ElseIf Me.rb_month.Checked = True Then
       iPostNum = -2
   ElseIf Me.rb_dayprior.Checked = True Then
       iPostNum = 0
   ElseIf Me.rb_Custom.Checked = True Then
       If IsNumeric(Me.txt_num.Text) AndAlso CInt(Me.txt_num.Text) > 0 Then
           iPostNum = Convert.ToInt32(Me.txt_num.Text)
       Else
           Me.txt_num.Text = "1"
           iPostNum = 1
       End If
   End If
```

7. Get the blog.

```
blogDataItem = blogApi.GetBlog(CInt(txtBlogId.Text), txtDate.Text, iPostNum, 1033)
```

**Note:** To get the blog using the path instead of the ID use the following code. `blogDataItem = blogApi.GetBlog("~SiteRoot~\TestBlog", txtDate.Text, -1, 1033)`

8. Check to see if the blog exists.

```
If (blogDataItem.Id = 0) Then
    lblResult.Text = "Error getting blog"
    Exit Sub
End If
```

9. Display the results for the user.

```
lblResult.Text = "Title: " & blogDataItem.Title & "<br/>"
lblResult.Text &= "Tagline: " & blogDataItem.Tagline & "<br/>"
If blogDataItem.Categories.Length <> 0 Then
    lblResult.Text &= "Categories: "
    For i = 0 To blogDataItem.Categories.Length - 1 Step 1
        lblResult.Text &= blogDataItem.Categories(i) & " "
    Next
    lblResult.Text &= "<br>"
End If

If blogDataItem.BlogRoll.Items.Length <> 0 Then
    lblResult.Text &= "Blog roll links: "
    For i = 0 To blogDataItem.BlogRoll.Items.Length - 1 Step 1
        lblResult.Text &= "<a href=""" & blogDataItem.BlogRoll.Items(i).URL & """
target=""mainFrame"">" & blogDataItem.BlogRoll.Items(i).LinkName & "</a>" & "  "
    Next
    lblResult.Text &= "<br>"
End If
lblResult.Text &= "Number of posts: " & blogDataItem.Content.Length & "<br/>"
```

10. If there is an error, let the user know what happened.

```
Catch ex As Exception
```

```
            lblResult.Text = "Error getting blog"
        End Try
```

### Blog.GetBlog Method (String, String, Integer, Integer)

Returns a BlogData class for a given path, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

**C#**

```
public BlogData GetBlog(String path, String StartDate, int MaxEntries, int LanguageID);
```

**Visual Basic**

```
Public Function GetBlog(ByVal path As String, ByVal StartDate As String, ByVal MaxEntries As
Integer, ByVal LanguageID As Integer) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| path | The path of the blog. This is a string value that corresponds to the folder path of the blog. |
| StartDate | The date to start getting entries. 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.<br>If set to 0, the `postsvisible` property of the blog is used.<br>If set to -1, the content is not returned, only the post info for the content. |
| LanguageID | The language ID for the content. An integer value. |

**Returns**

For a given path, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

**Remarks**

Developers are advised to check for null prior to using.

### GetBlogData Method

This is the overview for the GetBlogData method overload.

**Description**

Overloaded Method to obtain a BlogData Class.

**Overload List**

| | Name | Description |
|---|---|---|
| | Blog.GetBlogData (FolderData) (see page 801) | 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.**<br>This function is overloaded. See Also:<br>• BlogObject(Long DateTime)<br>• BlogObject(Long DateTime Integer Integer)<br>• BlogObject(Long) |

| | | |
|---|---|---|
| | Blog.GetBlogData (Long) (see page 802) | Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog. Returns either a populated BlogData class or null (nothing).<br>This function is overloaded. See Also:<br>• BlogObject(Long DateTime)<br>• BlogObject(Long DateTime Integer Integer)<br>• BlogObject(FolderData) |
| | Blog.GetBlogData (Long, DateTime) (see page 802) | Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing).<br>This function is overloaded. See Also:<br>• BlogObject(Long)<br>• BlogObject(Long DateTime Integer Integer)<br>• BlogObject(FolderData) |
| | Blog.GetBlogData (Long, DateTime, Integer, Integer) (see page 803) | Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing).<br>This function is overloaded. See Also:<br>• BlogObject(Long)<br>• BlogObject(Long DateTime)<br>• BlogObject(FolderData) |

### Blog.GetBlogData Method (FolderData)

Returns the corresponding blog (if applicable) for a complete folderData object. If there is code that already leverages our FolderData class, this can piggyback atop it. **Returns only if the folder is a blog.**

This function is overloaded. See Also:

- BlogObject(Long DateTime)
- BlogObject(Long DateTime Integer Integer)
- BlogObject(Long)

**C#**

```
public BlogData GetBlogData(FolderData folder_data);
```

**Visual Basic**

```
Public Function GetBlogData(ByVal folder_data As FolderData) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| folder_data | A populated FolderData class. (see FolderData) |

**Returns**

A populated BlogData class or nothing

**Remarks**

## Blog.GetBlogData Method (Long)

Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

- BlogObject(Long DateTime)
- BlogObject(Long DateTime Integer Integer)
- BlogObject(FolderData)

**C#**

```
public BlogData GetBlogData(Long BlogID);
```

**Visual Basic**

```
Public Function GetBlogData(ByVal BlogID As Long) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| BlogID | Numeric ID of the blog. This corresponds to the ID of the blog folder. |

**Returns**

Either a populated BlogData class or null (nothing).

**Remarks**

## Blog.GetBlogData Method (Long, DateTime)

Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

- BlogObject(Long)
- BlogObject(Long DateTime Integer Integer)
- BlogObject(FolderData)

**C#**

```
public BlogData GetBlogData(Long BlogID, DateTime StartDate);
```

**Visual Basic**

```
Public Function GetBlogData(ByVal BlogID As Long, ByVal StartDate As DateTime) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| BlogID | Numeric ID of the blog. This corresponds to the ID of the blog folder. |
| StartDate | The DateTime value from which to get posts. You can obtain historical posts with this property, or leave it blank to begin now. |

**Returns**

Either a populated BlogData class or null (nothing).

**Remarks**

### Blog.GetBlogData Method (Long, DateTime, Integer, Integer)

Overloaded Method to obtain a BlogData Class. Accepts the ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

- BlogObject(Long)
- BlogObject(Long DateTime)
- BlogObject(FolderData)

**C#**

```
public BlogData GetBlogData(Long BlogID, DateTime StartDate, int Postsvisible, int LanguageID);
```

**Visual Basic**

```
Public Function GetBlogData(ByVal BlogID As Long, ByVal StartDate As DateTime, ByVal
Postsvisible As Integer, ByVal LanguageID As Integer) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| BlogID | Numeric ID of the blog. This corresponds to the ID of the blog folder. |
| StartDate | The DateTime value from which to get posts. 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#**

```
public BlogData GetBlogDataForPost(Long post_id);
```

**Visual Basic**

```
Public Function GetBlogDataForPost(ByVal post_id As Long) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| post_id | Numeric value of the post ID |

**Returns**

BlogData class

### Blog.GetBlogPostData Method

Gets the Post data for a specific blog post. Accepts the ID of the post. Will return tags, trackback, pingback, and categories for the post.

**C#**

```
public BlogPostData GetBlogPostData(Long id);
```

**Visual Basic**

```
Public Function GetBlogPostData(ByVal id As Long) As BlogPostData
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Numeric value of the post ID. |

**Returns**

BlogPostData Class.

## GetBlogRoll Method

This is the overview for the GetBlogRoll method overload.

**Description**

Returns a populated blog roll class. This blog roll is the list of links for the blog. See the BlogRoll class for the properties and usage of the blog roll.

This is an overloaded method.

**Overload List**

| | Name | Description |
|---|---|---|
| | Blog.GetBlogRoll (Long) (see page 804) | Returns a populated blog roll class. Accepts a blog ID value. See the BlogRoll (see page 52) class for more information. |
| | Blog.GetBlogRoll (String) (see page 805) | 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 52) class for the properties and usage of the blog roll.<br>This is an overloaded method. See Also: GetBlogRoll(Long). |

## Blog.GetBlogRoll Method (Long)

Returns a populated blog roll class. Accepts a blog ID value. See the BlogRoll (see page 52) class for more information.

**C#**

```
public BlogRoll GetBlogRoll(Long blog_id);
```

**Visual Basic**

```
Public Function GetBlogRoll(ByVal blog_id As Long) As BlogRoll
```

**Parameters**

| Parameters | Description |
|---|---|
| blog_id | Numeric ID of the blog. This corresponds to the folder ID of the blog. |

**Returns**

A populated BlogRoll (see page 52) 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 52) class for the properties and usage of the blog roll.

This is an overloaded method. See Also: GetBlogRoll(Long).

**C#**

```
public BlogRoll GetBlogRoll(String Path);
```

**Visual Basic**

```
Public Function GetBlogRoll(ByVal Path As String) As BlogRoll
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 52) 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 (see page 1378) Class.

**C#**

```
public ContentData GetPost(Long post_id, ref BlogPostData PostInfo);
```

**Visual Basic**

```
Public Function GetPost(ByVal post_id As Long, ByRef PostInfo As BlogPostData) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| post_id | Numeric ID for the post. |
| PostInfo | BlogPostData class representing the blog specific post info. This is also in the metadata. |

**Returns**

ContentData (see page 1378) class with a ByRef BlogPostData

### Blog.GetPostbyID Method

Gets a blog post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class.

**C#**

```
public BlogPostData GetPostbyID(Long post_id);
```

**Visual Basic**

```
Public Function GetPostbyID(ByVal post_id As Long) As BlogPostData
```

**Parameters**

| Parameters | Description |
|---|---|
| post_id | Numeric ID for the post. |

**Returns**

BlogPostData class

### Blog.GetUserBlog Method

gets the user blog id

**C#**

```
public Long GetUserBlog(Long UserId);
```

**Visual Basic**

```
Public Function GetUserBlog(ByVal UserId As Long) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| UserId | userid |

### Blog.UpdateBlogProperties Method

Updates the properties of a blog. Accepts standard blogdata object. Returns boolean.

**C#**

```
public Boolean UpdateBlogProperties(BlogData blog);
```

**Visual Basic**

```
Public Function UpdateBlogProperties(ByVal blog As BlogData) As Boolean
```

**Parameters**

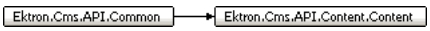
| Parameters | Description |
|---|---|
| blog | Information for the blog. Posts will be ignored, only blog properties will be updated. |

**Returns**

Boolean value

## Content Class

**Class Hierarchy**




**C#**

```
public class Content : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class Content
Inherits Ektron.Cms.API.Common
```

**File**

Content.vb

**Description**

This is class Ektron.Cms.API.Content.Content.

### Content Methods

The methods of the Content class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddAsset (see page 808) | Adds an asset to the CMS. |
| | AddContent (see page 808) | This is the overview for the AddContent method overload. |
| | AddXmlConfiguration (see page 813) | Adds an XML configuration for XML oriented content. |
| | CheckOutContent (see page 813) | 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 814) | Copy the content into required folder. |
| | DeleteContentItem (see page 814) | Delete the content by the content's ID.<br>*See Also:* The Example section. |
| | DeleteXmlConfiguration (see page 815) | Deletes an XML configuration. |
| | EditAsset (see page 816) | Edits an asset that was uploaded into the CMS. |
| | GetAddViewLanguage (see page 816) | 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 816) | Loads all of the comments. |
| | GetAllTemplates (see page 817) | Loads all of the templates. |
| | GetAllUnassignedItemApprovals (see page 817) | Returns all of the unassigned user/groups for a given content ID. |
| | GetAllXmlConfigurations (see page 817) | Reads all of the XML configurations. |
| | GetChildContent (see page 818) | This is the overview for the GetChildContent method overload. |
| | GetCommunityTemplate (see page 820) | Loads template for the given community type. |
| | GetContent (see page 820) | This is the overview for the GetContent method overload. |
| | GetContentByHistory (see page 823) | Retrieves the corresponding content by the history ID. |
| | GetContentDisplayLink (see page 823) | 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 823) | Loads the content details with checkout mode. |
| | GetContentReport (see page 824) | This is the overview for the GetContentReport method overload. |
| | GetContentState (see page 825) | Loads the content's details. |
| | GetContentStatus (see page 825) | 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 825) | Gets web domain for a piece of content. |
| | GetExpireContent (see page 826) | Loads all of the expired content. |
| | GetList (see page 826) | This is the overview for the GetList method overload. |
| | MoveContentToFolder (see page 827) | Moves content to a folder. |
| | PublishContent (see page 827) | Function takes in information about the content to be checked in and published. This goes through the normal approval chain. The content must be in a checked out state to the current logged in user for this to succeed. |
| | RelicateXmlConfiguration (see page 828) | **This method is obsolete. The recommended alternative is ReplicateXmlConfiguration (see page 828).**<br>Replicate the XML configuration. |
| | RenameContent (see page 828) | Renames the content. |
| | ReplicateXmlConfiguration (see page 828) | Replicate the XML configuration. |
| | RestoreHistoryContent (see page 829) | Restores the content to a particular version of content. |

| | | |
|---|---|---|
| | SaveContent (see page 829) | Updates the checked out content block. Only following properties can be updated:<br>• contentData.Title<br>• contentData.Teaser<br>• contentData.Comment<br>• contentData.GoLive<br>• contentData.Html<br>• contentData.EndDate<br>• contentData.EndDateAction<br>• contentData.ManualAlias |
| | SubmitForDelete (see page 830) | Delete the content by the content's Id and folder Id. |
| | UndoCheckout (see page 830) | This undoes the checkout for a piece of content which is checked out to the current user. |
| | UpdateContentMetaData (see page 830) | Updates a single piece of content metadata |
| | UpdatexmlCollectionPackage (see page 831) | Updates the XML collection package. |
| | UpdateXmlConfiguration (see page 831) | Updates the XML configuration details. |

## Content.AddAsset Method

Adds an asset to the CMS.

### C#

```
public Long AddAsset(System.IO.Stream inputStream, AssetUpdateData assetdata);
```

### Visual Basic

```
Public Function AddAsset(ByVal inputStream As System.IO.Stream, ByVal assetdata As
AssetUpdateData) As Long
```

### Parameters

| Parameters | Description |
|---|---|
| inputStream | Input Stream pointing to the file to be uploaded |
| assetdata | Asset (see page 789) details in AssetUpdateData |

### Returns

Content (see page 806) Id

## AddContent Method

This is the overview for the AddContent method overload.

### Description

Adds Content to the CMS. This method is overloaded with three AddContent methods.

The first method allows you to add content and have the CMS assign it a content ID. The second, allows you to add content and assign it a content ID. The third, allows you to add content and assign it an XML ID and Template ID, while the CMS assigns it a content ID

**Overload List**

| | Name | Description |
|---|---|---|
| | Content.AddContent (String, String, String, String, String, String, Long, Object, Object, Object) (see page 809) | Adds Content (see page 806) to the CMS. Upon successful completion, the CMS returns a Content (see page 806) ID. When content is added using this method, Web Alerts are **not** sent.<br>*See Also:* The Example section. |
| | Content.AddContent (String, String, String, String, String, String, Long, Object, Object, Object, Long) (see page 811) | Adds Content (see page 806) to the CMS and allows you to assign it a Content (see page 806) ID. Once the content is added the new ID is returned. When content is added using this method, Web Alerts are **not** sent. |
| | Content.AddContent (String, String, String, String, String, String, Long, Object, Object, Object, Long, Long) (see page 812) | Adds Content (see page 806) to the CMS and allows you to assign the content an XML ID and Template ID. Once the content is added the new Content (see page 806) ID is returned. When content is added using this method, Web Alerts are **not** sent. |
| | Content.AddContent (String, String, String, String, String, String, Long, Object, Object, Object, Long, Long, Boolean) (see page 812) | Adds Content (see page 806) to the CMS and allows you to assign the content an XML ID and Template ID. Once the content is added the new Content (see page 806) ID is returned. When content is added using this method, Web Alerts are **not** sent. |

## Content.AddContent Method (String, String, String, String, String, String, Long, Object, Object, Object)

Adds Content (see page 806) to the CMS. Upon successful completion, the CMS returns a Content (see page 806) ID. When content is added using this method, Web Alerts are **not** sent.

*See Also:* The Example section.

**C#**

```
public Long AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, Long FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml);
```

**Visual Basic**

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Long, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentTitle | Title of content to add. |
| ContentComment | Comments about the content being added. |
| ContentHtml | Content (see page 806) HTML to display. |
| SearchText | Search (see page 864) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank. |
| SummaryHtml | A summary of the HTML content. |
| ContentLanguage | The language to which the content is assigned. |

| | |
|---|---|
| FolderId | Folder (see page 652) in which to place the content. |
| GoLive | The date the content goes live on the site. |
| EndDate | The date the content stops appearing on the site. |
| MetaInfoXml | Meta Info XML |

**Returns**

Long - ID of newly created content.

**Example**

The following example shows how to create a Web page from which a user can add content to the CMS. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddContent method. This method uses the InternalAdmin to add the content. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
```
<asp:Label ID="Label1" runat="server" Text="Content Title: "></asp:Label>
```
2. Add a text box with the following information.
```
<asp:TextBox ID="txtTitle" runat="server"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Title Box.
```
<asp:RequiredFieldValidator ID="rfvTitle" runat="server" ControlToValidate="txtTitle"
    ErrorMessage="Please enter content title"></asp:RequiredFieldValidator>
```
4. Add a couple of breaks for spacing purposes.
```
<br /><br />
```
5. Add another label for the content text box.
```
<asp:Label ID="Label2" runat="server" Text="Content Text: " Width="87px"></asp:Label>
```
6. Add the text box where the content is entered. This is a MultiLine text box.
```
<asp:TextBox ID="txtContent" runat="server" Height="105px" TextMode="MultiLine"
Width="336px"></asp:TextBox>
```
7. Add a some more breaks for spacing purposes.
```
<br /><br />
```
8. Add a label for to show where to enter the Folder ID.
```
<asp:Label ID="Label3" runat="server" Text="Folder Id: " Width="65px"></asp:Label>
```
9. Add one last text box for the user to enter the Folder ID where the content will be stored.
   By default the text box is set to 0 (zero) which is the root folder.
```
<asp:TextBox ID="txtFolderId" runat="server" Width="44px">0</asp:TextBox>
```
10. Add another RequiredFieldValidator control. This validates that information is entered into the text box.
    Note: This does not validate that the folder actually exists in the CMS.
```
<asp:RequiredFieldValidator ID="rfvFolderId" runat="server" ControlToValidate="txtFolderId"
   ErrorMessage="Please enter the Folder Id"></asp:RequiredFieldValidator>
```
11. Add a break.
```
<br />
```
12. Add a label to describe the type of information that needs to be add to the folder ID box.
```
<asp:Label ID="Label4" runat="server" Font-Italic="True" Text="Enter a valid Folder ID from
the workarea, 0 points to the root folder."
   Width="483px"></asp:Label>
```
13. Add two more breaks.
```
<br /><br />
```
14. Add a button. We will set the button click event in the code behind.
```
<asp:Button ID="btnAdd" runat="server" Text="Add" />
```
15. Add two last breaks.
```
<br /><br />
```
16. Add a results label that lets a user know if the add succeeded.
```
<asp:Label ID="lblResult" runat="server"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
```

```
btnAdd.Click
```

2. Create an object for the content API (see page 639).

```
Dim contentApi As New Ektron.Cms.API.Content.Content
```

3. Check to make sure the folder ID is numeric.

```
If Not IsNumeric(txtFolderId.Text) Then
    lblResult.Text = "Invalid folder ID"
    Exit Sub
End If
```

4. Create an starting content ID of 0 (zero). When the content is added, it will be assigned an ID.

```
Dim ID As Integer = 0
```

5. Create a Try/Catch around the AddContent method. This allows you to catch any exceptions and helps when debugging.

```
Try
    ID = contentApi.AddContent(txtTitle.Text, "Content added by Example.aspx",
txtContent.Text, "", _
"", "1033", CInt(txtFolderId.Text), "", "", Nothing)
```

6. Add an If statement that lets the user know whether the content was added and the ID of the content.

```
    If (ID <> 0) Then
        lblResult.Text = "The above content was added into CMS400.NET with
ContentID = " & ID
    End If
```

7. If there is an error, let the user know what happened.

```
Catch ex As Exception
     lblResult.Text = ex.Message
End Try

End Sub
```

8. Add a page load event so that lblResult.Text is blank when the page first loads.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
   lblResult.Text = ""
End Sub
```

### Content.AddContent Method (String, String, String, String, String, String, Long, Object, Object, Object, Long)

Adds Content (see page 806) to the CMS and allows you to assign it a Content (see page 806) ID. Once the content is added the new ID is returned. When content is added using this method, Web Alerts are **not** sent.

#### C#

```
public Long AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, Long FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, Long ContentID);
```

#### Visual Basic

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Long, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object, ByVal ContentID As Long) As Long
```

#### Parameters

| Parameters | Description |
|---|---|
| ContentTitle | Title of content to add. |
| ContentComment | Comments about the content being added. |
| ContentHtml | Content (see page 806) HTML to display. |
| SearchText | Search (see page 864) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank. |
| SummaryHtml | A summary of the HTML content. |

| | |
|---|---|
| ContentLanguage | The language to which the content is assigned. |
| FolderId | Folder (see page 652) in which to place the content. |
| GoLive | The date the content goes live on the site. |
| EndDate | The date the content stops appearing on the site. |
| MetaInfoXml | Meta Info XML |
| ContentID | The ID you would like to assign the content. |

**Returns**

Long - ID of newly created content.

## Content.AddContent Method (String, String, String, String, String, String, Long, Object, Object, Object, Long, Long)

Adds Content (see page 806) to the CMS and allows you to assign the content an XML ID and Template ID. Once the content is added the new Content (see page 806) ID is returned. When content is added using this method, Web Alerts are **not** sent.

**C#**

```
public Long AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, Long FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, Long XmlID, Long TemplateID);
```

**Visual Basic**

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Long, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object, ByVal XmlID As Long, ByVal TemplateID As Long) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentTitle | Title of content to add. |
| ContentComment | Comments about the content being added. |
| ContentHtml | Content (see page 806) HTML to display. |
| SearchText | Search (see page 864) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank. |
| SummaryHtml | A summary of the HTML content. |
| ContentLanguage | The language to which the content is assigned. |
| FolderId | Folder (see page 652) in which to place the content. |
| GoLive | The date the content goes live on the site. |
| EndDate | The date the content stops appearing on the site. |
| MetaInfoXml | Meta Info XML |
| XmlID | XmlID |
| TemplateID | Template ID |

**Returns**

Long - ID of newly created content.

## Content.AddContent Method (String, String, String, String, String, String, Long, Object, Object, Object, Long, Long, Boolean)

Adds Content (see page 806) to the CMS and allows you to assign the content an XML ID and Template ID. Once the content is added the new Content (see page 806) ID is returned. When content is added using this method, Web Alerts are **not** sent.

**C#**

```
public Long AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, Long FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, Long XmlID, Long TemplateID, Boolean CleanHtml);
```

**Visual Basic**

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal ContentLanguage As String, ByVal FolderId As Long, ByVal GoLive As Object, ByVal EndDate As Object, ByVal MetaInfoXml As Object, ByVal XmlID As Long, ByVal TemplateID As Long, ByVal CleanHtml As Boolean) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentTitle | Title of content to add. |
| ContentComment | Comments about the content being added. |
| ContentHtml | Content (see page 806) HTML to display. |
| SearchText | Search (see page 864) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank. |
| SummaryHtml | A summary of the HTML content. |
| ContentLanguage | The language to which the content is assigned. |
| FolderId | Folder (see page 652) in which to place the content. |
| GoLive | The date the content goes live on the site. |
| EndDate | The date the content stops appearing on the site. |
| MetaInfoXml | Meta Info XML |
| XmlID | XmlID |
| TemplateID | Template ID |
| CleanHtml | Clean ContentHTML and SummaryHTML with tidy. |

**Returns**

Long - ID of newly created content.

## Content.AddXmlConfiguration Method

Adds an XML configuration for XML oriented content.

**C#**

```
public Long AddXmlConfiguration(Collection pagedata);
```

**Visual Basic**

```
Public Function AddXmlConfiguration(ByVal pagedata As Collection) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | XML configuration data |

**Returns**

A numeric value (Long).

## Content.CheckOutContent Method

This will check the content out to the current user if the content is currently checked in and the user has privileges to check it out.

**C#**

```
public Boolean CheckOutContent(Long ContentId);
```

**Visual Basic**

```
Public Function CheckOutContent(ByVal ContentId As Long) As Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| ContentId | Id of the content to check out |

### Returns

A Boolean representing the success of the check out

## Content.CopyContentToFolder Method

Copy the content into required folder.

### 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 806) ID |
| FID | Folder (see page 652) ID |
| Lang | |

### Returns

Ektron.Cms.ContentData (see page 1378)

## Content.DeleteContentItem Method

Delete the content by the content's ID.

*See Also:* The Example section.

### C#

```
public Boolean DeleteContentItem(Long Id);
```

### Visual Basic

```
Public Function DeleteContentItem(ByVal Id As Long) As Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| Id | The content's ID. |

### Returns

Boolean - returns false if the delete succeeds.

### Example

The following example shows how to create a Web page from which a user can delete content by entering its ID into a text box. This example uses some standard drag and drop controls and a small section of VB code utilizing the DeleteContentItem method. This method uses the InternalAdmin to delete the content. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
   ```
   <asp:Label ID="Label1" runat="server" Text="Content ID: "></asp:Label>
   ```
2. Add a text box for the user to enter the content ID.
   ```
   <asp:TextBox ID="txtContentId" runat="server" Width="44px"></asp:TextBox>
   ```

3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Content ID box.

```
<asp:RequiredFieldValidator ID="rfvContentId" runat="server"
ControlToValidate="txtContentId"
    ErrorMessage="Please enter content ID"></asp:RequiredFieldValidator>
```

4. Add a couple of breaks for spacing purposes.

```
<br /><br />
```

5. Add a button to delete the content. We will set the button click event in the code behind.

```
<asp:Button ID="btnDelete" runat="server" Text="Delete" />
```

6. Add a couple of more breaks for spacing purposes.

```
<br /><br />
```

7. Add a results label that lets a user know if the add succeeded.

```
<asp:Label ID="lblResult" runat="server"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
Protected Sub btnDelete_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnDelete.Click
```

2. Create an object for the content API (see page 639).

```
    Dim contentApi As New Ektron.Cms.API.Content.Content
```

3. Check to make sure the content ID is numeric.

```
        If Not IsNumeric(txtContentId.Text) Then
            lblResult.Text = "Invalid content ID"
            Exit Sub
        End If
```

4. Create an item *deleted* and set it to true. Note that DeleteContentItem returns false if the delete succeeds.

```
    Dim deleted As Boolean = True
```

5. Create a Try/Catch around the DeleteContentItem method. This allows you to catch any exceptions and helps when debugging.

```
    Try
        deleted = contentApi.DeleteContentItem(CInt(txtContentId.Text))
```

6. Add an If statement that lets the user know whether the content was deleted.

```
        If Not deleted Then
            lblResult.Text = "The above content was deleted from CMS400.Net"
        End If
```

7. If there is an error, let the user know what happened.

```
    Catch ex As Exception
        lblResult.Text = ex.Message
    End Try

End Sub
```

### Content.DeleteXmlConfiguration Method

Deletes an XML configuration.

**C#**

```
public DeleteXmlConfiguration(Long Id);
```

**Visual Basic**

```
Public Sub DeleteXmlConfiguration(ByVal Id As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | XML configuration ID |

## Content.EditAsset Method

Edits an asset that was uploaded into the CMS.

**C#**

```
public Boolean EditAsset(System.IO.Stream inputStream, AssetUpdateData assetdata);
```

**Visual Basic**

```
Public Function EditAsset(ByVal inputStream As System.IO.Stream, ByVal assetdata As
AssetUpdateData) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| inputStream | Input Stream pointing to the file to be uploaded if file is being overwritten |
| assetdata | Asset (see page 789) details in AssetUpdateData |

**Returns**

A Boolean indicating success, False for success

## Content.GetAddViewLanguage Method

Loads all the languages from the system for a given content ID and populates the Add/View drop down box with flag set on each item.

**C#**

```
public LanguageData[] GetAddViewLanguage(Long Id);
```

**Visual Basic**

```
Public Function GetAddViewLanguage(ByVal Id As Long) As LanguageData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | Content (see page 806) ID |

**Returns**

Ektron.Cms.LanguageData()

## Content.GetAllComments Method

Loads all of the comments.

**C#**

```
public CommentData[] GetAllComments(Long KeyId, Long CommentId, Long RefId, String RefType,
Long UserId, String OrderBy);
```

**Visual Basic**

```
Public Function GetAllComments(ByVal KeyId As Long, ByVal CommentId As Long, ByVal RefId As
Long, ByVal RefType As String, ByVal UserId As Long, ByVal OrderBy As String) As CommentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| KeyId | key ID |

| | |
|---|---|
| CommentId | Comment ID |
| RefId | Reference ID, (content or forms) |
| RefType | Reference type, C = content or T = Task (see page 871) |
| UserId | The user ID |
| OrderBy | The order of the comments |

**Returns**

Ektron.Cms.CommentData()

### Content.GetAllTemplates Method

Loads all of the templates.

**C#**

```
public TemplateData[] GetAllTemplates(String OrderBy);
```

**Visual Basic**

```
Public Function GetAllTemplates(ByVal OrderBy As String) As TemplateData()
```

**Parameters**

| Parameters | Description |
|---|---|
| OrderBy | Order by clause |

**Returns**

Ektron.Cms.TemplateData()

### Content.GetAllUnassignedItemApprovals Method

Returns all of the unassigned user/groups for a given content ID.

**C#**

```
public ApprovalData[] GetAllUnassignedItemApprovals(Long Id, String ItemType);
```

**Visual Basic**

```
Public Function GetAllUnassignedItemApprovals(ByVal Id As Long, ByVal ItemType As String) As
ApprovalData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The content'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()

## GetChildContent Method

This is the overview for the GetChildContent method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Content.GetChildContent (Long) (see page 818) | Loads all of the content for the given folder. |
| | Content.GetChildContent (Long, Boolean) (see page 818) | Loads all of the content for the given folder. |
| | Content.GetChildContent (Long, Boolean, String) (see page 819) | Loads all of the content for the given folder. |
| | Content.GetChildContent (Long, Boolean, String, PagingInfo) (see page 819) | Loads all of the content for the given folder. |

## Content.GetChildContent Method (Long)

Loads all of the content for the given folder.

**C#**

```
public ContentData[] GetChildContent(Long Id);
```

**Visual Basic**

```
Public Function GetChildContent(ByVal Id As Long) As ContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The folder's ID. |

**Returns**

Ektron.Cms.ContentData (see page 1378)()

## Content.GetChildContent Method (Long, Boolean)

Loads all of the content for the given folder.

**C#**

```
public ContentData[] GetChildContent(Long Id, Boolean Recursive);
```

**Visual Basic**

```
Public Function GetChildContent(ByVal Id As Long, ByVal Recursive As Boolean) As ContentData()
```

### Parameters

| Parameters | Description |
|---|---|
| Id | The folder's ID. |
| Recursive | True = recursive look up. False = none. Note: This parameter is not functional. Please use Content.GetList (see page 826)() api for recursive support. |

### Returns

Ektron.Cms.ContentData (see page 1378)()

## Content.GetChildContent Method (Long, Boolean, String)

Loads all of the content for the given folder.

### C#

```
public ContentData[] GetChildContent(Long Id, Boolean Recursive, String OrderBy);
```

### Visual Basic

```
Public Function GetChildContent(ByVal Id As Long, ByVal Recursive As Boolean, ByVal OrderBy As
String) As ContentData()
```

### Parameters

| Parameters | Description |
|---|---|
| Id | The folder's ID. |
| Recursive | True = recursive look up. False = none. Note: This parameter is not functional. Please use Content.GetList (see page 826)() api for recursive support. |
| OrderBy | An order by clause. Choices are:<br>• **id** - the content ID.<br>• **title** - the title of the content item.<br>• **status** - whether the content is checked-in, published, checked-out, etc.<br>• **language** - the language ID of the content.<br>• **datecreated** - the date the content was created.<br>• **datemodified** - the date the content was last modified.<br>• **userid** - the ID of the user who last worked with the content. |

### Returns

Ektron.Cms.ContentData (see page 1378)()

## Content.GetChildContent Method (Long, Boolean, String, PagingInfo)

Loads all of the content for the given folder.

### C#

```
public ContentData[] GetChildContent(Long Id, Boolean Recursive, String OrderBy, PagingInfo
pagingInfo);
```

### Visual Basic

```
Public Function GetChildContent(ByVal Id As Long, ByVal Recursive As Boolean, ByVal OrderBy As
String, ByVal pagingInfo As PagingInfo) As ContentData()
```

### Parameters

| Parameters | Description |
|---|---|
| Id | The folder's ID. |
| Recursive | True = recursive look up. False = none. |

| | |
|---|---|
| OrderBy | An order by clause. Choices are:<br>• **id** - the content ID.<br>• **title** - the title of the content item.<br>• **status** - whether the content is checked-in, published, checked-out, etc.<br>• **language** - the language ID of the content.<br>• **datecreated** - the date the content was created.<br>• **datemodified** - the date the content was last modified.<br>• **userid** - the ID of the user who last worked with the content. |
| pagingInfo | paging Information for the retrieval |

**Returns**

Ektron.Cms.ContentData (see page 1378)()

## Content.GetCommunityTemplate Method

Loads template for the given community type.

**C#**

```
public TemplateData[] GetCommunityTemplate(Ektron.Cms.Common.EkEnumeration.TemplateType type);
```

**Visual Basic**

```
Public Function GetCommunityTemplate(ByVal type As
Ektron.Cms.Common.EkEnumeration.TemplateType) As TemplateData()
```

**Parameters**

| Parameters | Description |
|---|---|
| type | type of template |

**Returns**

Ektron.Cms.TemplateData()

## GetContent Method

This is the overview for the GetContent method overload.

**Description**

Returns list of content. This function is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | Content.GetContent (Ektron.Cms.Common.ContentRequest, System.Web.UI.Page) (see page 821) | Returns list of contents using content request object. Only returns content of type CMSContentType.Content (see page 806) and CMSContentType.Assets. |
| | Content.GetContent (Long, ContentResultType) (see page 821) | Using the content's ID and the ContentType to get the content and return it as ContentData (see page 1378). Only returns content of type CMSContentType.Content (see page 806) and CMSContentType.Assets.<br>*See Also:* The Example section. |
| | Content.GetContent (Long, Ektron.Cms.Content.EkContent.ContentResultType) (see page 822) | Loads the content details using the content ID. Only returns content of type CMSContentType.Content (see page 806) and CMSContentType.Assets. |

## Content.GetContent Method (Ektron.Cms.Common.ContentRequest, System.Web.UI.Page)

Returns list of contents using content request object. Only returns content of type CMSContentType.Content (see page 806) and CMSContentType.Assets.

**C#**

```
public Ektron.Cms.Common.ContentResult GetContent(Ektron.Cms.Common.ContentRequest req,
Optional ByVal System.Web.UI.Page page);
```

**Visual Basic**

```
Public Function GetContent(ByVal req As Ektron.Cms.Common.ContentRequest, Optional ByVal page
As System.Web.UI.Page = Nothing) As Ektron.Cms.Common.ContentResult
```

**Parameters**

| Parameters | Description |
|---|---|
| req | ContentRequest |
| page | Page reference |

**Returns**

ContentResult

## Content.GetContent Method (Long, ContentResultType)

Using the content's ID and the ContentType to get the content and return it as ContentData (see page 1378). Only returns content of type CMSContentType.Content (see page 806) and CMSContentType.Assets.

*See Also:* The Example section.

**C#**

```
public ContentData GetContent(Long Id, ContentResultType ContentType);
```

**Visual Basic**

```
Public Function GetContent(ByVal Id As Long, ByVal ContentType As ContentResultType) As
ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The ID of the content to be returned. |
| ContentType | Whether the content is published or staged. Enter either:<br>Ektron.Cms.Content.EkContent.ContentResultType.Published<br>or<br>Ektron.Cms.Content.EkContent.ContentResultType.Staged |

**Returns**

ContentData (see page 1378)

**Example**

The following example shows how to create a Web page from which a user can view a content block and its title by entering the content ID into a text box. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetContent method. This method uses the InternalAdmin to get the content. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
   `<asp:Label ID="Label1" runat="server" Text="Content ID: "></asp:Label>`
2. Add a text box for the user to enter the content ID.
   `<asp:TextBox ID="txtContentId" runat="server" Width="44px"></asp:TextBox>`
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the

Content ID box.

```
   <asp:RequiredFieldValidator ID="rfvContentId" runat="server"
ControlToValidate="txtContentId"
      ErrorMessage="Please enter content ID"></asp:RequiredFieldValidator>
```

4. Add a couple of breaks for spacing purposes.
   ```
   <br /><br />
   ```
5. Add a button to get the content. We will set the button click event in the code behind.
   ```
   <asp:Button ID="btnGetContent" runat="server" Text="Get Content" />
   ```
6. Add a couple of more breaks for spacing purposes.
   ```
   <br /><br />
   ```
7. Add a results label where the content will appear. This also serves as a place to display an error if necessary.
   ```
   <asp:Label ID="lblResult" runat="server" Width="175px"></asp:Label>
   ```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
   Protected Sub btnGetContent_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetContent.Click
```

2. Create an object for the content API (see page 639).

```
       Dim contentApi As New Ektron.Cms.API.Content.Content
```

3. Check to make sure the content ID is numeric.

```
           If Not IsNumeric(txtContentId.Text) Then
               lblResult.Text = "Invalid content ID"
               Exit Sub
           End If
```

4. Create an object for to hold the content data that is returned.

```
       Dim contentItem As Ektron.Cms.ContentData
```

5. Create a Try/Catch around the GetContent method. This allows you to catch any exceptions and helps when debugging.

```
       Try
          contentItem = contentApi.GetContent(CInt(txtContentId.Text),
Ektron.Cms.Content.EkContent.ContentResultType.Published)
```

6. Add an If statement that displays the content or lets the user know if the content ID does not exist.

```
          If (contentItem Is Nothing) Then
             lblResult.Text = "ContentBlock does not exist"
          Else
             lblResult.Text = contentItem.Title & "<br/>" & contentItem.Html
          End If
```

7. If there is an error, let the user know what happened.

```
       Catch ex As Exception
           lblResult.Text = ex.Message
       End Try

   End Sub
```

### Content.GetContent Method (Long, Ektron.Cms.Content.EkContent.ContentResultType)

Loads the content details using the content ID. Only returns content of type CMSContentType.Content (see page 806) and CMSContentType.Assets.

**C#**

```
public ContentData GetContent(Long Id, Optional ByVal
Ektron.Cms.Content.EkContent.ContentResultType Type);
```

**Visual Basic**

```
Public Function GetContent(ByVal Id As Long, Optional ByVal Type As
```

```
Ektron.Cms.Content.EkContent.ContentResultType = ContentResultType.Published) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The content ID |
| Type | |

**Returns**

Ektron.Cms.ContentData (see page 1378)

### Content.GetContentByHistory Method

Retrieves the corresponding content by the history ID.

**C#**

```
public ContentData GetContentByHistory(Long Id);
```

**Visual Basic**

```
Public Function GetContentByHistory(ByVal Id As Long) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The history ID |

**Returns**

Ektron.Cms.ContentData (see page 1378)

### Content.GetContentDisplayLink Method

Function takes in the required information about the string then will decide the best link to use to display the content. It decides between Linkit, a generated quicklink or just the template file.

**C#**

```
public String GetContentDisplayLink(int ContentType, String ContentId, String ContentLanguage,
String FolderId, String host, String StagingDomain, String ProductionDomain);
```

**Visual Basic**

```
Public Function GetContentDisplayLink(ByVal ContentType As Integer, ByVal ContentId As String,
ByVal ContentLanguage As String, ByVal FolderId As String, ByVal host As String, ByVal
StagingDomain As String, ByVal ProductionDomain As String) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentType | Integer describing the content's type (word doc, multimedia, etc.) |
| ContentId | A string value of the content's ID number |
| ContentLanguage | A string value of the content's language number |
| FolderId | A string value of the content'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's staging domain |
| ProductionDomain | A string value of the content's production domain |

**Returns**

The page link as a String

### Content.GetContentForEditing Method

Loads the content details with checkout mode.

**C#**

```
public ContentEditData GetContentForEditing(Long Id);
```

**Visual Basic**

```
Public Function GetContentForEditing(ByVal Id As Long) As ContentEditData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The content ID. |

**Returns**

Ektron.Cms.ContentEditData

## GetContentReport Method

This is the overview for the GetContentReport method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Content.GetContentReport (Collection) (see page 824) | Loads the content data for the given constraints. |
| | Content.GetContentReport (ContentReportData) (see page 824) | Loads the content data for the given constraints. |

## Content.GetContentReport Method (Collection)

Loads the content data for the given constraints.

**C#**

```
public ContentData[] GetContentReport(Collection pagedata);
```

**Visual Basic**

```
Public Function GetContentReport(ByVal pagedata As Collection) As ContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | required report data |

**Returns**

Ektron.Cms.ContentData (see page 1378)()

## Content.GetContentReport Method (ContentReportData)

Loads the content data for the given constraints.

**C#**

```
public ContentData[] GetContentReport(ContentReportData ContReportData);
```

**Visual Basic**

```
Public Function GetContentReport(ByVal ContReportData As ContentReportData) As ContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| ContReportData | required report data |

**Returns**

Ektron.Cms.ContentData (see page 1378)()

### Content.GetContentState Method

Loads the content's details.

**C#**

```
public ContentStateData GetContentState(Long Id);
```

**Visual Basic**

```
Public Function GetContentState(ByVal Id As Long) As ContentStateData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The content's ID |

**Returns**

Ektron.Cms.ContentStateData

### Content.GetContentStatus Method

This will return a single letter string which is the status of the piece of content. The letters returned are the same letters used in the Workarea view.

**C#**

```
public String GetContentStatus(Long ContentId);
```

**Visual Basic**

```
Public Function GetContentStatus(ByVal ContentId As Long) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | Id of the content to return the status of |

**Returns**

A string with the single letter status of the content

### Content.GetDomain Method

Gets web domain for a piece of content.

**C#**

```
public String GetDomain(Long ContentId);
```

**Visual Basic**

```
Public Function GetDomain(ByVal ContentId As Long) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | The content's ID |

**Returns**

String

## Content.GetExpireContent Method

Loads all of the expired content.

**C#**

```
public ContentData[] GetExpireContent(Collection pagedata);
```

**Visual Basic**

```
Public Function GetExpireContent(ByVal pagedata As Collection) As ContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | Required data for the report as a value, key pair |

**Returns**

Ektron.Cms.ContentData (see page 1378)()

## GetList Method

This is the overview for the GetList method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Content.GetList (Criteria(Of ContentProperty)) (see page 826) | Returns a list of content based upon the supplied criteria. |
| | Content.GetList (Criteria(Of ContentProperty), Boolean) (see page 826) | Returns a list of content based upon the supplied criteria. |

## Content.GetList Method (Criteria(Of ContentProperty))

Returns a list of content based upon the supplied criteria.

**C#**

```
public List(Of ContentData) GetList(Criteria(Of ContentProperty) criteria);
```

**Visual Basic**

```
Public Function GetList(ByVal criteria As Criteria(Of ContentProperty)) As List(Of ContentData)
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | criteria used to restrict content retrieval. |

## Content.GetList Method (Criteria(Of ContentProperty), Boolean)

Returns a list of content based upon the supplied criteria.

**C#**

```
public List(Of ContentData) GetList(Criteria(Of ContentProperty) criteria, Boolean
folderRecursive);
```

**Visual Basic**

```
Public Function GetList(ByVal criteria As Criteria(Of ContentProperty), ByVal folderRecursive
As Boolean) As List(Of ContentData)
```

**Parameters**

| Parameters | Description |
|---|---|
| criteria | criteria used to restrict content retrieval. |
| folderRecursive | If true, retrieves all recursive folders based upon folderId in criteria filter. This parameter is only relative if the criteria includes a FolderId filter. |

### Content.MoveContentToFolder Method

Moves content to a folder.

**C#**

```
public ContentData MoveContentToFolder(String cId, String cLanguages, Long FID);
```

**Visual Basic**

```
Public Function MoveContentToFolder(ByVal cId As String, ByVal cLanguages As String, ByVal FID
As Long) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| cId | The content's ID |
| cLanguages | The content's language |
| FID | The folder's ID |

**Returns**

Ektron.Cms.ContentData (see page 1378)

### Content.PublishContent Method

Function takes in information about the content to be checked in and published. This goes through the normal approval chain. The content must be in a checked out state to the current logged in user for this to succeed.

**C#**

```
public Boolean PublishContent(Long ContentId, Long FolderId, int ContentLanguage, String
dontCreateTask, Long UserID, String TaskTitle);
```

**Visual Basic**

```
Public Function PublishContent(ByVal ContentId As Long, ByVal FolderId As Long, ByVal
ContentLanguage As Integer, ByVal dontCreateTask As String, ByVal UserID As Long, ByVal
TaskTitle As String) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | Numeric value of the content's ID. (Long) |
| FolderId | Numeric value of the content folder's ID. (Long) |
| ContentLanguage | Integer value of the content's language ID. |
| dontCreateTask | Set to **""** to create a task for this submit. If anything else appears, a task is not created. |
| UserID | The ID of the user assigning the task (current user). This is needed only when the dontCreateTask property is set to "". Otherwise, this property is ingnored. |

| | |
|---|---|
| TaskTitle | Title for the task. This is needed only when the dontCreateTask property is set to "". Otherwise, this property is ingnored. |

**Returns**

A Boolean indicating success or not.

### Content.RelicateXmlConfiguration Method

**This method is obsolete. The recommended alternative is ReplicateXmlConfiguration (see page 828).**

Replicate the XML configuration.

**C#**

```
[Obsolete("The recommended alternative is ReplicateXmlConfiguration(ByVal Id As Long, ByVal
Title As String)")]
public Long RelicateXmlConfiguration(Long Id, String Title);
```

**Visual Basic**

```
<Obsolete("The recommended alternative is ReplicateXmlConfiguration(ByVal Id As Long, ByVal
Title As String)")>
Public Function RelicateXmlConfiguration(ByVal Id As Long, ByVal Title As String) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The inherited XML ID |
| Title | Title |

**Returns**

Long

### Content.RenameContent Method

Renames the content.

**C#**

```
public ContentData RenameContent(Long cId, String title);
```

**Visual Basic**

```
Public Function RenameContent(ByVal cId As Long, ByVal title As String) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| cId | The content's ID |
| title | The content's new title |

**Returns**

Boolean

### Content.ReplicateXmlConfiguration Method

Replicate the XML configuration.

**C#**

```
public Long ReplicateXmlConfiguration(Long Id, String Title);
```

**Visual Basic**

```
Public Function ReplicateXmlConfiguration(ByVal Id As Long, ByVal Title As String) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The inherited XML ID. |
| Title | The title of the XML configuration. |

**Returns**

A numeric value. (Long)

## Content.RestoreHistoryContent Method

Restores the content to a particular version of content.

**C#**

```
public RestoreHistoryContent(Long Id);
```

**Visual Basic**

```
Public Sub RestoreHistoryContent(ByVal Id As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The content's history ID |

## Content.SaveContent Method

Updates the checked out content block. Only following properties can be updated:

- contentData.Title
- contentData.Teaser
- contentData.Comment
- contentData.GoLive
- contentData.Html
- contentData.EndDate
- contentData.EndDateAction
- contentData.ManualAlias

**C#**

```
public Boolean SaveContent(ContentEditData contentEditData);
```

**Visual Basic**

```
Public Function SaveContent(ByVal contentEditData As ContentEditData) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| contentEditData | _nt_ |

**Returns**

False if successful

**Remarks**

Content (see page 806) must be in the checked out state.

### Content.SubmitForDelete Method

Delete the content by the content's Id and folder Id.

**C#**

```
public Boolean SubmitForDelete(Long cId, Long fId);
```

**Visual Basic**

```
Public Function SubmitForDelete(ByVal cId As Long, ByVal fId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| cId | The content's Id |
| fId | The content's Folder (see page 652) Id |

**Returns**

Boolean

### Content.UndoCheckout Method

This undoes the checkout for a piece of content which is checked out to the current user.

**C#**

```
public Boolean UndoCheckout(Long ContentId);
```

**Visual Basic**

```
Public Function UndoCheckout(ByVal ContentId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | Id of the content for which to undo check out |

**Returns**

A Boolean representing the success of the undo.

### Content.UpdateContentMetaData Method

Updates a single piece of content metadata

**C#**

```
public Boolean UpdateContentMetaData(Long ContentID, Long MetaID, String MetaText);
```

**Visual Basic**

```
Public Function UpdateContentMetaData(ByVal ContentID As Long, ByVal MetaID As Long, ByVal
MetaText As String) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The Content (see page 806) ID |
| MetaID | The Metadata (see page 683) Type ID |
| MetaText | The new text to assign |

**Returns**

boolean

### Content.UpdatexmlCollectionPackage Method

Updates the XML collection package.

**C#**

```
public UpdatexmlCollectionPackage(Collection package);
```

**Visual Basic**

```
Public Sub UpdatexmlCollectionPackage(ByVal package As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| package | The XML details |

### Content.UpdateXmlConfiguration Method

Updates the XML configuration details.

**C#**

```
public UpdateXmlConfiguration(Collection pagedata);
```

**Visual Basic**

```
Public Sub UpdateXmlConfiguration(ByVal pagedata As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| pagedata | The XML configuration details |

## ContentRating Class

**Class Hierarchy**

**C#**

```
public class ContentRating : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class ContentRating
Inherits Ektron.Cms.API.Common
```

**File**

ContentRating.vb

**Description**

This is class Ektron.Cms.API.Content.ContentRating.

### ContentRating Methods

The methods of the ContentRating class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | GetContentRating (see page 832) | Constructs a string that represents a content rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content rating server control wraps this functionality. |
| | GetContentRatingAddRating (see page 834) | Adds a content rating to the database. Internally a check is done to ensure the user has not already rated the content, and passes down the rating and any comments the user supplied. |

| | | |
|---|---|---|
| | GetContentRatingResults (see page 834) | Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments. |
| | GetContentRatingStatistics (see page 834) | This is the overview for the GetContentRatingStatistics method overload. |
| | PurgeContentRatings (see page 836) | Purges all ratings of a particular content block over a defined date/time range. |

## ContentRating.GetContentRating Method

Constructs a string that represents a content rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content rating server control wraps this functionality.

### C#

```
public String GetContentRating(Long contentID, System.Web.UI.Page currentPage, String key,
ContentRatingLabels Labels, ContentAPI.RatingGraphDisplayType initialGraph,
ContentAPI.RatingGraphDisplayType ratingCompleteGraph, int padding, System.Drawing.Color
bgColor, System.Drawing.Color barColor, System.Drawing.Color TextColor);
```

### Visual Basic

```
Public Function GetContentRating(ByVal contentID As Long, ByVal currentPage As
System.Web.UI.Page, ByVal key As String, ByVal Labels As ContentRatingLabels, ByVal
initialGraph As ContentAPI.RatingGraphDisplayType, ByVal ratingCompleteGraph As
ContentAPI.RatingGraphDisplayType, ByVal padding As Integer, ByVal bgColor As
System.Drawing.Color, ByVal barColor As System.Drawing.Color, ByVal TextColor As
System.Drawing.Color) As String
```

### Parameters

| Parameters | Description |
|---|---|
| contentID | The numeric ID associated with the content 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 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 (see page 1391) |
| 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 831) server control nicely wraps this function.

### Example

The following example shows how to create a Web page that retrieves a content rating element used to rate content. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetContentRating method. This method uses the InternalAdmin to retrieve the content. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a Label with the following information.
   ```
   <asp:Label ID="Label6" runat="server" Text="Content ID:" Width="76px"></asp:Label>
   ```
2. Add a text box for the user to enter the content ID.
   ```
   <asp:TextBox ID="txtId" runat="server" Width="55px"></asp:TextBox>
   ```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Content ID box.

```
   <asp:RequiredFieldValidator ID="rfvId" runat="server" ControlToValidate="txtId"
ErrorMessage="Please enter content id" Width="140px"></asp:RequiredFieldValidator>
```

4. Add a couple of breaks for spacing purposes.
   ```
   <br /><br />
   ```
5. Add a button to submit the content ID and retrieve the content rating.
   ```
   <asp:Button ID="btnGet" runat="server" Text="Get ContentRating" Width="115px" />
   ```
6. Add a couple of more breaks for spacing purposes.
   ```
   <br /><br />
   ```
7. Add a results label to display the content rating.
   ```
   <asp:Label ID="lblResult" runat="server" Width="502px"></asp:Label>
   ```

Add the following information to the code behind of your aspx.vb page.

1. Add the following Imports statement to the top of the page.

```
Imports Ektron.Cms
```

2. Add a page load event that makes sure lblResult.Text is blank.

```
        Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
            lblResult.Text = ""
        End Sub
```

3. Add a button click event.

```
        Protected Sub btnGet_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGet.Click
```

4. Check to make sure the content ID is numeric.

```
            If Not IsNumeric(txtId.Text) Then
                lblResult.Text = "Invalid Content ID"
                Exit Sub
            End If
```

5. Create the following objects.

```
            Dim ratingApi As New Ektron.Cms.API.Content.ContentRating
            Dim labels As New Ektron.Cms.ContentRatingLabels
            Dim str As String
```

6. Define the labels for use in the content rating.

```
            labels.TotalRatingsLabel = "Total number of ratings"
            labels.RatingLevelLabel = "Rating Level: "
            labels.AlreadyRatedMessage = "You have rated this content"
            labels.GoodLabel = "Good"
            labels.BadLabel = "Bad"
            labels.FeedbackHeader = "How would you rate this content?"
            labels.UserCommentsHeader = "Enter any feedback below."
            labels.CharsRemainingLabel = "Remaining Characters: "
```

7. Create a Try/Catch around the GetContentRating method. This allows you to catch any exceptions and helps when debugging.

```
            Try
               str = ratingApi.GetContentRating(CInt(txtId.Text), Me.Page, "", labels,
ContentAPI.RatingGraphDisplayType.Horizontal, _
                     ContentAPI.RatingGraphDisplayType.Horizontal, 10, Drawing.Color.Aqua,
Drawing.Color.Red, Drawing.Color.Black)
```

8. Display the ratings.

```
            If str.Length > 0 Then
               lblResult.Text = str
            Else
               lblResult.Text = "No content rating information available"
            End If
```

9. If there is an error, let the user know what happened.

```
            Catch ex As Exception
                lblResult.Text = ex.Message
            End Try

        End Sub
```

### ContentRating.GetContentRatingAddRating Method

Adds a content rating to the database. Internally a check is done to ensure the user has not already rated the content, and passes down the rating and any comments the user supplied.

**C#**

```
public GetContentRatingAddRating(Long contentid, Long userid, int userRating, String
userComments);
```

**Visual Basic**

```
Public Sub GetContentRatingAddRating(ByVal contentid As Long, ByVal userid As Long, ByVal
userRating As Integer, ByVal userComments As String)
```

**Parameters**

| Parameters | Description |
|---|---|
| contentid | The numeric ID associated with the content 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 from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments.

**C#**

```
public System.Data.DataSet GetContentRatingResults(Long contentID, DateTime startDate,
DateTime endDate);
```

**Visual Basic**

```
Public Function GetContentRatingResults(ByVal contentID As Long, ByVal startDate As DateTime,
ByVal endDate As DateTime) As System.Data.DataSet
```

**Parameters**

| Parameters | Description |
|---|---|
| contentID | The numeric ID associated with the content 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

This is the overview for the GetContentRatingStatistics method overload.

**Description**

Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate.

This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | ContentRating.GetContentRatingStatistics (Long, DateTime, DateTime) (see page 835) | 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.<br>This function is overloaded. See Also: GetContentRatingStatistics(Long Long String) |
| | ContentRating.GetContentRatingStatistics (Long, Long, String) (see page 835) | 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.<br>This function is overloaded. See Also: GetContentRatingStatistics(Long DateTime DateTime) |

## ContentRating.GetContentRatingStatistics Method (Long, DateTime, DateTime)

Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate.

This function is overloaded. See Also: GetContentRatingStatistics(Long Long String)

**C#**

```
public Collection GetContentRatingStatistics(Long contentID, DateTime startDate, DateTime
endDate);
```

**Visual Basic**

```
Public Function GetContentRatingStatistics(ByVal contentID As Long, ByVal startDate As
DateTime, ByVal endDate As DateTime) As Collection
```

**Parameters**

| Parameters | Description |
|---|---|
| contentID | The numeric ID associated with the content 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 (Long, Long, String)

Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate.

This function is overloaded. See Also: GetContentRatingStatistics(Long DateTime DateTime)

**C#**

```
public Collection GetContentRatingStatistics(Long contentid, Long user_id, String visitor_id);
```

**Visual Basic**

```
Public Function GetContentRatingStatistics(ByVal contentid As Long, ByVal user_id As Long,
ByVal visitor_id As String) As Collection
```

### Parameters

| Parameters | Description |
|---|---|
| contentid | The numeric ID associated with the content block being rated. |

### Returns

Returns a collection of data with keys "total" => number of ratings, "sum" => sum of all ratings, "r1, "r2", "r3", "r4", "r5", "r6", "r7", "r8", "r9".

### Remarks

Note that average rating = sum/total.

## ContentRating.PurgeContentRatings Method

Purges all ratings of a particular content block over a defined date/time range.

### C#

```
public PurgeContentRatings(Long contentid, DateTime startDate, DateTime endDate);
```

### Visual Basic

```
Public Sub PurgeContentRatings(ByVal contentid As Long, ByVal startDate As DateTime, ByVal
endDate As DateTime)
```

### Parameters

| Parameters | Description |
|---|---|
| contentid | The ID of the content that is having its ratings purged. |
| startDate | The beginning of the date range of content to be purged. |
| endDate | The end of the date range of content to be purged. |

# Form Class

### Class Hierarchy

### C#

```
public class Form : Content;
```

### Visual Basic

```
Public Class Form
Inherits Content
```

### File

Form.vb

### Description

This is class Ektron.Cms.API.Content.Form.

## Form Methods

The methods of the Form class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AssignFormTask (see page 837) | Assigns a form submission task, as specified in FormInfo (see page 560), of the SubmittedDataXml. |

| | | |
|---|---|---|
| | EmailFormFieldData (see page 837) | Sends an email, as specified in FormInfo (see page 560), of the SubmittedDataXml. |
| | GetAllForms (see page 838) | This is the overview for the GetAllForms method overload. |
| | GetForm (see page 839) | Retrieves information about the specified form. |
| | GetFormFieldData (see page 839) | This is the overview for the GetFormFieldData method overload. |
| | GetFormFieldList (see page 840) | Retrieves the list of form fields defined in the specified form. |
| | GetFormFieldQuestions (see page 840) | 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 841) | Same as GetFormById, but returns only the form's title. |
| | SubmitFormFieldData (see page 841) | Adds form data as if it was submitted during a browser post to the web server. |
| | UpdateFormFieldData (see page 841) | Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form. |

## Form.AssignFormTask Method

Assigns a form submission task, as specified in FormInfo (see page 560), 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: Ektron.Cms.FormData (see page 1394) |
| SubmittedDataXml | The form data submitted in XML format. Value Type: System.String |

**Returns**

Returns True if the task was successfully assigned. Value Type: System.Boolean

**Remarks**

This routine is called automatically when calling SubmitFormFieldData (see page 841) 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 841).

## Form.EmailFormFieldData Method

Sends an email, as specified in FormInfo (see page 560), 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: Ektron.Cms.FormData (see page 1394) |
| SubmittedDataXml | The form data submitted in XML format. Value Type: System.String |

### Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 841) 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 841).

## GetAllForms Method

This is the overview for the GetAllForms method overload.

### Description

Retrieves an array of all the forms contained with the specified folder. This method is overloaded.

### Overload List

| | Name | Description |
|---|---|---|
| | Form.GetAllForms () (see page 838) | Returns all of the forms as an array of FormData (see page 1394). |
| | Form.GetAllForms (Long, String, Integer, Cms.ContentAPI.FormResultType) (see page 838) | 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 (see page 1394).

### C#

```
public FormData[] GetAllForms();
```

### Visual Basic

```
Public Function GetAllForms() As FormData()
```

### Returns

Ektron.Cms.FormData (see page 1394)()

## Form.GetAllForms Method (Long, String, Integer, Cms.ContentAPI.FormResultType)

Retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type.

### C#

```
public FormData[] GetAllForms(Long Id, String OrderBy, int LanguageId,
Cms.ContentAPI.FormResultType Type);
```

### Visual Basic

```
Public Function GetAllForms(ByVal Id As Long, ByVal OrderBy As String, ByVal LanguageId As
Integer, ByVal Type As Cms.ContentAPI.FormResultType) As FormData()
```

### Parameters

| Parameters | Description |
|---|---|
| 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. |

## Form.GetForm Method

Retrieves information about the specified form.

**C#**

```
public FormData GetForm(Long Id);
```

**Visual Basic**

```
Public Function GetForm(ByVal Id As Long) As FormData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | Specifies the form. |

**Returns**

Returns information about the form.

## GetFormFieldData Method

This is the overview for the GetFormFieldData method overload.

**Description**

Retrieves the submitted data for all fields from a specified form. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | Form.GetFormFieldData (Long, Long) (see page 839) | Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form. |
| | Form.GetFormFieldData (Long, String, String) (see page 840) | Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range. |

## Form.GetFormFieldData Method (Long, Long)

Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

**C#**

```
public FormSubmittedData GetFormFieldData(Long FormId, Long FormDataId);
```

**Visual Basic**

```
Public Function GetFormFieldData(ByVal FormId As Long, ByVal FormDataId As Long) As
FormSubmittedData
```

**Parameters**

| Parameters | Description |
|---|---|
| FormId | Specifies the form. Value Type: System.Int32 |
| FormDataId | Specifies the instance of submitted data. Value Type: System.Int32 |

**Returns**

Returns the submitted data or null/Nothing, if none found. Ektron.Cms.FormSubmittedData (see page 1412)

### Form.GetFormFieldData Method (Long, String, String)

Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range.

**C#**

```
public FormSubmittedData[] GetFormFieldData(Long FormId, Optional ByVal String StartDate,
Optional ByVal String EndDate);
```

**Visual Basic**

```
Public Function GetFormFieldData(ByVal FormId As Long, Optional ByVal StartDate As String =
"", Optional ByVal EndDate As String = "") As FormSubmittedData()
```

**Parameters**

| Parameters | Description |
|---|---|
| FormId | Specifies the form. Value Type: System.Int32 |
| StartDate | Optionally limit the data returned to include only those submitted on or after the given date. Value Type: System.String |
| EndDate | Optionally limit the data returned to include only those submitted on or before the given date. Value Type: System.String |

**Returns**

Returns an array of submitted data. Ektron.Cms.FormSubmittedData (see page 1412)

### Form.GetFormFieldList Method

Retrieves the list of form fields defined in the specified form.

**C#**

```
public FormFieldList GetFormFieldList(Long FormId);
```

**Visual Basic**

```
Public Function GetFormFieldList(ByVal FormId As Long) As FormFieldList
```

**Parameters**

| Parameters | Description |
|---|---|
| FormId | Specifies the form. Value Type: System.Int32 |

**Returns**

The list for defined fields in the given form. Ektron.Cms.FormFieldList (see page 1407)

### Form.GetFormFieldQuestions Method

Retrieves the table of form fields and their questions in a specified form. In a CMS form, the form question is defined by the user in the descriptive name field in the form element dialog during design time.

**C#**

```
public Hashtable GetFormFieldQuestions(Long FormId);
```

**Visual Basic**

```
Public Function GetFormFieldQuestions(ByVal FormId As Long) As Hashtable
```

**Parameters**

| Parameters | Description |
|---|---|
| FormId | Specifies the form. Value Type: System.Int32 |

**Returns**

A table of defined fields with the corresponding questions in a given form.

**Remarks**

These questions are mainly used for reporting.

## Form.GetFormTitle Method

Same as GetFormById, but returns only the form's title.

**C#**

```
public String GetFormTitle(Long Id);
```

**Visual Basic**

```
Public Function GetFormTitle(ByVal Id As Long) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | Specifies the form. |

**Returns**

The title of the specified form.

## Form.SubmitFormFieldData Method

Adds form data as if it was submitted during a browser post to the web server.

**C#**

```
public SubmitFormFieldData(Long FormId, FormSubmittedData Data, Optional ByVal Boolean
SendNotification, [Out()] Optional ByRef String SubmittedDataXml);
```

**Visual Basic**

```
Public Sub SubmitFormFieldData(ByVal FormId As Long, ByVal Data As FormSubmittedData, Optional
ByVal SendNotification As Boolean = True, <Out()> Optional ByRef SubmittedDataXml As String =
"")
```

**Parameters**

| Parameters | Description |
|---|---|
| FormId | Specifies the form that corresponds to the data. Value Type: System.Int32 |
| Data | The data to submit. Value Type: Ektron.Cms.FormSubmittedData (see page 1412) |
| 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: System.Boolean |
| SubmittedDataXml | (Optional) XML packet of submitted data. Reference Type: System.String |

## Form.UpdateFormFieldData Method

Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

**C#**

```
public UpdateFormFieldData(Long FormId, Long FormDataId, FormSubmittedData SubmittedData);
```

**Visual Basic**

```
Public Sub UpdateFormFieldData(ByVal FormId As Long, ByVal FormDataId As Long, ByVal
SubmittedData As FormSubmittedData)
```

### Parameters

| Parameters | Description |
|---|---|
| FormId | Specifies the form. Value Type: System.Int32 |
| FormDataId | Specifies the instance of submitted data. Value Type: System.Int32 |
| SubmittedData | The field data values to be updated. Value Type: Ektron.Cms.FormSubmittedData (see page 1412) |

## Taxonomy Class

### Class Hierarchy

### C#

```
public class Taxonomy : Content;
```

### Visual Basic

```
Public Class Taxonomy
Inherits Content
```

### File

Taxonomy.vb

### Description

This is class Ektron.Cms.API.Content.Taxonomy.

## Taxonomy Methods

The methods of the Taxonomy class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddTaxonomyItem (see page 843) | This is the overview for the AddTaxonomyItem method overload. |
| | AddTaxonomySyncFolder (see page 843) | Adds a folder entry into taxonomy. |
| | CloneTaxonomy (see page 844) | Copy or move a group of taxonomy categories from one taxonomy to another. For example, you want move categories one, two and three from taxonomy A to taxonomy B. Copying or moving a taxonomy category depends on whether the DeleteSource parameter is set True or False. If DeleteSouce is set to True, the taxonomy category is deleted. You can specify how many node levels are included by setting the Depth parameter. You can also choose whether to move/copy items in the category or just clone the category's structure. |
| | CreateTaxonomy (see page 844) | Creates a new taxonomy. |
| | DeleteTaxonomy (see page 844) | Deletes the existing taxonomy. |
| | GetTaxonomyIdByPath (see page 845) | Returns the taxonomy ID by a given path. |
| | ImportTaxonomy (see page 845) | This is the overview for the ImportTaxonomy method overload. |
| | LoadTaxonomy (see page 846) | Returns the TaxonomyData with all of its subcategory and its items. |
| | ReadAllAssignedCategory (see page 846) | Returns all the assigned categories for the content ID. |
| | ReadTaxonomy (see page 847) | Read the taxonomy data. |
| | RemoveTaxonomyFolder (see page 847) | Removes the sync up folder from taxonomy. |
| | RemoveTaxonomyItem (see page 847) | Removes an item from the taxonomy. |
| | ReOrderTaxonomyItems (see page 847) | Reorder the display of the taxonomy items. |

| | | |
|---|---|---|
| | UpdateTaxonomy (see page 848) | Updates an existing taxonomy. |

## AddTaxonomyItem Method

This is the overview for the AddTaxonomyItem method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | Taxonomy.AddTaxonomyItem (TaxonomyContentRequest) (see page 843) | Inserts an item into taxonomy. |
| | Taxonomy.AddTaxonomyItem (TaxonomyRequest) (see page 843) | Inserts an item into the taxonomy. |

## Taxonomy.AddTaxonomyItem Method (TaxonomyContentRequest)

Inserts an item into taxonomy.

**C#**

```
public AddTaxonomyItem(TaxonomyContentRequest request);
```

**Visual Basic**

```
Public Sub AddTaxonomyItem(ByVal request As TaxonomyContentRequest)
```

### Parameters

| Parameters | Description |
|---|---|
| request | TaxonomyContentRequest |

## Taxonomy.AddTaxonomyItem Method (TaxonomyRequest)

Inserts an item into the taxonomy.

**C#**

```
public AddTaxonomyItem(TaxonomyRequest request);
```

**Visual Basic**

```
Public Sub AddTaxonomyItem(ByVal request As TaxonomyRequest)
```

### Parameters

| Parameters | Description |
|---|---|
| request | TaxonomyRequest (see page 1414) |

## Taxonomy.AddTaxonomySyncFolder Method

Adds a folder entry into taxonomy.

**C#**

```
public AddTaxonomySyncFolder(TaxonomySyncRequest request);
```

**Visual Basic**

```
Public Sub AddTaxonomySyncFolder(ByVal request As TaxonomySyncRequest)
```

**Parameters**

| Parameters | Description |
|---|---|
| request | TaxonomyRequest (see page 1414) |

## Taxonomy.CloneTaxonomy Method

Copy or move a group of taxonomy categories from one taxonomy to another. For example, you want move categories one, two and three from taxonomy A to taxonomy B. Copying or moving a taxonomy category depends on whether the DeleteSource parameter is set True or False. If DeleteSouce is set to True, the taxonomy category is deleted. You can specify how many node levels are included by setting the Depth parameter. You can also choose whether to move/copy items in the category or just clone the category's structure.

**C#**

```
public Long CloneTaxonomy(Long sourceId, Long destinationId, int LanguageId, int Depth,
Boolean WithItems, Boolean DeleteSource);
```

**Visual Basic**

```
Public Function CloneTaxonomy(ByVal sourceId As Long, ByVal destinationId As Long, ByVal
LanguageId As Integer, ByVal Depth As Integer, ByVal WithItems As Boolean, ByVal DeleteSource
As Boolean) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| sourceId | The ID of the source category being moved or copied. |
| destinationId | The target ID where the category will be moved or copied. |
| LanguageId | The category's language ID. |
| Depth | Specify an integer that represents the amount of category sub levels to be copied or moved. For example, if you want to copy two category levels below the source category, enter **2**.<br>To copy or move all sub categories enter **-1**. |
| WithItems | Enter True to copy or move all items from the source structure. If you enter false, copy/move only the taxonomy category's structure. |
| DeleteSource | Enter True to move the source category. This deletes the source category after the node is cloned. Set to false to copy the source category. This leaves the original category in its structure. |

**Returns**

int

## Taxonomy.CreateTaxonomy Method

Creates a new taxonomy.

**C#**

```
public Long CreateTaxonomy(TaxonomyData request);
```

**Visual Basic**

```
Public Function CreateTaxonomy(ByVal request As TaxonomyData) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| request | TaxonomyData |

**Returns**

Returns an ID for the newly created Taxonomy (see page 842).

## Taxonomy.DeleteTaxonomy Method

Deletes the existing taxonomy.

**C#**

```
public DeleteTaxonomy(TaxonomyRequest request);
```

**Visual Basic**

```
Public Sub DeleteTaxonomy(ByVal request As TaxonomyRequest)
```

**Parameters**

| Parameters | Description |
|---|---|
| request | TaxonomyRequest (see page 1414) |

## Taxonomy.GetTaxonomyIdByPath Method

Returns the taxonomy ID by a given path.

**C#**

```
public Long GetTaxonomyIdByPath(String path);
```

**Visual Basic**

```
Public Function GetTaxonomyIdByPath(ByVal path As String) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| path | The path of the taxonomy category. For example, "\New Hampshire\Restaurants\Pizza" |

**Returns**

Taxonomy (see page 842) ID

## ImportTaxonomy Method

This is the overview for the ImportTaxonomy method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Taxonomy.ImportTaxonomy (String, String) (see page 845) | Creates a taxonomy using XML file. |
| | Taxonomy.ImportTaxonomy (String, String, List(Of TaxonomyType)) (see page 846) | Creates a taxonomy using an XML file. |

## Taxonomy.ImportTaxonomy Method (String, String)

Creates a taxonomy using XML file.

**C#**

```
public Long ImportTaxonomy(String TaxonomyXml, String Title);
```

**Visual Basic**

```
Public Function ImportTaxonomy(ByVal TaxonomyXml As String, ByVal Title As String) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| TaxonomyXml | The XML being used to create a taxonomy. |
| Title | The title for the new taxonomy. |

**Returns**

Returns a numeric ID for the newly created taxonomy.

### Taxonomy.ImportTaxonomy Method (String, String, List(Of TaxonomyType))

Creates a taxonomy using an XML file.

**C#**

```
public Long ImportTaxonomy(String TaxonomyXml, String Title, List(Of TaxonomyType) config);
```

**Visual Basic**

```
Public Function ImportTaxonomy(ByVal TaxonomyXml As String, ByVal Title As String, ByVal
config As List(Of TaxonomyType)) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| TaxonomyXml | The XML being used to create a taxonomy. |
| Title | The title for the new taxonomy. |
| config | Configuration list that can be used. |

**Returns**

Returns a numeric ID for the newly created taxonomy.

### Taxonomy.LoadTaxonomy Method

Returns the TaxonomyData with all of its subcategory and its items.

**C#**

```
public TaxonomyData LoadTaxonomy(ref TaxonomyRequest request);
```

**Visual Basic**

```
Public Function LoadTaxonomy(ByRef request As TaxonomyRequest) As TaxonomyData
```

**Parameters**

| Parameters | Description |
|---|---|
| request | TaxonomyRequest (see page 1414) |

**Returns**

TaxonomyData

### Taxonomy.ReadAllAssignedCategory Method

Returns all the assigned categories for the content ID.

**C#**

```
public TaxonomyBaseData[] ReadAllAssignedCategory(Long contentid);
```

**Visual Basic**

```
Public Function ReadAllAssignedCategory(ByVal contentid As Long) As TaxonomyBaseData()
```

**Parameters**

| Parameters | Description |
|---|---|
| contentid | A content's numeric ID. |

**Returns**

An array of TaxonomyBaseData.

### Taxonomy.ReadTaxonomy Method

Read the taxonomy data.

**C#**

```
public TaxonomyData ReadTaxonomy(ref TaxonomyRequest request);
```

**Visual Basic**

```
Public Function ReadTaxonomy(ByRef request As TaxonomyRequest) As TaxonomyData
```

**Parameters**

| Parameters | Description |
|---|---|
| request | TaxonomyRequest (see page 1414) |

**Returns**

Returns Taxonomydata

### Taxonomy.RemoveTaxonomyFolder Method

Removes the sync up folder from taxonomy.

**C#**

```
public RemoveTaxonomyFolder(TaxonomySyncRequest request);
```

**Visual Basic**

```
Public Sub RemoveTaxonomyFolder(ByVal request As TaxonomySyncRequest)
```

**Parameters**

| Parameters | Description |
|---|---|
| request | TaxonomyRequest (see page 1414) |

### Taxonomy.RemoveTaxonomyItem Method

Removes an item from the taxonomy.

**C#**

```
public RemoveTaxonomyItem(TaxonomyRequest request);
```

**Visual Basic**

```
Public Sub RemoveTaxonomyItem(ByVal request As TaxonomyRequest)
```

**Parameters**

| Parameters | Description |
|---|---|
| request | TaxonomyRequest (see page 1414) |

### Taxonomy.ReOrderTaxonomyItems Method

Reorder the display of the taxonomy items.

**C#**

```
public ReOrderTaxonomyItems(TaxonomyRequest request);
```

**Visual Basic**

```
Public Sub ReOrderTaxonomyItems(ByVal request As TaxonomyRequest)
```

### Parameters

| Parameters | Description |
|---|---|
| request | TaxonomyReqeust |

### Taxonomy.UpdateTaxonomy Method

Updates an existing taxonomy.

**C#**

```
public UpdateTaxonomy(TaxonomyData request);
```

**Visual Basic**

```
Public Sub UpdateTaxonomy(ByVal request As TaxonomyData)
```

### Parameters

| Parameters | Description |
|---|---|
| request | TaxonomyData |

## ThreadedDiscussion Class

The ThreadedDiscussion class allows you to work with threaded discussion by adding and removing discussion boards, forums, and topics. It also allows you to subscribe or unsubscribe to a board, forum or topic. In addition, you can retrieve a specific topic or search a topic's replies.

The Threaded Discussion feature provides a discussion board where a forum can be created and topics can be discussed on your Web site. A site or membership user with Add Topic permission creates a topic and posts a starter question. Other site visitors can then reply to the topic.

### Class Hierarchy

**C#**

```
public class ThreadedDiscussion : Content;
```

**Visual Basic**

```
Public Class ThreadedDiscussion
Inherits Content
```

### File

ThreadedDiscussion.vb

### ThreadedDiscussion Methods

The methods of the ThreadedDiscussion class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddBoard (see page 849) | Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns numeric ID of new discussion board. |
| | AddForum (see page 850) | Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns numeric ID of new discussion forum as Long.<br>*See Also:* The Example section. |
| | AddReply (see page 852) | This is the overview for the AddReply method overload. |
| | AddTopic (see page 853) | This is the overview for the AddTopic method overload. |
| | DeleteBoard (see page 854) | Deletes a board with a given ID. This deletes the board as well as all forums, topics and posts under it. BoardID must be greater than 0. |

| | | |
|---|---|---|
| | DeleteForum (see page 855) | Deletes a forum with a given ID. This deletes the forum as well as all topics and posts under it. forumID must be greater than 0 (zero). |
| | DeleteTopic (see page 855) | Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value. |
| | GetActiveTopics (see page 855) | This is the overview for the GetActiveTopics method overload. |
| | GetActiveTopicsForForum (see page 856) | Gets Most active topics for a forum. Activity is measured by number of posts on the topic. |
| | GetActiveUsers (see page 856) | Gets Most active topics for a board. Activity is measured by number of posts on the topic. |
| | GetBoardCategories (see page 857) | 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 857) | 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 857) | This is the overview for the GetDiscussionBoard method overload. |
| | GetDiscussionBoardString (see page 858) | Method to get the rendered view of a discussion board by ID. Returns a string of the output based upon the input parameters. Display of the board can be controlled by using cascading style sheets (CSS). |
| | GetForum (see page 859) | Method to obtain a particular forum by ID. Returns a discussion board with a 1 length Discussion Forum (see page 256) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.<br>*See Also:* The Example section. |
| | GetRepliesForTopics (see page 860) | Method returns a list of replies for a given forum ID.<br>• ArrayList Structure ArrayListItem(j)(0)<br>• content_id ArrayListItem(j)(1)<br>• comment_total ArrayListItem(j)(2)<br>• comment_sum ArrayListItem(j)(3)<br>• last_posted ArrayListItem(j)(4)<br>• content_language |
| | GetTopic (see page 861) | Method to obtain a particular Topic by ID. Returns a discussion board with a 1 length Discussion Forum (see page 256) 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 861) | Method to obtain an array of topic replies matching search criteria. Returns an instance of EkTasks, which is effectively an array of EkTask objects. If no results are returned, then a zero length EkTasks will be returned. |
| | SubscribeToBoard (see page 862) | Subscribe a user to a board by providing the ID of the board, the user's ID and the type of subscription. |
| | SubscribeToForum (see page 862) | Subscribe a user to a forum by providing the ID of the forum, the user's ID and the type of subscription. |
| | SubscribeToTopic (see page 862) | Subscribe a user to a topic by providing the ID of the topic, the user's ID and the type of subscription. |
| | UnSubscribeToBoard (see page 863) | Unsubscribe a user from a discussion board by providing the discussion board's ID and user ID. |
| | UnSubscribeToForum (see page 863) | Unsubscribe a user from a forum by providing the topic ID and user ID. |
| | UnSubscribeToTopic (see page 864) | Unsubscribe a user from a topic by providing the topic ID and user ID. |

## ThreadedDiscussion.AddBoard Method

Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns numeric ID of new discussion board.

**C#**

```
public Long AddBoard(Long ParentID, String Name, String Title, Boolean RequiresAuthentication,
String StyleSheet, String Categories);
```

**Visual Basic**

```
Public Function AddBoard(ByVal ParentID As Long, ByVal Name As String, ByVal Title As String,
ByVal RequiresAuthentication As Boolean, ByVal StyleSheet As String, ByVal Categories() As
String) As Long
```

### Parameters

| Parameters | Description |
|---|---|
| ParentID | Numeric ID of folder to which you are adding the discussion board. |
| Name | Name of discussion board. |
| Title | Title of the discussion board. |
| RequiresAuthentication | Set to True to require authentication. |
| StyleSheet | The stylesheet to apply to the discussion board. |
| Categories | String array of categories for the discussion board. |

### Returns

Numeric ID of new discussion board. (Long)

## ThreadedDiscussion.AddForum Method

Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns numeric ID of new discussion forum as Long.

*See Also:* The Example section.

### C#

```
public Long AddForum(Long BoardID, String Name, String Description, Boolean ModeratePosts,
Boolean LockForum, int SortOrder, Long CategoryID);
```

### Visual Basic

```
Public Function AddForum(ByVal BoardID As Long, ByVal Name As String, ByVal Description As
String, ByVal ModeratePosts As Boolean, ByVal LockForum As Boolean, ByVal SortOrder As
Integer, ByVal CategoryID As Long) As Long
```

### Parameters

| Parameters | Description |
|---|---|
| BoardID | The numeric ID of the board to which the forum will be added. |
| Name | The string name of the forum being added. |
| Description | String value for the forum summary. |
| ModeratePosts | Are posts moderated? |
| LockForum | Is the forum locked?<br>**True** - prevent posting to the forum. Only members of the Administrators user group can add, edit or delete posts.<br>**False** - the forum is unlocked. |
| SortOrder | Sort order for the new forum. For example, if you have multiple forums, you can assign an integer order to sort them. The order in which the forums appear, correspond to the numbers assigned to each one. 1 (one) is first on the list, 2 (two) appears second and so on. If you assign the same number to more than one forum, those forums are sorted in alphabetical order. |
| CategoryID | Numeric ID of the category to which the forum belongs. |

### Returns

Numeric ID of new discussion forum. (Long)

### Example

The following example shows how to create a Web page from which a forum can be added to a discussion board. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddForum method. This method uses the InternalAdmin to add the forum. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. In the Web form, create a table with four rows and three columns.
2. In the first row, first column, add the following label:
   ```
   <asp:Label ID="Label1" runat="server" Text="Board Id: "></asp:Label>
   ```
3. In the first row, second column, add a text box to enter the board ID.
   ```
   <asp:TextBox ID="txtBoardId" runat="server" Width="63px"></asp:TextBox>
   ```

4. In the first row, third column, add a RequiredFieldValidator control. This validates that something appears in the board ID box.
```
<asp:RequiredFieldValidator ID="rfvBoardId" runat="server" ControlToValidate="txtBoardId"
  ErrorMessage="Please enter board ID"></asp:RequiredFieldValidator>
```
5. In the second row, first column, add the following label:
```
<asp:Label ID="Label4" runat="server" Text="Forum name: " Width="80px"></asp:Label>
```
6. In the second row, second column, add a text box to enter the forum name.
```
<asp:TextBox ID="txtForumName" runat="server" Width="254px"></asp:TextBox>
```
7. In the second row, third column, add a RequiredFieldValidator control. This validates that something appears in the Forum Name box.
```
<asp:RequiredFieldValidator ID="rfvForumName" runat="server" ErrorMessage="Please enter
forum name"
   ControlToValidate="txtForumName"></asp:RequiredFieldValidator>
```
8. In the third row, first column, add the following label:
```
<asp:Label ID="Label7" runat="server" Text="Description:" Width="76px"></asp:Label>
```
9. In the third row, second column, add a text box to enter a description.
```
<asp:TextBox ID="txtDescription" runat="server" Width="253px"></asp:TextBox>
```
10. In the third row, third column, add a RequiredFieldValidator control. This validates that something appears in the description box.
```
<asp:RequiredFieldValidator ID="rfvDescription" runat="server" ErrorMessage="Please enter
forum description"
   ControlToValidate="txtDescription" Width="186px"></asp:RequiredFieldValidator>
```
11. In the fourth row, first column, add the following label:
```
<asp:Label ID="Label2" runat="server" Text="Category ID: " Width="76px"></asp:Label>
```
12. In the fourth row, second column, add a text box to enter a Category ID.
```
<asp:TextBox ID="txtCategoryId" runat="server" Width="75px"></asp:TextBox>
```
13. In the fourth row, third column, add a RequiredFieldValidator control. This validates that something appears in the Category ID box.
```
<asp:RequiredFieldValidator ID="rfvCatId" runat="server" ErrorMessage="Please enter
category Id"
   ControlToValidate="txtCategoryId"></asp:RequiredFieldValidator>
```
14. After the table, add a couple of breaks for spacing purposes.
```
<br /><br />
```
15. Add a button, to submit the information.
```
<asp:Button ID="btnAddForum" runat="server" Text="Add Forum" Width="72px" />
```
16. Add a couple of more breaks for spacing purposes.
```
<br /><br />
```
17. Add a results label that will let the person using the page know whether or not the add succeeded.
```
<asp:Label ID="lblResult" runat="server" Width="549px" Height="25px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
    Protected Sub btnGetForum_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnAddForum.Click
```
2. Create objects for the following items.
```
        Dim board As New Ektron.Cms.DiscussionBoard
        Dim forumApi As New Ektron.Cms.API.Content.ThreadedDiscussion
        Dim forumID As long
```
3. Check to make sure the Board ID is numeric.
```
        If Not IsNumeric(txtBoardId.Text) Then
            lblResult.Text = "Invalid Board ID"
            Exit Sub
        End If
```
4. Check to make sure the Category ID is numeric.
```
        If Not IsNumeric(txtCategoryId.Text) Then
            lblResult.Text = "Invalid Category ID"
            Exit Sub
        End If
```

5. Create a Try/Catch around the AddForum method. This allows you to catch any exceptions and helps when debugging.

```
        Try
            forumID = forumApi.AddForum(CInt(txtBoardId.Text), txtForumName.Text,
txtDescription.Text, False, False, 1, CInt(txtCategoryId.Text))
```

6. Add an If statement that displays a message stating if the forum was added and its ID.

```
            If (forumID <> 0) Then
                lblResult.Text = "Forum added to the board with Forum ID = " & forumID
            End If
```

7. If there is an error, let the user know what happened.

```
        Catch ex As Exception
            lblResult.Text = ex.Message
        End Try

    End Sub
```

## AddReply Method

This is the overview for the AddReply method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | ThreadedDiscussion.AddReply (Long, String, Long, Integer, Boolean, Boolean, String, FileAttachment) (see page 852) | Adds a Post to the Forum (see page 256). You can specify a post to reply to. You can post anonymously. |
| | ThreadedDiscussion.AddReply (Long, String, Long, Integer, DateTime, Boolean, Boolean, String, FileAttachment) (see page 853) | Adds a post to a topic. Set PostDate property to backdate a post. You can specify a post to reply to. You can post anonymously. |

## ThreadedDiscussion.AddReply Method (Long, String, Long, Integer, Boolean, Boolean, String, FileAttachment)

Adds a Post to the Forum (see page 256). You can specify a post to reply to. You can post anonymously.

### C#

```
public Long AddReply(Long TopicID, String Message, Long ForUserID, int ReplyToPost, Boolean
Approved, Boolean Anonymous, String IPAddress, FileAttachment FileAttachments);
```

### Visual Basic

```
Public Function AddReply(ByVal TopicID As Long, ByVal Message As String, ByVal ForUserID As
Long, ByVal ReplyToPost As Integer, ByVal Approved As Boolean, ByVal Anonymous As Boolean,
ByVal IPAddress As String, ByVal FileAttachments() As FileAttachment) As Long
```

### Parameters

| Parameters | Description |
|---|---|
| TopicID | The numeric ID of the topic to add the post to. |
| Message | The unencoded HTML of the message. |
| ForUserID | The ID of the user who is posting. |
| ReplyToPost | The ID of the post to which the user is replying. |
| Approved | Set to True to approve the post. Otherwise, the post is set to appending. |
| Anonymous | Set to True to add the post anonymously. Otherwise, the post will be attributed to the logged in user. |
| IPAddress | IP address of the poster. |

| | |
|---|---|
| FileAttachments | Array of file attachment classes. If attaching documents or images to the post. |

**Returns**

Numeric representing the ID of the post that was added. (Long)

### ThreadedDiscussion.AddReply Method (Long, String, Long, Integer, DateTime, Boolean, Boolean, String, FileAttachment)

Adds a post to a topic. Set PostDate property to backdate a post. You can specify a post to reply to. You can post anonymously.

**C#**

```
public Long AddReply(Long TopicID, String Message, Long ForUserID, int ReplyToPost, DateTime
PostDate, Boolean Approved, Boolean Anonymous, String IPAddress, FileAttachment
FileAttachments);
```

**Visual Basic**

```
Public Function AddReply(ByVal TopicID As Long, ByVal Message As String, ByVal ForUserID As
Long, ByVal ReplyToPost As Integer, ByVal PostDate As DateTime, ByVal Approved As Boolean,
ByVal Anonymous As Boolean, ByVal IPAddress As String, ByVal FileAttachments() As
FileAttachment) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| TopicID | The topic to add the post to. |
| Message | The unencoded HTML of the message. |
| ForUserID | The ID of the user who is posting. |
| ReplyToPost | The id of the post to which the user is replying to. |
| PostDate | DateTime value of when the post occurred. |
| Approved | Set to True to approve the post. Otherwise, the post is set to appending. |
| Anonymous | Set to True to add the post anonymously. Otherwise, the post will be attributed to the logged in user. |
| IPAddress | IP address of the poster. |
| FileAttachments | Array of file attachment classes. If attaching documents or images to the post. |

**Returns**

Numeric value representing the ID of the post that was added. (Long)

### AddTopic Method

This is the overview for the AddTopic method overload.

**Description**

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as Long. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | ThreadedDiscussion.AddTopic (Long, String, String) (see page 854) | Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as Long. |
| | ThreadedDiscussion.AddTopic (Long, String, String, Integer) (see page 854) | Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as Long. |

### ThreadedDiscussion.AddTopic Method (Long, String, String)

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as Long.

**C#**

```
public Long AddTopic(Long ForumID, String Title, String Message);
```

**Visual Basic**

```
Public Function AddTopic(ByVal ForumID As Long, ByVal Title As String, ByVal Message As
String) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| ForumID | Numeric ID of forum to which the topic is added. |
| Title | String value of the topic title. |
| Message | String value of the topic message. |

**Returns**

Numeric value for the topic ID. (Long)

### ThreadedDiscussion.AddTopic Method (Long, String, String, Integer)

Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as Long.

**C#**

```
public Long AddTopic(Long ForumID, String Title, String Message, int LanguageId);
```

**Visual Basic**

```
Public Function AddTopic(ByVal ForumID As Long, ByVal Title As String, ByVal Message As
String, ByVal LanguageId As Integer) As Long
```

**Parameters**

| Parameters | Description |
|---|---|
| ForumID | Numeric ID of forum to which the topic is added. |
| Title | String value of the topic title. |
| Message | String value of the topic message. |
| LanguageId | Numeric ID of the Language. |

**Returns**

Numeric value for the topic ID. (Long)

### ThreadedDiscussion.DeleteBoard Method

Deletes a board with a given ID. This deletes the board as well as all forums, topics and posts under it. BoardID must be greater than 0.

**C#**

```
public DeleteBoard(Long BoardID);
```

**Visual Basic**

```
Public Sub DeleteBoard(ByVal BoardID As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| BoardID | The numeric ID of the board. This corresponds to the folder ID. |

## ThreadedDiscussion.DeleteForum Method

Deletes a forum with a given ID. This deletes the forum as well as all topics and posts under it. forumID must be greater than 0 (zero).

**C#**

```
public DeleteForum(Long forumID);
```

**Visual Basic**

```
Public Sub DeleteForum(ByVal forumID As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| forumID | The numeric ID of the forum. This corresponds to the folder ID. |

## ThreadedDiscussion.DeleteTopic Method

Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value.

**C#**

```
public Boolean DeleteTopic(Long topicid);
```

**Visual Basic**

```
Public Function DeleteTopic(ByVal topicid As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| topicid | Numeric ID of the topic to delete. |

**Returns**

Boolean

## GetActiveTopics Method

This is the overview for the GetActiveTopics method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | ThreadedDiscussion.GetActiveTopics (Long, Integer) (see page 855) | Gets Most active topics for a board. Activity is measured by number of posts on the topic. |
| | ThreadedDiscussion.GetActiveTopics (Long, Integer, TimeSpan) (see page 856) | Gets Most active topics for a board. Activity is measured by number of posts on the topic. |

## ThreadedDiscussion.GetActiveTopics Method (Long, Integer)

Gets Most active topics for a board. Activity is measured by number of posts on the topic.

**C#**

```
public DiscussionTopic[] GetActiveTopics(Long boardId, int MaxNumber);
```

**Visual Basic**

```
Public Function GetActiveTopics(ByVal boardId As Long, ByVal MaxNumber As Integer) As
DiscussionTopic()
```

**Parameters**

| Parameters | Description |
|---|---|
| boardId | The numeric ID value of the discussion board. |
| MaxNumber | The maximum number of topics to retrieve. |

**Returns**

An array of Discussion Topic classes.

### ThreadedDiscussion.GetActiveTopics Method (Long, Integer, TimeSpan)

Gets Most active topics for a board. Activity is measured by number of posts on the topic.

**C#**

```
public DiscussionTopic[] GetActiveTopics(Long boardId, int MaxNumber, TimeSpan timeSpan);
```

**Visual Basic**

```
Public Function GetActiveTopics(ByVal boardId As Long, ByVal MaxNumber As Integer, ByVal
timeSpan As TimeSpan) As DiscussionTopic()
```

**Parameters**

| Parameters | Description |
|---|---|
| boardId | The numeric representing the board ID. |
| MaxNumber | The maximum number of topics to retrieve. |

**Returns**

An array of Discussion Topic classes.

### ThreadedDiscussion.GetActiveTopicsForForum Method

Gets Most active topics for a forum. Activity is measured by number of posts on the topic.

**C#**

```
public DiscussionTopic[] GetActiveTopicsForForum(Long forumId, int MaxNumber);
```

**Visual Basic**

```
Public Function GetActiveTopicsForForum(ByVal forumId As Long, ByVal MaxNumber As Integer) As
DiscussionTopic()
```

**Parameters**

| Parameters | Description |
|---|---|
| forumId | The numeric ID value of the forum. |
| MaxNumber | The maximum number of topics to retrieve. |

**Returns**

An array of Discussion Topic classes.

### ThreadedDiscussion.GetActiveUsers Method

Gets Most active topics for a board. Activity is measured by number of posts on the topic.

**C#**

```
public UserData[] GetActiveUsers(Long boardId, int MaxNumber, TimeSpan timeSpan);
```

**Visual Basic**

```
Public Function GetActiveUsers(ByVal boardId As Long, ByVal MaxNumber As Integer, ByVal
timeSpan As TimeSpan) As UserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| boardId | The numeric value representing the board ID. |
| MaxNumber | The maximum number of topics to retrieve. |

**Returns**

An array of Discussion Topic classes.

### ThreadedDiscussion.GetBoardCategories Method

Method to get the Categories for a board by ID. Returns either zero length or populated array. See DiscussionCategory class for more information.

**C#**

```
public DiscussionCategory[] GetBoardCategories(Long boardID);
```

**Visual Basic**

```
Public Function GetBoardCategories(ByVal boardID As Long) As DiscussionCategory()
```

**Parameters**

| Parameters | Description |
|---|---|
| boardID | Numeric ID of the discussion board. |

**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(Long boardID);
```

**Visual Basic**

```
Public Function GetBoardForums(ByVal boardID As Long) As DiscussionForum()
```

**Parameters**

| Parameters | Description |
|---|---|
| boardID | Numeric value of the board ID. |

**Returns**

Returns an array of type DiscussionForum.

### GetDiscussionBoard Method

This is the overview for the GetDiscussionBoard method overload.

**Description**

Method to get a discussion board by ID or path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

**Overload List**

| | Name | Description |
|---|---|---|
| | ThreadedDiscussion.GetDiscussionBoard (Long) (see page 858) | 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 858) | Method to get a discussion board by using its path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object. |

## ThreadedDiscussion.GetDiscussionBoard Method (Long)

Method to get a board by ID. Returns a DiscussionBoard class for a given ID value. See DiscussionBoard class for more information on the properties of the returned object.

**C#**

```
public DiscussionBoard GetDiscussionBoard(Long boardID);
```

**Visual Basic**

```
Public Function GetDiscussionBoard(ByVal boardID As Long) As DiscussionBoard
```

**Parameters**

| Parameters | Description |
|---|---|
| boardID | Numeric value of the board ID. |

**Returns**

Returns a class of type DiscussionBoard.

## ThreadedDiscussion.GetDiscussionBoard Method (String)

Method to get a discussion board by using its path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

**C#**

```
public DiscussionBoard GetDiscussionBoard(String boardpath);
```

**Visual Basic**

```
Public Function GetDiscussionBoard(ByVal boardpath As String) As DiscussionBoard
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 cascading style sheets (CSS).

**C#**

```
public String GetDiscussionBoardString(Long boardID, String URLpath);
```

**Visual Basic**

```
Public Function GetDiscussionBoardString(ByVal boardID As Long, ByVal URLpath As String) As
String
```

**Parameters**

| Parameters | Description |
|---|---|
| boardID | Numeric ID of the board. |
| 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. Returns a discussion board with a 1 length Discussion Forum (see page 256) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.

*See Also:* The Example section.

**C#**

```
public DiscussionBoard GetForum(Long forumid);
```

**Visual Basic**

```
Public Function GetForum(ByVal forumid As Long) As DiscussionBoard
```

**Parameters**

| Parameters | Description |
|---|---|
| forumid | Numeric value of the forum ID. |

**Returns**

An instance of type DiscussionBoard.

**Example**

The following example shows how to create a Web page from which forum data can be retrieved. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetForum method. This method uses the InternalAdmin to retrieve the forum data. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a label with the following information.
   ```
   <asp:Label ID="Label1" runat="server" Text="Forum ID: "></asp:Label>
   ```
2. Add a text box to enter the Forum ID.
   ```
   <asp:TextBox ID="txtForumId" runat="server" Width="63px"></asp:TextBox>
   ```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Forum ID Box.
   ```
   <asp:RequiredFieldValidator ID="rfvForumId" runat="server" ControlToValidate="txtForumId"
      ErrorMessage="Please enter forum ID"></asp:RequiredFieldValidator>
   ```
4. Add a couple of breaks for spacing purposes.
   ```
   <br /><br />
   ```
5. Add a button, to retrieve the forum data.
   ```
   <asp:Button ID="btnGetForum" runat="server" Text="Get Forum" Width="72px" />
   ```
6. Add a couple of more breaks for spacing purposes.
   ```
   <br /><br />
   ```
7. Add a results label where the forum data will appear. This label is also used to display any errors that might occur.
   ```
   <asp:Label ID="lblResult" runat="server" Width="604px" Height="113px"></asp:Label>
   ```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.

```
        Protected Sub btnGetForum_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetForum.Click
```

2. Create objects for the following items.

```
        Dim board As New Ektron.Cms.DiscussionBoard
        Dim forumApi As New Ektron.Cms.API.Content.ThreadedDiscussion
```

3. Check to make sure the forum ID is numeric.

```
        If Not IsNumeric(txtForumId.Text) Then
            lblResult.Text = "Invalid Forum ID"
            Exit Sub
        End If
```

4. Create a Try/Catch around the GetForum method. This allows you to catch any exceptions and helps when debugging.

```
        Try
            board = forumApi.GetForum(CInt(txtForumId.Text))
```

5. Check to make sure the discussion board and forum are valid.

```
            If (board.Id = 0) Or (board.Forums.Length <> 1) Then
                lblResult.Text = "Error getting forum"
                Exit Sub
            End If
```

6. Display the Discussion Board data.

```
            lblResult.Text = "Board Title: " & board.Title & "<br/>"
            lblResult.Text &= "Board Name: " & board.Name & "<br/>"
```

7. Display the Forum (see page 256) data.

```
            lblResult.Text &= "Forum Category: " & board.Forums(0).CategoryName & "<br/>"
            lblResult.Text &= "Forum Name: " & board.Forums(0).ForumName & "<br/>"
            lblResult.Text &= "Forum Title: " & board.Forums(0).ForumTitle & "<br/>"
            lblResult.Text &= "Number of topics: " & board.Forums(0).NumberofTopics &
"<br/>"
            lblResult.Text &= "Number of posts: " & board.Forums(0).NumberofPosts & "<br/>"
```

8. If there is an error, let the user know what happened.

```
        Catch ex As Exception
            lblResult.Text = ex.Message
        End Try

    End Sub
```

### ThreadedDiscussion.GetRepliesForTopics Method

Method returns a list of replies for a given forum ID.

- ArrayList Structure ArrayListItem(j)(0)
- content_id ArrayListItem(j)(1)
- comment_total ArrayListItem(j)(2)
- comment_sum ArrayListItem(j)(3)
- last_posted ArrayListItem(j)(4)
- content_language

**C#**

```
public ArrayList GetRepliesForTopics(Long ForumID);
```

**Visual Basic**

```
Public Function GetRepliesForTopics(ByVal ForumID As Long) As ArrayList
```

**Parameters**

| Parameters | Description |
|---|---|
| ForumID | Numeric value of the forum ID. |

**Returns**

An instance of type ArrayList.

**Remarks**

This is an older method and is deprecated. The information returned herein was merged into the higher performance GetForum (see page 859)(int forumID) method, and that should be used instead.

### ThreadedDiscussion.GetTopic Method

Method to obtain a particular Topic by ID. Returns a discussion board with a 1 length Discussion Forum (see page 256) array containing a 1 length DiscussionTopic array. If the topic does not exist, then a DiscussionBoard with a 1 length Forums property containing a zero length Topics property will be returned.

**C#**

```
public DiscussionBoard GetTopic(Long topicid);
```

**Visual Basic**

```
Public Function GetTopic(ByVal topicid As Long) As DiscussionBoard
```

**Parameters**

| Parameters | Description |
|---|---|
| topicid | Numeric ID of the Topic. |

**Returns**

An instance of type DiscussionBoard.

### ThreadedDiscussion.SearchReplies Method

Method to obtain an array of topic replies matching search criteria. Returns an instance of EkTasks, which is effectively an array of EkTask objects. If no results are returned, then a zero length EkTasks will be returned.

**C#**

```
public EkTasks SearchReplies(Long ForumIDs, String SearchTerms, EkEnumeration.SearchTypes
stType);
```

**Visual Basic**

```
Public Function SearchReplies(ByVal ForumIDs() As Long, ByVal SearchTerms() As String, ByVal
stType As EkEnumeration.SearchTypes) As EkTasks
```

**Parameters**

| Parameters | Description |
|---|---|
| ForumIDs | Numeric (Long) array of the Forum (see page 256) 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).

### ThreadedDiscussion.SubscribeToBoard Method

Subscribe a user to a board by providing the ID of the board, the user's ID and the type of subscription.

**C#**

```
public ObjectWatch SubscribeToBoard(Long BoardID, Long ForUserID,
EkEnumeration.NotificationType WatchType);
```

**Visual Basic**

```
Public Function SubscribeToBoard(ByVal BoardID As Long, ByVal ForUserID As Long, ByVal
WatchType As EkEnumeration.NotificationType) As ObjectWatch
```

**Parameters**

| Parameters | Description |
|---|---|
| BoardID | Numeric ID of the discussion board to subscribe to. |
| ForUserID | Numeric ID of the user subscribing. |
| WatchType | Type of subscription. Choices are:<br>• **All** - subscribes a user to all discussion board items.<br>• **Replies** - subscribes a user to all replies to his posts in the specified discussion board.<br>• **Never** - unsubscribes a user from the discussion board. |

**Returns**

Object watch of the current subscription status.

### ThreadedDiscussion.SubscribeToForum Method

Subscribe a user to a forum by providing the ID of the forum, the user's ID and the type of subscription.

**C#**

```
public ObjectWatch SubscribeToForum(Long ForumID, Long ForUserID,
EkEnumeration.NotificationType WatchType);
```

**Visual Basic**

```
Public Function SubscribeToForum(ByVal ForumID As Long, ByVal ForUserID As Long, ByVal
WatchType As EkEnumeration.NotificationType) As ObjectWatch
```

**Parameters**

| Parameters | Description |
|---|---|
| ForumID | Numeric ID of the forum to subscribe to. |
| ForUserID | Numeric ID of the user subscribing. |
| WatchType | Type of subscription. Choices are:<br>• **All** - subscribes a user to all forum items.<br>• **Replies** - subscribes a user to all replies to his posts in the specified forum.<br>• **Never** - unsubscribes a user from the forum. |

**Returns**

Object watch of the current subscription status.

### ThreadedDiscussion.SubscribeToTopic Method

Subscribe a user to a topic by providing the ID of the topic, the user's ID and the type of subscription.

**C#**

```
public ObjectWatch SubscribeToTopic(Long TopicID, Long ForUserID,
EkEnumeration.NotificationType WatchType);
```

**Visual Basic**

```
Public Function SubscribeToTopic(ByVal TopicID As Long, ByVal ForUserID As Long, ByVal
WatchType As EkEnumeration.NotificationType) As ObjectWatch
```

**Parameters**

| Parameters | Description |
|---|---|
| TopicID | Numeric ID of the topic to subscribe to. |
| ForUserID | Numeric ID of the user subscribing. |
| WatchType | Type of subscription. Choices are:<br>• **All** - subscribes a user to all topics.<br>• **Replies** - subscribes a user to any replies to the topic he created.<br>• **Never** - unsubscribes a user from the topic. |

**Returns**

Object watch of the current subscription status.

### ThreadedDiscussion.UnSubscribeToBoard Method

Unsubscribe a user from a discussion board by providing the discussion board's ID and user ID.

**C#**

```
public ObjectWatch UnSubscribeToBoard(Long BoardID, Long ForUserID);
```

**Visual Basic**

```
Public Function UnSubscribeToBoard(ByVal BoardID As Long, ByVal ForUserID As Long) As
ObjectWatch
```

**Parameters**

| Parameters | Description |
|---|---|
| BoardID | Numeric ID of the discussion board to unsubscribe from. |
| ForUserID | Numeric ID of the user to unsubscribe. |

**Returns**

Object watch of the current subscription status.

### ThreadedDiscussion.UnSubscribeToForum Method

Unsubscribe a user from a forum by providing the topic ID and user ID.

**C#**

```
public ObjectWatch UnSubscribeToForum(Long ForumID, Long ForUserID);
```

**Visual Basic**

```
Public Function UnSubscribeToForum(ByVal ForumID As Long, ByVal ForUserID As Long) As
ObjectWatch
```

**Parameters**

| Parameters | Description |
|---|---|
| ForumID | Numeric ID of the forum to unsubscribe from. |
| ForUserID | Numeric ID of the user to unsubscribe. |

**Returns**

Object watch of the current subscription status.

### ThreadedDiscussion.UnSubscribeToTopic Method

Unsubscribe a user from a topic by providing the topic ID and user ID.

**C#**

```
public ObjectWatch UnSubscribeToTopic(Long TopicID, Long ForUserID);
```

**Visual Basic**

```
Public Function UnSubscribeToTopic(ByVal TopicID As Long, ByVal ForUserID As Long) As
ObjectWatch
```

**Parameters**

| Parameters | Description |
|---|---|
| TopicID | Numeric ID of the topic to unsubscribe from. |
| ForUserID | Numeric ID of the user unsubscribing. |

**Returns**

Object watch of the current subscription status.

# Ektron.Cms.API.Search Namespace

The Search Developer API (see page 639) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 694), if you chose to install it, has excellent examples of programmatically using the SearchManager (see page 865). It is installed at http://~YourSite~/CMS400Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#.

Here is a list of some of the examples.

- Retrieve a document or HTML content that contains the word *cms*.
- Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer.
- Retrieve images that contain *RotatingGraphic1* in their title.
- Retrieve geo-content from CMS400.Net.

**Classes**

| | Name | Description |
|---|---|---|
| | SearchManager (see page 865) | The Search (see page 868) Developer API (see page 639) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 694), if you chose to install it, has excellent examples of programmatically using the SearchManager. It is installed at http://~YourSite~/CMS400Developer/Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#.<br>Here is a list of some of the examples.<br>• Retrieve a document or HTML content that contains the word *cms*.<br>• Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer.<br>• Retrieve images that contain *RotatingGraphic1* in their title.<br>• Retrieve geo-content from CMS400.Net. |

# Classes

The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | SearchManager (see page 865) | The Search (see page 868) Developer API (see page 639) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 694), if you chose to install it, has excellent examples of programmatically using the SearchManager. It is installed at http://~YourSite~/CMS400Developer/Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#.<br>Here is a list of some of the examples.<br>• Retrieve a document or HTML content that contains the word *cms*.<br>• Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer.<br>• Retrieve images that contain *RotatingGraphic1* in their title.<br>• Retrieve geo-content from CMS400.Net. |

## SearchManager Class

The Search (see page 868) Developer API (see page 639) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 694), if you chose to install it, has excellent examples of programmatically using the SearchManager. It is installed at http://~YourSite~/CMS400Developer/Developer/Websearch/WebsearchApi.aspx. Examples are in VB and C#.

Here is a list of some of the examples.

- Retrieve a document or HTML content that contains the word *cms*.
- Retrieve content that contains *research* and has a searchable metadata *wellness* set to Cancer.
- Retrieve images that contain *RotatingGraphic1* in their title.
- Retrieve geo-content from CMS400.Net.

### Class Hierarchy

### C#

```
public class SearchManager : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class SearchManager
Inherits Ektron.Cms.API.Common
```

### File

SearchManager.vb

## SearchManager Methods

The methods of the SearchManager class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Execute (see page 866) | This is the overview for the Execute method overload. |
| | Search (see page 867) | This is the overview for the Search method overload. |

| | | |
|---|---|---|
| | SearchForCategories (see page 870) | Search (see page 868) based on taxonomy category. |
| | SearchTable (see page 870) | This API (see page 639) is used internally by the Ektron Windows Service to update catalogs when a taxonomy category is updated<br>This ensures that subsequent searches function correctly. |

## Execute Method

This is the overview for the Execute method overload.

### Description

Returns the array of ContentData. This method is overloaded.

### Overload List

| | Name | Description |
|---|---|---|
| | SearchManager.Execute (Ektron.Cms.ContentSearchCondition) (see page 866) | Returns the array of ContentData (see page 1378) based on the criteria ContentSearchCondition. |
| | SearchManager.Execute (Ektron.Cms.UserSearchCondition) (see page 867) | Returns the array of UserData (see page 1419) based on the criteria UserSearchCondition. |

## SearchManager.Execute Method (Ektron.Cms.ContentSearchCondition)

Returns the array of ContentData (see page 1378) 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 (see page 1378)

### 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 (see page 1419) 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 (see page 1419)

**Example**

```
Dim isMemberShip As UserSearchCondition = New UserSearchCondition
isMemberShip.setType = SearchType.EQUAL
isMemberShip.setValue = 1 'The value can be integer,string,date and boolean should be match
with db type
isMemberShip.setVariable = users.membership_user

Dim isInEktron As UserSearchCondition = New UserSearchCondition
isInEktron.setType = SearchType.EQUAL
isInEktron.setValue = "03031"  'The value can be integer,string,date and boolean should be
match with dynamic_data_tbl labels type
isInEktron.setVariable = "customproperties.zip code"

Dim condition As UserSearchCondition = New UserSearchCondition
condition.setType = SearchType.AND
condition.AddCondition(isInEktron)
condition.AddCondition(isMemberShip)

Dim search As New SearchManager
Dim result As userdata() = search.Execute(condition)
```

### Search Method

This is the overview for the Search method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | SearchManager.Search (IndexSearchRequestData, HttpContext, Integer) (see page 868) | Search smart form content based on indexed fields.<br>The IndexSearchFieldList property (an array of IndexSearchFieldData) of the IndexSearchRequestData object is used to specify the XPATH and their values. |
| | SearchManager.Search (MapRequestData, HttpContext, Integer, Integer, Integer) (see page 868) | Map (see page 340) search for retrieving geographical content from CMS400.Net site.<br>This function has two additional overloads. |

| | | |
|---|---|---|
| | SearchManager.Search (SearchRequestData, HttpContext, Integer) (see page 869) | Search HTML, content, images, multimedia and documents based on content or metadata.<br>This function has two additional overloads. |
| | SearchManager.Search (SearchRequestData, HttpContext, Integer, DataTable) (see page 869) | This is Search, a member of class SearchManager. |
| | SearchManager.Search (String, HttpContext, Integer) (see page 869) | Search the site for occurrence(s) of search text.<br>This function has two additional overloads. |

### SearchManager.Search Method (IndexSearchRequestData, HttpContext, Integer)

Search smart form content based on indexed fields.

The IndexSearchFieldList property (an array of IndexSearchFieldData) of the IndexSearchRequestData object is used to specify the XPATH and their values.

**C#**

```
public SearchResponseData[] Search(IndexSearchRequestData indexSearchReqData, HttpContext
context, ref int resultCount);
```

**Visual Basic**

```
Public Function Search(ByVal indexSearchReqData As IndexSearchRequestData, ByVal context As
HttpContext, ByRef resultCount As Integer) As SearchResponseData()
```

**Parameters**

| Parameters | Description |
|---|---|
| indexSearchReqData | Search criteria bundled in a IndexSearchRequestData object. |
| context | The current Http request. |
| resultCount | Number of records returned. |

**Returns**

Array of SearchResponseData - content satisfying search criteria

### SearchManager.Search Method (MapRequestData, HttpContext, Integer, Integer, Integer)

Map (see page 340) search for retrieving geographical content from CMS400.Net site.

This function has two additional overloads.

**C#**

```
public MapResponseData[] Search(MapRequestData mapReqData, HttpContext context, int index, int
count, ref int resultCount);
```

**Visual Basic**

```
Public Function Search(ByVal mapReqData As MapRequestData, ByVal context As HttpContext, ByVal
index As Integer, ByVal count As Integer, ByRef resultCount As Integer) As MapResponseData()
```

**Parameters**

| Parameters | Description |
|---|---|
| mapReqData | Search criteria bundled in a MapRequestData object. |
| context | The current Http request. |
| index | Page number. |

| | |
|---|---|
| count | Page size. |
| resultCount | Number of records returned. |

**Returns**

Array of MapResponseData - content satisfying location criteria

### SearchManager.Search Method (SearchRequestData, HttpContext, Integer)

Search HTML, content, images, multimedia and documents based on content or metadata.

This function has two additional overloads.

**C#**

```
public SearchResponseData[] Search(SearchRequestData searchReqData, HttpContext context, ref
int resultCount);
```

**Visual Basic**

```
Public Function Search(ByVal searchReqData As SearchRequestData, ByVal context As HttpContext,
ByRef resultCount As Integer) As SearchResponseData()
```

**Parameters**

| Parameters | Description |
|---|---|
| searchReqData | Search criteria bundled in a SearchRequestData object. |
| context | The current Http request. |
| resultCount | Number of records returned. |

**Returns**

Array of SearchResponseData - content satisfying search criteria

### SearchManager.Search Method (SearchRequestData, HttpContext, Integer, DataTable)

**C#**

```
public SearchResponseData[] Search(SearchRequestData searchReqData, HttpContext context, ref
int resultCount, ref DataTable suggestedResultTable);
```

**Visual Basic**

```
Public Function Search(ByVal searchReqData As SearchRequestData, ByVal context As HttpContext,
ByRef resultCount As Integer, ByRef suggestedResultTable As DataTable) As SearchResponseData()
```

**Description**

This is Search, a member of class SearchManager.

### SearchManager.Search Method (String, HttpContext, Integer)

Search the site for occurrence(s) of search text.

This function has two additional overloads.

**C#**

```
public SearchResponseData[] Search(String searchText, HttpContext context, ref int
resultCount);
```

**Visual Basic**

```
Public Function Search(ByVal searchText As String, ByVal context As HttpContext, ByRef
resultCount As Integer) As SearchResponseData()
```

### Parameters

| Parameters | Description |
|---|---|
| searchText | The text for which you are searching. |
| context | The current Http request. |
| resultCount | Number of records returned. |

### Returns

Array of SearchResponseData - content satisfying search criteria

## SearchManager.SearchForCategories Method

Search (see page 868) based on taxonomy category.

### C#

```
public String[] SearchForCategories(SearchRequestData searchReqData, HttpContext context, int index, int count, ref int resultCount);
```

### Visual Basic

```
Public Function SearchForCategories(ByVal searchReqData As SearchRequestData, ByVal context As HttpContext, ByVal index As Integer, ByVal count As Integer, ByRef resultCount As Integer) As String()
```

### Parameters

| Parameters | Description |
|---|---|
| searchReqData | Search (see page 868) criteria bundled in a SearchRequestData object. |
| context | The current Http request. |
| index | Page number. |
| count | Page size. |
| resultCount | Number of records returned. |

### Returns

Array of String - List of categories

## SearchManager.SearchTable Method

This API (see page 639) is used internally by the Ektron Windows Service to update catalogs when a taxonomy category is updated

This ensures that subsequent searches function correctly.

### C#

```
public DataSet SearchTable(SearchRequestData searchReqData, String searchParams, HttpContext context, ref int resultCount);
```

### Visual Basic

```
Public Function SearchTable(ByVal searchReqData As SearchRequestData, ByVal searchParams As String, ByVal context As HttpContext, ByRef resultCount As Integer) As DataSet
```

### Parameters

| Parameters | Description |
|---|---|
| searchReqData | Search (see page 868) criteria bundled in a SearchRequestData object. |
| searchParams | Database columns to be included in search. |
| context | The current Http request. |
| resultCount | Number of records returned. |

### Returns

DataSet with a single DataTable which contains content whose taxonomy was updated.

# Ektron.Cms.API.Task Namespace

This is namespace Ektron.Cms.API.Task.

### Classes

| | Name | Description |
|---|---|---|
| | Task (see page 871) | This is class Ektron.Cms.API.Task.Task. |
| | TaskCategory (see page 882) | This is class Ektron.Cms.API.Task.TaskCategory. |
| | TaskCategoryType (see page 884) | This is class Ektron.Cms.API.Task.TaskCategoryType. |

## Classes

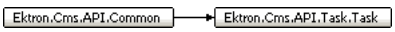
The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | Task (see page 871) | This is class Ektron.Cms.API.Task.Task. |
| | TaskCategory (see page 882) | This is class Ektron.Cms.API.Task.TaskCategory. |
| | TaskCategoryType (see page 884) | This is class Ektron.Cms.API.Task.TaskCategoryType. |

## Task Class

### Class Hierarchy




### C#

```
public class Task : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class Task
Inherits Ektron.Cms.API.Common
```

### File

Task.vb

### Description

This is class Ektron.Cms.API.Task.Task.

### Task Methods

The methods of the Task class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddTask (see page 872) | Adds a task in the system. |

| | | |
|---|---|---|
| | AssignTaskToUser (see page 872) | This is the overview for the AssignTaskToUser method overload. |
| | AssignTaskToUsergroup (see page 873) | This is the overview for the AssignTaskToUsergroup method overload. |
| | DeleteTask (see page 874) | This is the overview for the DeleteTask method overload. |
| | DeleteTasks (see page 875) | This is the overview for the DeleteTasks method overload. |
| | GetTask (see page 876) | Get the task information. |
| | GetTasks (see page 876) | This is the overview for the GetTasks method overload. |
| | GetTasksAssignedByUser (see page 877) | This is the overview for the GetTasksAssignedByUser method overload. |
| | GetTasksAssignedToUser (see page 878) | This is the overview for the GetTasksAssignedToUser method overload. |
| | GetTasksAssignedToUsergroup (see page 879) | This is the overview for the GetTasksAssignedToUsergroup method overload. |
| | GetTaskState (see page 881) | Get the task state. |
| | SetStateForTask (see page 881) | Set the task state. |
| | UpdateTask (see page 881) | Updates the task |

## Task.AddTask Method

Adds a task in the system.

**C#**

```
public Boolean AddTask(ref TaskData data);
```

**Visual Basic**

```
Public Function AddTask(ByRef data As TaskData) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| data | TaskData |

**Returns**

Returns False if successfully added, otherwise returns True.

## AssignTaskToUser Method

This is the overview for the AssignTaskToUser method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.AssignTaskToUser (Long, Long) (see page 872) | Assign a task to a user. |
| | Task.AssignTaskToUser (Long, Long, Long) (see page 873) | Assign a task to a user by a specified user. |

## Task.AssignTaskToUser Method (Long, Long)

Assign a task to a user.

**C#**

```
public Boolean AssignTaskToUser(Long taskId, Long assignToUserId);
```

**Visual Basic**

```
Public Function AssignTaskToUser(ByVal taskId As Long, ByVal assignToUserId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 871) ID |
| assignToUserId | User (see page 913)'s ID |

**Returns**

Returns False if the task is successfully assigned to a user. Otherwise, it returns True.

## Task.AssignTaskToUser Method (Long, Long, Long)

Assign a task to a user by a specified user.

**C#**

```
public Boolean AssignTaskToUser(Long taskId, Long assignToUserId, Long assignByUserId);
```

**Visual Basic**

```
Public Function AssignTaskToUser(ByVal taskId As Long, ByVal assignToUserId As Long, ByVal
assignByUserId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 871) ID |
| assignToUserId | Assign to User (see page 913)'s ID |
| assignByUserId | Assign by User (see page 913)'s ID |

**Returns**

Returns False if the task is successfully assigned to a user. Otherwise, it returns True.

## AssignTaskToUsergroup Method

This is the overview for the AssignTaskToUsergroup method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.AssignTaskToUsergroup (Long, Long) (see page 873) | Assign a task to a usergroup. |
| | Task.AssignTaskToUsergroup (Long, Long, Long) (see page 874) | Assign a task to a usergroup by a specified user. |

## Task.AssignTaskToUsergroup Method (Long, Long)

Assign a task to a usergroup.

**C#**

```
public Boolean AssignTaskToUsergroup(Long taskId, Long assignToUsergroup);
```

**Visual Basic**

```
Public Function AssignTaskToUsergroup(ByVal taskId As Long, ByVal assignToUsergroup As Long)
As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 871) ID |
| assignToUsergroup | Assign to Usergroup's ID |

**Returns**

Returns False if the task is successfully assigned to a usergroup. Otherwise, it returns True.

## Task.AssignTaskToUsergroup Method (Long, Long, Long)

Assign a task to a usergroup by a specified user.

**C#**

```
public Boolean AssignTaskToUsergroup(Long taskId, Long assignToUsergroup, Long assignByUserId);
```

**Visual Basic**

```
Public Function AssignTaskToUsergroup(ByVal taskId As Long, ByVal assignToUsergroup As Long,
ByVal assignByUserId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 871) ID |
| assignToUsergroup | Numeric ID of the Usergroup to which the task is being assigned. |
| assignByUserId | Numeric ID of the User (see page 913) who's assigning the task. |

**Returns**

Returns False if the task is successfully assigned to a usergroup. Otherwise, it returns True.

## DeleteTask Method

This is the overview for the DeleteTask method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.DeleteTask (Long) (see page 874) | Mark the task deleted. |
| | Task.DeleteTask (Long, Boolean) (see page 875) | Mark the task deleted or delete the task permanently. |

## Task.DeleteTask Method (Long)

Mark the task deleted.

**C#**

```
public Boolean DeleteTask(Long taskId);
```

**Visual Basic**

```
Public Function DeleteTask(ByVal taskId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 871) ID. |

**Returns**

Returns False if successfully deleted, otherwise returns True.

## Task.DeleteTask Method (Long, Boolean)

Mark the task deleted or delete the task permanently.

**C#**

```
public Boolean DeleteTask(Long taskId, Boolean Permanently);
```

**Visual Basic**

```
Public Function DeleteTask(ByVal taskId As Long, ByVal Permanently As Boolean) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 871) ID. |
| Permanently | True=Delete the task. |

**Returns**

Returns False if successfully deleted otherwise returns True.

## DeleteTasks Method

This is the overview for the DeleteTasks method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.DeleteTasks (Long) (see page 875) | Mark tasks deleted. |
| | Task.DeleteTasks (Long, Boolean) (see page 876) | Mark tasks deleted or delete tasks permanently. |

## Task.DeleteTasks Method (Long)

Mark tasks deleted.

**C#**

```
public Boolean DeleteTasks(Long taskId);
```

**Visual Basic**

```
Public Function DeleteTasks(ByVal taskId() As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Array of Task (see page 871) ID. |

**Returns**

Returns False if successfully deleted, otherwise returns True.

**Remarks**

When an error occurs the method will quit without deleting rest of the tasks.

## Task.DeleteTasks Method (Long, Boolean)

Mark tasks deleted or delete tasks permanently.

**C#**

```
public Boolean DeleteTasks(Long taskId, Boolean Permanently);
```

**Visual Basic**

```
Public Function DeleteTasks(ByVal taskId() As Long, ByVal Permanently As Boolean) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Array of Task (see page 871) ID. |
| Permanently | True=Delete tasks. |

**Returns**

Returns False if successfully deleted, otherwise returns True.

**Remarks**

When an error occurs the method will quit without deleting rest of the tasks.

## Task.GetTask Method

Get the task information.

**C#**

```
public TaskData GetTask(Long taskId);
```

**Visual Basic**

```
Public Function GetTask(ByVal taskId As Long) As TaskData
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | The task's numeric ID. |

**Returns**

TaskData

## GetTasks Method

This is the overview for the GetTasks method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.GetTasks () (see page 876) | Get All tasks from the system. |
| | Task.GetTasks (Long) (see page 877) | Get All tasks from the system. |

## Task.GetTasks Method ()

Get All tasks from the system.

**C#**

```
public TaskData[] GetTasks();
```

**Visual Basic**

```
Public Function GetTasks() As TaskData()
```

**Returns**

Array of task data.

### Task.GetTasks Method (Long)

Get All tasks from the system.

**C#**

```
public TaskData[] GetTasks(Long folderId);
```

**Visual Basic**

```
Public Function GetTasks(ByVal folderId() As Long) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| folderId | Array of FolderIDs |

**Returns**

Array of task data.

### GetTasksAssignedByUser Method

This is the overview for the GetTasksAssignedByUser method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.GetTasksAssignedByUser (Long) (see page 877) | Get all tasks assigned by the user. |
| | Task.GetTasksAssignedByUser (Long, EkEnumeration.TaskState) (see page 878) | Get all tasks assign by a user with a specified task state and returns them as array of TaskData. |

### Task.GetTasksAssignedByUser Method (Long)

Get all tasks assigned by the user.

**C#**

```
public TaskData[] GetTasksAssignedByUser(Long userId);
```

**Visual Basic**

```
Public Function GetTasksAssignedByUser(ByVal userId As Long) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | Numeric value of the user's ID. |

**Returns**

Array of tasks.

## Task.GetTasksAssignedByUser Method (Long, EkEnumeration.TaskState)

Get all tasks assign by a user with a specified task state and returns them as array of TaskData.

**C#**

```
public TaskData[] GetTasksAssignedByUser(Long userId, EkEnumeration.TaskState taskState);
```

**Visual Basic**

```
Public Function GetTasksAssignedByUser(ByVal userId As Long, ByVal taskState As
EkEnumeration.TaskState) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | Numeric value of the user's ID. |
| taskState | The state of the task. Choices are:<br>• NotStarted<br>• Active<br>• AwaitingData<br>• OnHold<br>• Pending<br>• Reopened<br>• Completed<br>• Archived<br>• Deleted |

**Returns**

Array of tasks.

**Description**

## GetTasksAssignedToUser Method

This is the overview for the GetTasksAssignedToUser method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.GetTasksAssignedToUser (Long) (see page 878) | Get all tasks assigned to the user and returns them as array of TaskData. |
| | Task.GetTasksAssignedToUser (Long, EkEnumeration.TaskState) (see page 879) | Get all tasks assigned to the user by a specified task state and returns them as array of TaskData. |

## Task.GetTasksAssignedToUser Method (Long)

Get all tasks assigned to the user and returns them as array of TaskData.

**C#**

```
public TaskData[] GetTasksAssignedToUser(Long userId);
```

**Visual Basic**

```
Public Function GetTasksAssignedToUser(ByVal userId As Long) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The numeric value of the User (see page 913) ID. |

**Returns**

Returns an array of tasks assigned to the user.

## Task.GetTasksAssignedToUser Method (Long, EkEnumeration.TaskState)

Get all tasks assigned to the user by a specified task state and returns them as array of TaskData.

**C#**

```
public TaskData[] GetTasksAssignedToUser(Long userId, EkEnumeration.TaskState taskState);
```

**Visual Basic**

```
Public Function GetTasksAssignedToUser(ByVal userId As Long, ByVal taskState As
EkEnumeration.TaskState) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | Numeric value of the User (see page 913)'s ID. |
| taskState | The state of the task. Choices are:<br>• NotStarted<br>• Active<br>• AwaitingData<br>• OnHold<br>• Pending<br>• Reopened<br>• Completed<br>• Archived<br>• Deleted |

**Returns**

Returns an array of TaskData.

**Description**

## GetTasksAssignedToUsergroup Method

This is the overview for the GetTasksAssignedToUsergroup method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.GetTasksAssignedToUsergroup (Long) (see page 880) | Get all tasks assigned to the usergroup and returns them as array of TaskData. |

| | | |
|---|---|---|
| | Task.GetTasksAssignedToUsergroup (Long, EkEnumeration.TaskState) (see page 880) | Get all tasks assigned to the usergroup by a specified task state and returns them as array of TaskData. |

### Task.GetTasksAssignedToUsergroup Method (Long)

Get all tasks assigned to the usergroup and returns them as array of TaskData.

**C#**

```
public TaskData[] GetTasksAssignedToUsergroup(Long groupId);
```

**Visual Basic**

```
Public Function GetTasksAssignedToUsergroup(ByVal groupId As Long) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | The numeric value of the usergroup's ID. |

**Returns**

Returns an array of tasks.

### Task.GetTasksAssignedToUsergroup Method (Long, EkEnumeration.TaskState)

Get all tasks assigned to the usergroup by a specified task state and returns them as array of TaskData.

**C#**

```
public TaskData[] GetTasksAssignedToUsergroup(Long groupId, EkEnumeration.TaskState taskState);
```

**Visual Basic**

```
Public Function GetTasksAssignedToUsergroup(ByVal groupId As Long, ByVal taskState As
EkEnumeration.TaskState) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | Numeric value of the Usergroup's ID. |
| taskState | The state of the task. Choices are:<br>• NotStarted<br>• Active<br>• AwaitingData<br>• OnHold<br>• Pending<br>• Reopened<br>• Completed<br>• Archived<br>• Deleted |

**Returns**

Returns an array of tasks.

**Description**

### Task.GetTaskState Method

Get the task state.

**C#**

```
public EkEnumeration.TaskState GetTaskState(Long taskId);
```

**Visual Basic**

```
Public Function GetTaskState(ByVal taskId As Long) As EkEnumeration.TaskState
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 871) ID. |

**Returns**

If successful returns task state. Otherwise, it will throw an error.

### Task.SetStateForTask Method

Set the task state.

**C#**

```
public Boolean SetStateForTask(Long taskId, EkEnumeration.TaskState newTaskState);
```

**Visual Basic**

```
Public Function SetStateForTask(ByVal taskId As Long, ByVal newTaskState As
EkEnumeration.TaskState) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 871) ID |
| newTaskState | New Task (see page 871) State |

**Returns**

Returns False if the state is successfully changed. Otherwise, it returns True.

### Task.UpdateTask Method

Updates the task

**C#**

```
public Boolean UpdateTask(ref TaskData data);
```

**Visual Basic**

```
Public Function UpdateTask(ByRef data As TaskData) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| data | Task (see page 871) Data. |

**Returns**

Returns False if successfully updated, otherwise returns True.

# TaskCategory Class

### Class Hierarchy

### C#

```
public class TaskCategory : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class TaskCategory
Inherits Ektron.Cms.API.Common
```

### File

TaskCategory.vb

### Description

This is class Ektron.Cms.API.Task.TaskCategory.

## TaskCategory Methods

The methods of the TaskCategory class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddTaskCategory (see page 882) | Adds a task category by providing a title. |
| | DeleteTaskCategory (see page 882) | Deletes the task category by providing its numeric ID. |
| | GetAllCategories (see page 883) | Gets All Task (see page 871) categories and all of the task types that belong to those categories. |
| | GetTaskCategory (see page 883) | Gets the specified task category and all the task types that belong to it by providing the numeric ID of the category. |
| | UpdateTaskCategory (see page 883) | Updates the category title by providing its ID and a new title. |

## TaskCategory.AddTaskCategory Method

Adds a task category by providing a title.

### C#

```
public Long AddTaskCategory(String title);
```

### Visual Basic

```
Public Function AddTaskCategory(ByVal title As String) As Long
```

### Parameters

| Parameters | Description |
|---|---|
| title | The new category title. |

### Returns

Returns a numeric ID assigned to the new task Category. The data type is Long.

## TaskCategory.DeleteTaskCategory Method

Deletes the task category by providing its numeric ID.

**C#**

```
public Boolean DeleteTaskCategory(Long categoryId);
```

**Visual Basic**

```
Public Function DeleteTaskCategory(ByVal categoryId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| categoryId | The numeric ID of the task category. |

**Returns**

Returns False if successfully deleted. Otherwise, it returns True.

### TaskCategory.GetAllCategories Method

Gets All Task (see page 871) categories and all of the task types that belong to those categories.

**C#**

```
public TaskCategoryData[] GetAllCategories();
```

**Visual Basic**

```
Public Function GetAllCategories() As TaskCategoryData()
```

**Returns**

Returns an array of TaskCategoryData

### TaskCategory.GetTaskCategory Method

Gets the specified task category and all the task types that belong to it by providing the numeric ID of the category.

**C#**

```
public TaskCategoryData GetTaskCategory(Long categoryID);
```

**Visual Basic**

```
Public Function GetTaskCategory(ByVal categoryID As Long) As TaskCategoryData
```

**Parameters**

| Parameters | Description |
|---|---|
| categoryID | The numeric ID of the category. |

**Returns**

Returns TaskCategoryData.

### TaskCategory.UpdateTaskCategory Method

Updates the category title by providing its ID and a new title.

**C#**

```
public Boolean UpdateTaskCategory(Long categoryId, String title);
```

**Visual Basic**

```
Public Function UpdateTaskCategory(ByVal categoryId As Long, ByVal title As String) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| categoryId | The numeric ID of the Category. |

| | |
|---|---|
| title | The new title of the category. |

### Returns

Returns False if successfully updated. Otherwise, it returns True.

## TaskCategoryType Class

### Class Hierarchy

### C#

```
public class TaskCategoryType : Ektron.Cms.API.Common;
```

### Visual Basic

```
Public Class TaskCategoryType
Inherits Ektron.Cms.API.Common
```

### File

TaskCategoryType.vb

### Description

This is class Ektron.Cms.API.Task.TaskCategoryType.

### TaskCategoryType Methods

The methods of the TaskCategoryType class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddTaskType (see page 884) | Adds a new task type to a category by providing the new task type's title and description, the category's ID and whether to make the type active. Once the new task type has been added, an new ID is returned. |
| | DeleteTaskType (see page 885) | Deletes the task type by providing its ID. |
| | GetTaskType (see page 885) | Gets the task type information for a specified ID. |
| | UpdateTaskType (see page 885) | Updates task type information by providing its ID. You can update the Title, Description and set whether the type is active or not. |

### TaskCategoryType.AddTaskType Method

Adds a new task type to a category by providing the new task type's title and description, the category's ID and whether to make the type active. Once the new task type has been added, an new ID is returned.

### C#

```
public Long AddTaskType(String title, String description, Long categoryId, Boolean active);
```

### Visual Basic

```
Public Function AddTaskType(ByVal title As String, ByVal description As String, ByVal
categoryId As Long, ByVal active As Boolean) As Long
```

### Parameters

| Parameters | Description |
|---|---|
| title | A title for the new task type. |
| description | A description for the new task type. |
| categoryId | The ID of the category with which the new task type will be associated. |
| active | Set to True to make the new task type active. Otherwise, set to false. |

**Returns**

Returns the task type's numeric ID.

### TaskCategoryType.DeleteTaskType Method

Deletes the task type by providing its ID.

**C#**

```
public Boolean DeleteTaskType(Long taskTypeId);
```

**Visual Basic**

```
Public Function DeleteTaskType(ByVal taskTypeId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskTypeId | The task type numeric ID to be deleted. |

**Returns**

Returns False if the task type is successfully deleted. Otherwise, it returns True.

### TaskCategoryType.GetTaskType Method

Gets the task type information for a specified ID.

**C#**

```
public TaskTypeData GetTaskType(Long taskTypeId);
```

**Visual Basic**

```
Public Function GetTaskType(ByVal taskTypeId As Long) As TaskTypeData
```

**Parameters**

| Parameters | Description |
|---|---|
| taskTypeId | The numeric ID of the task type. |

**Returns**

Returns TaskTypeData information.

### TaskCategoryType.UpdateTaskType Method

Updates task type information by providing its ID. You can update the Title, Description and set whether the type is active or not.

**C#**

```
public Boolean UpdateTaskType(Long taskTypeId, String title, String description, Boolean
active);
```

**Visual Basic**

```
Public Function UpdateTaskType(ByVal taskTypeId As Long, ByVal title As String, ByVal
description As String, ByVal active As Boolean) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| taskTypeId | The numeric ID of the task type. |
| title | The title of the task type. |
| description | A description for the task type. |
| active | Set to True to make the task type active. Otherwise, set to false. |

### Returns

Returns False if the task type is successfully updated. Otherwise, it returns True.

# Ektron.Cms.API.UrlAliasing Namespace

This is namespace Ektron.Cms.API.UrlAliasing.

### Classes

| | Name | Description |
|---|---|---|
| | UrlAliasAuto (see page 886) | This class allows you to work with automatic aliasing. |
| | UrlAliasCommon (see page 894) | This class allows you work with the common components of URL Aliasing. |
| | UrlAliasManual (see page 895) | This class allows you to work with manual aliasing. |
| | UrlAliasRegEx (see page 903) | This class allows you to work with the Regular Expressions within aliasing. |
| | UrlAliasSettings (see page 907) | This class allows you to work with settings for enabling and disabling the various types of aliasing available in CMS400.NET. In addition, you can work with alias settings for caching, tracing and page extensions. These settings reflect the same settings in the **Workarea** > **Settings** > **Configuration** > **URL Aliasing** > **Settings** screen. |

# Classes

The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | UrlAliasAuto (see page 886) | This class allows you to work with automatic aliasing. |
| | UrlAliasCommon (see page 894) | This class allows you work with the common components of URL Aliasing. |
| | UrlAliasManual (see page 895) | This class allows you to work with manual aliasing. |
| | UrlAliasRegEx (see page 903) | This class allows you to work with the Regular Expressions within aliasing. |
| | UrlAliasSettings (see page 907) | This class allows you to work with settings for enabling and disabling the various types of aliasing available in CMS400.NET. In addition, you can work with alias settings for caching, tracing and page extensions. These settings reflect the same settings in the **Workarea** > **Settings** > **Configuration** > **URL Aliasing** > **Settings** screen. |

## UrlAliasAuto Class

This class allows you to work with automatic aliasing.

### Class Hierarchy

### C#

```
public class UrlAliasAuto : Ektron.Cms.API.CmsApi;
```

### Visual Basic

```
Public Class UrlAliasAuto
Inherits Ektron.Cms.API.CmsApi
```

### File

UrlAliasAuto.cs

## UrlAliasAuto Methods

The methods of the UrlAliasAuto class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 887) | Adds a new automatic alias configuration by providing information for the UrlAliasAutoData (see page 1365) data class. |
| | ClearCache (see page 887) | Clears the auto alias cache. |
| | Delete (see page 888) | Deletes an automatic aliasing configuration by providing its ID. |
| | GetAutoAliasNameTypes (see page 888) | Gets a list of page names for automatic alias types. Types include:<br>• ContentTitle = 1<br>• ContentId = 2<br>• ContentIdAndLanguage = 3 |
| | GetAutoAliasTypes (see page 888) | Gets a list of automatic aliasing types. Types include:<br>• Taxonomy (see page 842) = 1<br>• Folder (see page 652) = 2 |
| | GetFolderItem (see page 888) | Gets a content's single record for folder automatic aliases by providing a content's ID and language. |
| | GetFolderList (see page 889) | Gets a content's list of records for folder automatic aliases by providing a content's ID and language.. |
| | GetFolderSource (see page 889) | Gets a list of all folders within a site that are avaliable for automatic alising. |
| | GetItem (see page 889) | Gets a single record for a given automatic aliasing configuration ID. |
| | GetItemForContent (see page 890) | Gets the first record in the list of automatic aliases for a given content ID. The first record is returned for both folder aliases and taxonomy aliases. |
| | GetList (see page 890) | Gets a list of records. |
| | GetListForContent (see page 891) | Gets a list of automatic aliases for a given content ID. The list includes both folder aliases and taxonomy aliases. |
| | GetTarget (see page 891) | This is the overview for the GetTarget method overload. |
| | GetTaxonomyItem (see page 893) | Gets a content's single record for taxonomy automatic aliases by providing a content's ID and language. |
| | GetTaxonomyList (see page 893) | Gets a specified content's list of records for taxonomy automatic aliases by providing a content's ID and language. |
| | GetTaxonomySource (see page 893) | Gets a list of all taxonomies for a specified language that are avaliable for automatic aliasing. |
| | Update (see page 893) | Modifies an existing automatic alias configuration by providing information for the UrlAliasAutoData (see page 1365) data class. |

## UrlAliasAuto.Add Method

Adds a new automatic alias configuration by providing information for the UrlAliasAutoData (see page 1365) data class.

**C#**

```
public UrlAliasAutoData Add(UrlAliasAutoData data);
```

**Visual Basic**

```
Public Function Add(data As UrlAliasAutoData) As UrlAliasAutoData
```

### Parameters

| Parameters | Description |
|---|---|
| data | UrlAliasAutoData (see page 1365) information. |

## UrlAliasAuto.ClearCache Method

Clears the auto alias cache.

**C#**

```
public void ClearCache();
```

**Visual Basic**

```
Public Function ClearCache() As void
```

### UrlAliasAuto.Delete Method

Deletes an automatic aliasing configuration by providing its ID.

**C#**

```
public void Delete(long autoId);
```

**Visual Basic**

```
Public Function Delete(autoId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| autoId | The ID of the automatic aliasing configuration. |
| callerUserId | The ID of the user deleting the automatic aliasing. |

### UrlAliasAuto.GetAutoAliasNameTypes Method

Gets a list of page names for automatic alias types. Types include:

- ContentTitle = 1
- ContentId = 2
- ContentIdAndLanguage = 3

**C#**

```
public List<EkEnumeration.AutoAliasNameType> GetAutoAliasNameTypes();
```

**Visual Basic**

```
Public Function GetAutoAliasNameTypes() As List<EkEnumeration.AutoAliasNameType>
```

**Returns**

A list of page names for automatic alias types.

### UrlAliasAuto.GetAutoAliasTypes Method

Gets a list of automatic aliasing types. Types include:

- Taxonomy (see page 842) = 1
- Folder (see page 652) = 2

**C#**

```
public List<EkEnumeration.AutoAliasType> GetAutoAliasTypes();
```

**Visual Basic**

```
Public Function GetAutoAliasTypes() As List<EkEnumeration.AutoAliasType>
```

**Returns**

A list of automatic aliasing types.

### UrlAliasAuto.GetFolderItem Method

Gets a content's single record for folder automatic aliases by providing a content's ID and language.

**C#**

```
public string GetFolderItem(long contentId, int contentLanguage);
```

**Visual Basic**

```
Public Function GetFolderItem(contentId As long, contentLanguage As Integer) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The content ID. |
| contentLanguage | The content's language ID. |

**Returns**

A content's single record for folder automatic aliases. Returns an empty record if no data is found.

### UrlAliasAuto.GetFolderList Method

Gets a content's list of records for folder automatic aliases by providing a content's ID and language..

**C#**

```
public List<UrlAliasAutoData> GetFolderList(long contentId, int contentLanguage);
```

**Visual Basic**

```
Public Function GetFolderList(contentId As long, contentLanguage As Integer) As
List<UrlAliasAutoData>
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The content ID. |
| contentLanguage | The content's language ID. |

**Returns**

A content's list of records for folder automatic aliases. Returns an empty list if no data is found.

### UrlAliasAuto.GetFolderSource Method

Gets a list of all folders within a site that are avaliable for automatic alising.

**C#**

```
public List<FolderData> GetFolderSource(long siteId);
```

**Visual Basic**

```
Public Function GetFolderSource(siteId As long) As List<FolderData>
```

**Parameters**

| Parameters | Description |
|---|---|
| siteId | The ID of the site from which to retrieve a list of folders. |

**Returns**

A list of all folders within a site that are avaliable for automatic alising.

### UrlAliasAuto.GetItem Method

Gets a single record for a given automatic aliasing configuration ID.

**C#**

```
public UrlAliasAutoData GetItem(long autoId);
```

**Visual Basic**

```
Public Function GetItem(autoId As long) As UrlAliasAutoData
```

**Parameters**

| Parameters | Description |
|---|---|
| autoId | The ID of the automatic aliasing configuration. |

**Returns**

A single record for a given automatic aliasing configuration ID. Returns an empty record if no data is found.

### UrlAliasAuto.GetItemForContent Method

Gets the first record in the list of automatic aliases for a given content ID. The first record is returned for both folder aliases and taxonomy aliases.

**C#**

```
public string GetItemForContent(long contentId);
```

**Visual Basic**

```
Public Function GetItemForContent(contentId As long) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The content ID. |

**Returns**

The first record in the list of automatic aliases for a given content ID.

### UrlAliasAuto.GetList Method

Gets a list of records.

**C#**

```
public List<UrlAliasAutoData> GetList(PagingInfo paging, long siteId, int contentLanguage,
EkEnumeration.AutoAliasSearchField searchField, string searchValue,
EkEnumeration.AutoAliasOrderBy orderBy);
```

**Visual Basic**

```
Public Function GetList(paging As PagingInfo, siteId As long, contentLanguage As Integer,
searchField As EkEnumeration.AutoAliasSearchField, searchValue As string, orderBy As
EkEnumeration.AutoAliasOrderBy) As List<UrlAliasAutoData>
```

**Parameters**

| Parameters | Description |
|---|---|
| paging | The paging object. |
| siteId | The Site (see page 694) ID. |
| contentLanguage | The language ID. |

| | |
|---|---|
| searchField | The type of search to perform.<br>• None<br>• All<br>• ContentId<br>• ContentTitle<br>• Target<br>• [Alias] |
| searchValue | The value for which to search. |
| orderBy | Select the order of the list. Choices are:<br>• None<br>• ContentId<br>• ContentLanguage<br>• ContentTitle<br>• IsDefault<br>• IsEnabled<br>• Target<br>• SiteId<br>• QueryString<br>• FileExtension<br>• DisplayAlias |

**Returns**

Gets a list of records. Returns an empty list if no data is found.

### UrlAliasAuto.GetListForContent Method

Gets a list of automatic aliases for a given content ID. The list includes both folder aliases and taxonomy aliases.

**C#**

```
public List<UrlAliasAutoData> GetListForContent(long contentId);
```

**Visual Basic**

```
Public Function GetListForContent(contentId As long) As List<UrlAliasAutoData>
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The ID of the content for which to retrieve a list of aliases. |

**Returns**

A list of automatic aliases for a given content ID.

### GetTarget Method

This is the overview for the GetTarget method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | UrlAliasAuto.GetTarget (string, string) (see page 892) | Get the destination for a alias name. |
| | UrlAliasAuto.GetTarget (string, string, EkEnumeration.AutoAliasType) (see page 892) | Get the destination for a alias name for a perticular source. |

## UrlAliasAuto.GetTarget Method (string, string)

Get the destination for a alias name.

**C#**

```
public string GetTarget(string urlAlias, string host);
```

**Visual Basic**

```
Public Function GetTarget(urlAlias As string, host As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| urlAlias | The alias name without the QueryString. |
| host | The host name. |

**Returns**

The destination for a alias name. Returns an empty string ID if no destination is found.

## UrlAliasAuto.GetTarget Method (string, string, EkEnumeration.AutoAliasType)

Get the destination for a alias name for a perticular source.

**C#**

```
public string GetTarget(string urlAlias, string host, EkEnumeration.AutoAliasType
autoAliasType);
```

**Visual Basic**

```
Public Function GetTarget(urlAlias As string, host As string, autoAliasType As
EkEnumeration.AutoAliasType) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| urlAlias | The alias name without the query string. |
| host | The host name. |
| autoAliasType | The source type. Choices are:<br>• Taxonomy (see page 842)<br>• Folder (see page 652) |

**Returns**

The destination for a alias name for a perticular source.

### UrlAliasAuto.GetTaxonomyItem Method

Gets a content's single record for taxonomy automatic aliases by providing a content's ID and language.

**C#**

```
public string GetTaxonomyItem(long contentId, int contentLanguage);
```

**Visual Basic**

```
Public Function GetTaxonomyItem(contentId As long, contentLanguage As Integer) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The content ID. |
| contentLanguage | The content's language ID. |

**Returns**

A content's single record for taxonomy automatic aliases.

### UrlAliasAuto.GetTaxonomyList Method

Gets a specified content's list of records for taxonomy automatic aliases by providing a content's ID and language.

**C#**

```
public List<UrlAliasAutoData> GetTaxonomyList(long contentId, int contentLanguage);
```

**Visual Basic**

```
Public Function GetTaxonomyList(contentId As long, contentLanguage As Integer) As
List<UrlAliasAutoData>
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The content ID. |
| contentLanguage | The content's language ID. |

### UrlAliasAuto.GetTaxonomySource Method

Gets a list of all taxonomies for a specified language that are avaliable for automatic aliasing.

**C#**

```
public List<TaxonomyData> GetTaxonomySource(int languageId);
```

**Visual Basic**

```
Public Function GetTaxonomySource(languageId As Integer) As List<TaxonomyData>
```

**Parameters**

| Parameters | Description |
|---|---|
| languageId | The taxonomy's language ID. |

**Returns**

A list of all taxonomies for a specified language that are avaliable for automatic aliasing.

### UrlAliasAuto.Update Method

Modifies an existing automatic alias configuration by providing information for the UrlAliasAutoData (see page 1365) data class.

**C#**

```
public UrlAliasAutoData Update(UrlAliasAutoData data);
```

**Visual Basic**

```
Public Function Update(data As UrlAliasAutoData) As UrlAliasAutoData
```

**Parameters**

| Parameters | Description |
|---|---|
| data | UrlAliasAutoData (see page 1365) information. |

## UrlAliasCommon Class

This class allows you work with the common components of URL Aliasing.

**Class Hierarchy**

**C#**

```
public class UrlAliasCommon : Ektron.Cms.API.CmsApi;
```

**Visual Basic**

```
Public Class UrlAliasCommon
Inherits Ektron.Cms.API.CmsApi
```

**File**

UrlAliasCommon.cs

### UrlAliasCommon Methods

The methods of the UrlAliasCommon class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | GetAliasForContent (see page 894) | Gets all the aliases for a content item. |
| | GetTarget (see page 894) | This is the overview for the GetTarget method overload. |

### UrlAliasCommon.GetAliasForContent Method

Gets all the aliases for a content item.

**C#**

```
public string GetAliasForContent(long contentId);
```

**Visual Basic**

```
Public Function GetAliasForContent(contentId As long) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The content's ID. |

### GetTarget Method

This is the overview for the GetTarget method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | UrlAliasCommon.GetTarget (System.Web.HttpContext) (see page 895) | Get the destination link (partial URL) for an alias. |
| | UrlAliasCommon.GetTarget (Uri) (see page 895) | Get the destination link (partial URL) for an alias. |

### UrlAliasCommon.GetTarget Method (System.Web.HttpContext)

Get the destination link (partial URL) for an alias.

**C#**

```
public string GetTarget(System.Web.HttpContext context);
```

**Visual Basic**

```
Public Function GetTarget(context As System.Web.HttpContext) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| url | The Uri object. |

### UrlAliasCommon.GetTarget Method (Uri)

Get the destination link (partial URL) for an alias.

**C#**

```
public string GetTarget(Uri url);
```

**Visual Basic**

```
Public Function GetTarget(url As Uri) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| url | The Uri object. |

## UrlAliasManual Class

This class allows you to work with manual aliasing.

**Class Hierarchy**

**C#**

```
public class UrlAliasManual : Ektron.Cms.API.CmsApi;
```

**Visual Basic**

```
Public Class UrlAliasManual
Inherits Ektron.Cms.API.CmsApi
```

**File**

UrlAliasManual.cs

## UrlAliasManual Methods

The methods of the UrlAliasManual class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Add (see page 896) | Adds a new manual alias by providing information for the UrlAliasManualData (see page 1370) data class. |
| | ClearCache (see page 896) | Clears the cache. |
| | Delete (see page 896) | Deletes an alias by providing its ID. |
| | GetDefaultAlias (see page 897) | Gets the primary alias for a given content ID. |
| | GetFileExtensions (see page 897) | Gets a list of file extensions. |
| | GetItem (see page 897) | Gets a single record for an alias ID. |
| | GetList (see page 898) | This is the overview for the GetList method overload. |
| | GetSiteList (see page 902) | Gets a list of sites. |
| | GetTemplateList (see page 902) | Gets a list of templates for a given content item by providing the content's ID. |
| | Update (see page 902) | Update's an existing manual alias by providing information for the UrlAliasManualData (see page 1370) data class. |
| | ValidateName (see page 903) | Validates the Url Alais. |

## UrlAliasManual.Add Method

Adds a new manual alias by providing information for the UrlAliasManualData (see page 1370) data class.

**C#**

```
public UrlAliasManualData Add(UrlAliasManualData data);
```

**Visual Basic**

```
Public Function Add(data As UrlAliasManualData) As UrlAliasManualData
```

### Parameters

| Parameters | Description |
|---|---|
| data | UrlAliasManualData (see page 1370) data. |
| callerUserId | The ID of the user addding the new manual alias. |

### Returns

Returns the new record.

## UrlAliasManual.ClearCache Method

Clears the cache.

**C#**

```
public void ClearCache();
```

**Visual Basic**

```
Public Function ClearCache() As void
```

## UrlAliasManual.Delete Method

Deletes an alias by providing its ID.

**C#**

```
public void Delete(long aliasId);
```

**Visual Basic**

```
Public Function Delete(aliasId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| aliasId | The ID of the alias to be deleted. |

### UrlAliasManual.GetDefaultAlias Method

Gets the primary alias for a given content ID.

**C#**

```
public UrlAliasManualData GetDefaultAlias(long contentId);
```

**Visual Basic**

```
Public Function GetDefaultAlias(contentId As long) As UrlAliasManualData
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The content's ID. |

**Returns**

The primary alias for a given content ID.

### UrlAliasManual.GetFileExtensions Method

Gets a list of file extensions.

**C#**

```
public List<string> GetFileExtensions();
```

**Visual Basic**

```
Public Function GetFileExtensions() As List<string>
```

**Returns**

A list of file extensions.

### UrlAliasManual.GetItem Method

Gets a single record for an alias ID.

**C#**

```
public UrlAliasManualData GetItem(long aliasId);
```

**Visual Basic**

```
Public Function GetItem(aliasId As long) As UrlAliasManualData
```

**Parameters**

| Parameters | Description |
|---|---|
| aliasId | The ID of the alias. |

**Returns**

A single record for an alias ID.

## GetList Method

This is the overview for the GetList method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | UrlAliasManual.GetList (PagingInfo, EkEnumeration.UrlAliasingOrderBy) (see page 898) | Gets a list of all the manual aliases. |
| | UrlAliasManual.GetList (PagingInfo, long, bool, EkEnumeration.UrlAliasingOrderBy) (see page 899) | Gets a list of active aliases or a list of all aliases for a content item by providing its ID. |
| | UrlAliasManual.GetList (PagingInfo, long, EkEnumeration.UrlAliasingOrderBy) (see page 899) | Gets a list of manual aliases for a site by providing the site's ID. |
| | UrlAliasManual.GetList (PagingInfo, long, int, EkEnumeration.UrlAliasingOrderBy) (see page 901) | Gets a list of aliases for a site by providing paging, site ID, language and order by information. |
| | UrlAliasManual.GetList (PagingInfo, long, int, EkEnumeration.UrlAliasSearchField, string, EkEnumeration.UrlAliasingOrderBy) (see page 900) | Gets a list of aliases for a site based on specified search criteria by providing paging, site ID, language, search criteria and order by information. |

## UrlAliasManual.GetList Method (PagingInfo, EkEnumeration.UrlAliasingOrderBy)

Gets a list of all the manual aliases.

### C#

```
public List<UrlAliasManualData> GetList(PagingInfo paging, EkEnumeration.UrlAliasingOrderBy
orderBy);
```

### Visual Basic

```
Public Function GetList(paging As PagingInfo, orderBy As EkEnumeration.UrlAliasingOrderBy) As
List<UrlAliasManualData>
```

### Parameters

| Parameters | Description |
|---|---|
| paging | The paging object. |

| | |
|---|---|
| orderBy | Select the order of the list. Choices are:<br>• None<br>• ContentId<br>• ContentLanguage<br>• ContentTitle<br>• IsDefault<br>• IsEnabled<br>• Target<br>• SiteId<br>• QueryString<br>• FileExtension<br>• DisplayAlias |

**Returns**

A list of all the manual aliases.

### UrlAliasManual.GetList Method (PagingInfo, long, EkEnumeration.UrlAliasingOrderBy)

Gets a list of manual aliases for a site by providing the site's ID.

**C#**

```
public List<UrlAliasManualData> GetList(PagingInfo paging, long siteId,
EkEnumeration.UrlAliasingOrderBy orderBy);
```

**Visual Basic**

```
Public Function GetList(paging As PagingInfo, siteId As long, orderBy As
EkEnumeration.UrlAliasingOrderBy) As List<UrlAliasManualData>
```

**Parameters**

| Parameters | Description |
|---|---|
| paging | The paging object. |
| siteId | The ID of the site. |
| orderBy | Select the order of the list. Choices are:<br>• None<br>• ContentLanguage<br>• SourceName<br>• Active<br>• Type |

**Returns**

A list of manual aliases for a site.

### UrlAliasManual.GetList Method (PagingInfo, long, bool, EkEnumeration.UrlAliasingOrderBy)

Gets a list of active aliases or a list of all aliases for a content item by providing its ID.

**C#**

```
public List<UrlAliasManualData> GetList(PagingInfo paging, long contentId, bool onlyActive,
EkEnumeration.UrlAliasingOrderBy orderBy);
```

**Visual Basic**

```
Public Function GetList(paging As PagingInfo, contentId As long, onlyActive As bool, orderBy
As EkEnumeration.UrlAliasingOrderBy) As List<UrlAliasManualData>
```

**Parameters**

| Parameters | Description |
|---|---|
| paging | The paging object. |
| contentId | The content's ID. |
| onlyActive | Set to true to only return a list of active manual aliases. |
| orderBy | Select the order of the list. Choices are:<br>• None<br>• ContentId<br>• ContentLanguage<br>• ContentTitle<br>• IsDefault<br>• IsEnabled<br>• Target<br>• SiteId<br>• QueryString<br>• FileExtension<br>• DisplayAlias |

**Returns**

A list of only active or inactive aliases for a content item.

### UrlAliasManual.GetList Method (PagingInfo, long, int, EkEnumeration.UrlAliasSearchField, string, EkEnumeration.UrlAliasingOrderBy)

Gets a list of aliases for a site based on specified search criteria by providing paging, site ID, language, search criteria and order by information.

**C#**

```
public List<UrlAliasManualData> GetList(PagingInfo paging, long siteId, int contentLanguage,
EkEnumeration.UrlAliasSearchField searchField, string searchValue,
EkEnumeration.UrlAliasingOrderBy orderBy);
```

**Visual Basic**

```
Public Function GetList(paging As PagingInfo, siteId As long, contentLanguage As Integer,
searchField As EkEnumeration.UrlAliasSearchField, searchValue As string, orderBy As
EkEnumeration.UrlAliasingOrderBy) As List<UrlAliasManualData>
```

**Parameters**

| Parameters | Description |
|---|---|
| paging | The paging object. |
| siteId | The ID of the Site (see page 694). |
| contentLanguage | The language ID. |

| | |
|---|---|
| searchField | The type of search to perform.<br>• None<br>• All<br>• ContentId<br>• ContentTitle<br>• Target<br>• [Alias] |
| searchValue | The value for which to search. |
| orderBy | Select the order of the list. Choices are:<br>• None<br>• ContentId<br>• ContentLanguage<br>• ContentTitle<br>• IsDefault<br>• IsEnabled<br>• Target<br>• SiteId<br>• QueryString<br>• FileExtension<br>• DisplayAlias |

**Returns**

A list of aliases for a site based on specified search criteria.

### UrlAliasManual.GetList Method (PagingInfo, long, int, EkEnumeration.UrlAliasingOrderBy)

Gets a list of aliases for a site by providing paging, site ID, language and order by information.

**C#**

```
public List<UrlAliasManualData> GetList(PagingInfo paging, long siteId, int contentLanguage,
EkEnumeration.UrlAliasingOrderBy orderBy);
```

**Visual Basic**

```
Public Function GetList(paging As PagingInfo, siteId As long, contentLanguage As Integer,
orderBy As EkEnumeration.UrlAliasingOrderBy) As List<UrlAliasManualData>
```

**Parameters**

| Parameters | Description |
|---|---|
| paging | The paging object. |
| siteId | The ID of the site. |
| contentLanguage | The language ID. |

| | |
|---|---|
| orderBy | Select the order of the list. Choices are:<br>• None<br>• ContentId<br>• ContentLanguage<br>• ContentTitle<br>• IsDefault<br>• IsEnabled<br>• Target<br>• SiteId<br>• QueryString<br>• FileExtension<br>• DisplayAlias |

**Returns**

A list of aliases for a site.

### UrlAliasManual.GetSiteList Method

Gets a list of sites.

**C#**

```
public Dictionary<long,string> GetSiteList();
```

**Visual Basic**

```
Public Function GetSiteList() As Dictionary<long,string>
```

**Returns**

A list of sites.

### UrlAliasManual.GetTemplateList Method

Gets a list of templates for a given content item by providing the content's ID.

**C#**

```
public List<LibraryData> GetTemplateList(long contentId);
```

**Visual Basic**

```
Public Function GetTemplateList(contentId As long) As List<LibraryData>
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The ID of the content. |

**Returns**

A list of templates for a given content item.

### UrlAliasManual.Update Method

Update's an existing manual alias by providing information for the UrlAliasManualData (see page 1370) data class.

**C#**

```
public UrlAliasManualData Update(UrlAliasManualData data);
```

**Visual Basic**

```
Public Function Update(data As UrlAliasManualData) As UrlAliasManualData
```

**Parameters**

| Parameters | Description |
|---|---|
| data | UrlAliasManualData (see page 1370) data |

**Returns**

Returns the record with the updated field and any calculated field.

### UrlAliasManual.ValidateName Method

Validates the Url Alais.

**C#**

```
public string ValidateName(string aliasName);
```

**Visual Basic**

```
Public Function ValidateName(aliasName As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| aliasName | |

**Returns**

An empty string if it passes validation or the error message.

## UrlAliasRegEx Class

This class allows you to work with the Regular Expressions within aliasing.

**Class Hierarchy**

**C#**

```
public class UrlAliasRegEx : Ektron.Cms.API.CmsApi;
```

**Visual Basic**

```
Public Class UrlAliasRegEx
Inherits Ektron.Cms.API.CmsApi
```

**File**

UrlAliasRegEx.cs

### UrlAliasRegEx Methods

The methods of the UrlAliasRegEx class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Add (see page 904) | Adds a new regular expression based on UrlAliasRegExData (see page 1374) data. |
| | ClearCache (see page 904) | Refreshes the cache. |

| | | |
|---|---|---|
| | Delete (see page 904) | Deletes an regular expression record from the database. |
| | EvaluateExpression (see page 904) | Evaluate an expression by passing the URL, expression and expression map. |
| | GetItem (see page 905) | Gets a single expression for a specified ID. |
| | GetList (see page 905) | Gets a list of either the active regular expression or a list of all expressions. |
| | GetPriorityList (see page 906) | Gets a list of page names for regular expressions alias types. |
| | GetTarget (see page 906) | Gets the target URL for a matching expression by providing the RegEx alias and server host name. |
| | Update (see page 906) | Updates an existing regular expression record based on UrlAliasRegExData (see page 1374) data. |

## UrlAliasRegEx.Add Method

Adds a new regular expression based on UrlAliasRegExData (see page 1374) data.

**C#**

```
public UrlAliasRegExData Add(UrlAliasRegExData data);
```

**Visual Basic**

```
Public Function Add(data As UrlAliasRegExData) As UrlAliasRegExData
```

**Parameters**

| Parameters | Description |
|---|---|
| data | UrlAliasRegExData (see page 1374) data. |

## UrlAliasRegEx.ClearCache Method

Refreshes the cache.

**C#**

```
public void ClearCache();
```

**Visual Basic**

```
Public Function ClearCache() As void
```

## UrlAliasRegEx.Delete Method

Deletes an regular expression record from the database.

**C#**

```
public void Delete(long regExId);
```

**Visual Basic**

```
Public Function Delete(regExId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| regExId | The ID of the regular expression. |

## UrlAliasRegEx.EvaluateExpression Method

Evaluate an expression by passing the URL, expression and expression map.

**C#**

```
public string EvaluateExpression(Uri url, string expression, string expressionMap);
```

**Visual Basic**

```
Public Function EvaluateExpression(url As Uri, expression As string, expressionMap As string)
As string
```

### Parameters

| Parameters | Description |
|---|---|
| url | The URL to be evaluated. |
| expression | The regular expression. |
| expressionMap | The destination URL for the expression. |

### Returns

Returns the destination URL or an empty string if no match is found.

## UrlAliasRegEx.GetItem Method

Gets a single expression for a specified ID.

### C#

```
public UrlAliasRegExData GetItem(long regExId);
```

### Visual Basic

```
Public Function GetItem(regExId As long) As UrlAliasRegExData
```

### Parameters

| Parameters | Description |
|---|---|
| regExId | The ID of the regular expression. |

### Returns

A single expression for a specified ID.

## UrlAliasRegEx.GetList Method

Gets a list of either the active regular expression or a list of all expressions.

### C#

```
public List<UrlAliasRegExData> GetList(PagingInfo paging, long SiteId, bool onlyActive,
EkEnumeration.RegExAliasSearchField searchField, string searchValue,
EkEnumeration.RegExOrderBy orderBy);
```

### Visual Basic

```
Public Function GetList(paging As PagingInfo, SiteId As long, onlyActive As bool, searchField
As EkEnumeration.RegExAliasSearchField, searchValue As string, orderBy As
EkEnumeration.RegExOrderBy) As List<UrlAliasRegExData>
```

### Parameters

| Parameters | Description |
|---|---|
| paging | The paging object. |
| onlyActive | Set to true to return only the active RegEx. |
| searchField | Decide which field is being searched. Choices are:<br>• None<br>• All<br>• ExpressionName<br>• TransformedUrl |
| searchValue | The value for which to search. |

| | |
|---|---|
| orderBy | Select how to order the results. Choices are:<br>• None<br>• Priority<br>• Expression<br>• Target<br>• Active<br>• ExpressionName<br>• TransformedUrl |

**Returns**

Returns either the active regular expression or a list of all expressions.

### UrlAliasRegEx.GetPriorityList Method

Gets a list of page names for regular expressions alias types.

**C#**

```
public List<EkEnumeration.RegExPriority> GetPriorityList();
```

**Visual Basic**

```
Public Function GetPriorityList() As List<EkEnumeration.RegExPriority>
```

**Returns**

A list of page names for regular expressions alias types.

### UrlAliasRegEx.GetTarget Method

Gets the target URL for a matching expression by providing the RegEx alias and server host name.

**C#**

```
public string GetTarget(string urlAlias, string host);
```

**Visual Basic**

```
Public Function GetTarget(urlAlias As string, host As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| host | The server's host name. For example, if your site is www.example.com, enter **example**. |
| url | The RegEx aliased URL. |

**Returns**

The target URL for a matching expression.

### UrlAliasRegEx.Update Method

Updates an existing regular expression record based on UrlAliasRegExData (see page 1374) data.

**C#**

```
public UrlAliasRegExData Update(UrlAliasRegExData data);
```

**Visual Basic**

```
Public Function Update(data As UrlAliasRegExData) As UrlAliasRegExData
```

### Parameters

| Parameters | Description |
|---|---|
| data | UrlAliasRegExData (see page 1374) data |

## UrlAliasSettings Class

This class allows you to work with settings for enabling and disabling the various types of aliasing available in CMS400.NET. In addition, you can work with alias settings for caching, tracing and page extensions. These settings reflect the same settings in the **Workarea** > **Settings** > **Configuration** > **URL Aliasing** > **Settings** screen.

### Class Hierarchy

### C#

```
public class UrlAliasSettings : Ektron.Cms.API.CmsApi;
```

### Visual Basic

```
Public Class UrlAliasSettings
Inherits Ektron.Cms.API.CmsApi
```

### File

UrlAliasSettings.cs

### UrlAliasSettings Methods

The methods of the UrlAliasSettings class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | RefreshSettingsCache (see page 907) | Refreshes the settings cache. |

### UrlAliasSettings.RefreshSettingsCache Method

Refreshes the settings cache.

### C#

```
public void RefreshSettingsCache();
```

### Visual Basic

```
Public Function RefreshSettingsCache() As void
```

### UrlAliasSettings Properties

The properties of the UrlAliasSettings class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AutoCacheDuration (see page 909) | The time for which the page should reside in cache in seconds. |
| | AutoCacheSize (see page 909) | The maximum amount of items to store in the cache. If EnableAutoCache is set to true, this number reflects the number of automatic aliases that your server will cache. If this number is increased, it is possible that your performance will improve. But at some point, the size of the cache may degrade your server's performance. |
| | DefaultPage (see page 909) | Get or set the default page for the site. This page appears if someone enters your Web site's address without a page name. You can also specify a default document in IIS. |
| | DisableLanguageAwareness (see page 909) | This Setting will use the default language when it does not find an alis for the requested language. |

| | | |
|---|---|---|
| | EnableAutoAliasing (see page 909) | Gets or sets whether automatic aliasing is enabled.<br>• **True** = automatic aliasing is enabled.<br>• **False** = automatic aliasing is disable. |
| | EnableAutoCaching (see page 910) | Enable or disable automatic aliasing caching. Enabling the automatic cache improves the server's performance. It minimizes the number of database by placing all automatic aliasing information in your server's memory.<br>• **True** - enable automatic aliasing caching.<br>• **False** - disable automatic aliasing caching. |
| | EnabledTracing (see page 910) | Enable or disable tracing. Tracing allows you to troubleshoot problems with aliasing. Developers can use this field to test Ektron's APIs for getting the target/destination URL.<br>• **True** - enable tracing.<br>• **False** - disable tracing. |
| | EnableManualAliasing (see page 910) | Gets or sets whether manual aliasing is enabled.<br>• **True** = manual aliasing is enabled.<br>• **False** = manual aliasing is disable. |
| | EnableManualCaching (see page 910) | Enable or disable the Manual aliasing cache. Enabling the manual cache improves the server's performance. It minimizes the number of database by placing all manual aliasing information in your server's memory. When you are developing a site, you might want to set this to false to make sure that everything is coming from the database.<br>• **True** - enable manual aliasing cache.<br>• **False** - disable manual aliasing cache. |
| | EnableRegExAliasing (see page 910) | Gets or sets whether RegEx aliasing is enabled.<br>• **True** = RegEx aliasing is enabled.<br>• **False** = RegEx aliasing is disable. |
| | EnableRegExCaching (see page 911) | Enable or disable the regular expression aliasing cache.<br>• **True** - enable regular expression aliasing cache.<br>• **False** - disable regular expression aliasing cache. |
| | IsAutoAliasEnabled (see page 911) | Enable or disable automatic aliasing.<br>• **True** - enable automatic aliasing.<br>• **False** - disable automatic aliasing. |
| | IsManualAliasEnabled (see page 911) | Enable or disable the manual aliasing.<br>• **True** - enable manual aliasing.<br>• **False** - disable manual aliasing. |
| | IsRegExAliasEnabled (see page 911) | Enable or disable regular expression aliasing.<br>• **True** - enable regular expression aliasing.<br>• **False** - disable regular expression aliasing. |
| | ManualCacheDuration (see page 912) | The time for which the page should reside in cache in seconds. |
| | ManualCacheSize (see page 912) | But, if you specify one here, the response time to display the page will be much quicker. The maximum amount of items to store in the cache. |
| | OverrideTemplate (see page 912) | The alias takes precedence over a templcate with the same name. |
| | PageExtensions (see page 912) | Get or set the page extension for which aliases can be created. |
| | QueryStringAction (see page 912) | The action to take when an url alias has a querystring. |

| | | |
|---|---|---|
| | RegExCacheDuration (see page 912) | The time for which the page should reside in cache in seconds. |
| | RegExCacheSize (see page 913) | The maximum amount of items to store in the cache. |

### UrlAliasSettings.AutoCacheDuration Property

The time for which the page should reside in cache in seconds.

**C#**

```
public int AutoCacheDuration;
```

**Visual Basic**

```
Public Property AutoCacheDuration() As Integer
```

### UrlAliasSettings.AutoCacheSize Property

The maximum amount of items to store in the cache. If EnableAutoCache is set to true, this number reflects the number of automatic aliases that your server will cache. If this number is increased, it is possible that your performance will improve. But at some point, the size of the cache may degrade your server's performance.

**C#**

```
public int AutoCacheSize;
```

**Visual Basic**

```
Public Property AutoCacheSize() As Integer
```

### UrlAliasSettings.DefaultPage Property

Get or set the default page for the site. This page appears if someone enters your Web site's address without a page name. You can also specify a default document in IIS.

**C#**

```
public string DefaultPage;
```

**Visual Basic**

```
Public Property DefaultPage() As string
```

### UrlAliasSettings.DisableLanguageAwareness Property

This Setting will use the default language when it does not find an alis for the requested language.

**C#**

```
public bool DisableLanguageAwareness;
```

**Visual Basic**

```
Public Property DisableLanguageAwareness() As bool
```

### UrlAliasSettings.EnableAutoAliasing Property

Gets or sets whether automatic aliasing is enabled.

- **True** = automatic aliasing is enabled.
- **False** = automatic aliasing is disable.

**C#**

```
public bool EnableAutoAliasing;
```

**Visual Basic**

```
Public Property EnableAutoAliasing() As bool
```

### UrlAliasSettings.EnableAutoCaching Property

Enable or disable automatic aliasing caching. Enabling the automatic cache improves the server's performance. It minimizes the number of database by placing all automatic aliasing information in your server's memory.

- **True** - enable automatic aliasing caching.
- **False** - disable automatic aliasing caching.

**C#**

```
public bool EnableAutoCaching;
```

**Visual Basic**

```
Public Property EnableAutoCaching() As bool
```

### UrlAliasSettings.EnabledTracing Property

Enable or disable tracing. Tracing allows you to troubleshoot problems with aliasing. Developers can use this field to test Ektron's APIs for getting the target/destination URL.

- **True** - enable tracing.
- **False** - disable tracing.

**C#**

```
public bool EnabledTracing;
```

**Visual Basic**

```
Public Property EnabledTracing() As bool
```

### UrlAliasSettings.EnableManualAliasing Property

Gets or sets whether manual aliasing is enabled.

- **True** = manual aliasing is enabled.
- **False** = manual aliasing is disable.

**C#**

```
public bool EnableManualAliasing;
```

**Visual Basic**

```
Public Property EnableManualAliasing() As bool
```

### UrlAliasSettings.EnableManualCaching Property

Enable or disable the Manual aliasing cache. Enabling the manual cache improves the server's performance. It minimizes the number of database by placing all manual aliasing information in your server's memory. When you are developing a site, you might want to set this to false to make sure that everything is coming from the database.

- **True** - enable manual aliasing cache.
- **False** - disable manual aliasing cache.

**C#**

```
public bool EnableManualCaching;
```

**Visual Basic**

```
Public Property EnableManualCaching() As bool
```

### UrlAliasSettings.EnableRegExAliasing Property

Gets or sets whether RegEx aliasing is enabled.

- **True** = RegEx aliasing is enabled.
- **False** = RegEx aliasing is disable.

**C#**

```
public bool EnableRegExAliasing;
```

**Visual Basic**

```
Public Property EnableRegExAliasing() As bool
```

### UrlAliasSettings.EnableRegExCaching Property

Enable or disable the regular expression aliasing cache.

- **True** - enable regular expression aliasing cache.
- **False** - disable regular expression aliasing cache.

**C#**

```
public bool EnableRegExCaching;
```

**Visual Basic**

```
Public Property EnableRegExCaching() As bool
```

### UrlAliasSettings.IsAutoAliasEnabled Property

Enable or disable automatic aliasing.

- **True** - enable automatic aliasing.
- **False** - disable automatic aliasing.

**C#**

```
public bool IsAutoAliasEnabled;
```

**Visual Basic**

```
Public Property IsAutoAliasEnabled() As bool
```

### UrlAliasSettings.IsManualAliasEnabled Property

Enable or disable the manual aliasing.

- **True** - enable manual aliasing.
- **False** - disable manual aliasing.

**C#**

```
public bool IsManualAliasEnabled;
```

**Visual Basic**

```
Public Property IsManualAliasEnabled() As bool
```

### UrlAliasSettings.IsRegExAliasEnabled Property

Enable or disable regular expression aliasing.

- **True** - enable regular expression aliasing.
- **False** - disable regular expression aliasing.

**C#**

```
public bool IsRegExAliasEnabled;
```

**Visual Basic**

```
Public Property IsRegExAliasEnabled() As bool
```

### UrlAliasSettings.ManualCacheDuration Property

The time for which the page should reside in cache in seconds.

**C#**

```
public int ManualCacheDuration;
```

**Visual Basic**

```
Public Property ManualCacheDuration() As Integer
```

### UrlAliasSettings.ManualCacheSize Property

But, if you specify one here, the response time to display the page will be much quicker. The maximum amount of items to store in the cache.

**C#**

```
public int ManualCacheSize;
```

**Visual Basic**

```
Public Property ManualCacheSize() As Integer
```

### UrlAliasSettings.OverrideTemplate Property

The alias takes precedence over a templcate with the same name.

**C#**

```
public bool OverrideTemplate;
```

**Visual Basic**

```
Public Property OverrideTemplate() As bool
```

### UrlAliasSettings.PageExtensions Property

Get or set the page extension for which aliases can be created.

**C#**

```
public string PageExtensions;
```

**Visual Basic**

```
Public Property PageExtensions() As string
```

### UrlAliasSettings.QueryStringAction Property

The action to take when an url alias has a querystring.

**C#**

```
public EkEnumeration.QueryStringActionType QueryStringAction;
```

**Visual Basic**

```
Public Property QueryStringAction() As EkEnumeration.QueryStringActionType
```

### UrlAliasSettings.RegExCacheDuration Property

The time for which the page should reside in cache in seconds.

**C#**

```
public int RegExCacheDuration;
```

**Visual Basic**

```
Public Property RegExCacheDuration() As Integer
```

### UrlAliasSettings.RegExCacheSize Property

The maximum amount of items to store in the cache.

**C#**

```
public int RegExCacheSize;
```

**Visual Basic**

```
Public Property RegExCacheSize() As Integer
```

# Ektron.Cms.API.User Namespace

This is namespace Ektron.Cms.API.User.

**Classes**

| | Name | Description |
|---|---|---|
| | ADUser (see page 913) | This is class Ektron.Cms.API.User.ADUser. |
| | User (see page 920) | This is class Ektron.Cms.API.User.User. |

## Classes

The following table lists classes in this documentation.

**Classes**

| | Name | Description |
|---|---|---|
| | ADUser (see page 913) | This is class Ektron.Cms.API.User.ADUser. |
| | User (see page 920) | This is class Ektron.Cms.API.User.User. |

## ADUser Class

**Class Hierarchy**

**C#**

```
public class ADUser : User;
```

**Visual Basic**

```
Public Class ADUser
Inherits User
```

**File**

ADUser.vb

**Description**

This is class Ektron.Cms.API.User.ADUser.

### ADUser Methods

The methods of the ADUser class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddADGroupToCMS (see page 914) | This function allows you add Active Directory (see page 189) groups to the CMS. |

| | | |
|---|---|---|
| | AddADUsersToCMSByUsername (see page 914) | Adds Active Directory (see page 189) users to the CMS by their user name and domain. |
| | DeSynchUserGroups (see page 915) | Updates the group with a new group name. |
| | DeSynchUsers (see page 915) | Update the user with new user details. |
| | GetADMapping (see page 915) | Loads the CMS and Active Directory (see page 189) name. |
| | GetADStatus (see page 916) | Loads the required Synch, Desynch and relationship flags of the configured Active Directory (see page 189). |
| | GetAvailableADGroups (see page 916) | Loads all of the Active Directory (see page 189) groups for a given domain and filter. |
| | GetAvailableADUsers (see page 916) | Loads all available Active Directory (see page 189) users. |
| | GetCMSGroupsToSync (see page 917) | Loads all the CMS groups that need to be synchronized. |
| | GetCMSRelationshipsToSync (see page 917) | Loads all the CMS relationships that need to be synchronized with Active Directory (see page 189). |
| | GetCMSUsersToSync (see page 917) | Loads all the CMS users that need to be synchronized with Active Directory (see page 189). |
| | GetDomains (see page 918) | Loads a list of domains from Active Directory (see page 189). |
| | MapCMSUserGroupToAD (see page 918) | Updates the given CMS group corresponding to the Active Directory (see page 189) group. |
| | MapCMSUserToAD (see page 918) | Maps the given user ID to the corresponding Active Directory (see page 189) entry. |
| | SynchCMSGroupsToAD (see page 919) | Synchronizes the CMS group data to the Active Directory (see page 189) group data. This updates the CMS data with the corresponding to Active Directory (see page 189) changes. |
| | SynchCMSRelationshipsToAD (see page 919) | Synchronizes the CMS relations, such as groups, to the Active Directory (see page 189) for the given users. |
| | SynchCMSUsersToAD (see page 919) | Synchronize the CMS user data to the Active Directory (see page 189) data. |
| | UpdateADMapping (see page 920) | Updates the Active directory information into the CMS based on the user ID and user data. |
| | UpdateADUsersGroups (see page 920) | Updates Active Directory (see page 189) users in a CMS group. |
| | UpdateUsersInfoFromAD (see page 920) | Updates the user details from the Active Directory (see page 189) details. |

## ADUser.AddADGroupToCMS Method

This function allows you add Active Directory (see page 189) 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 (see page 189) group paths. |

## ADUser.AddADUsersToCMSByUsername Method

Adds Active Directory (see page 189) 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. |

## 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 set to true, updates the user group with the new group information. Otherwise, nothing is updated. |

**Returns**

Ektron.Cms.AdDeSyncGroupData()

## ADUser.DeSynchUsers Method

Update the user with new user details.

**C#**

```
public AdDeSyncUserData[] DeSynchUsers(Boolean Flag);
```

**Visual Basic**

```
Public Function DeSynchUsers(ByVal Flag As Boolean) As AdDeSyncUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Flag | If set to true, updates the user with new user information. Otherwise, nothing is updated. |

**Returns**

Ektron.Cms.AdDeSyncUserData()

## ADUser.GetADMapping Method

Loads the CMS and Active Directory (see page 189) name.

**C#**

```
public Overloads AdMappingData[] GetADMapping(Long CallerId, String MapType, int Map, int
Exclude, int Editable);
```

**Visual Basic**

```
Public Overloads Function GetADMapping(ByVal CallerId As Long, ByVal MapType As String, ByVal
Map As Integer, ByVal Exclude As Integer, ByVal Editable As Integer) As AdMappingData()
```

**Parameters**

| Parameters | Description |
|---|---|
| CallerId | The ID of the 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 (see page 1044) (0 – false, 1- true). |
| Editable | Flag (see page 1044) (0 - false, 1 - true). |

**Returns**

Ektron.Cms.AdMappingData()

### ADUser.GetADStatus Method

Loads the required Synch, Desynch and relationship flags of the configured Active Directory (see page 189).

**C#**

```
public AdSyncData GetADStatus();
```

**Visual Basic**

```
Public Function GetADStatus() As AdSyncData
```

**Returns**

Ektron.Cms.AdSyncData

### ADUser.GetAvailableADGroups Method

Loads all of the Active Directory (see page 189) groups for a given domain and filter.

**C#**

```
public GroupData[] GetAvailableADGroups(String Filter, String Domain);
```

**Visual Basic**

```
Public Function GetAvailableADGroups(ByVal Filter As String, ByVal Domain As String) As
GroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Filter | Filter by complete or a specific group name. |
| Domain | The domain name. (Required) |

**Returns**

Ektron.Cms.GroupData()

### ADUser.GetAvailableADUsers Method

Loads all available Active Directory (see page 189) 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 (see page 1419)()

### 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()

### ADUser.GetCMSRelationshipsToSync Method

Loads all the CMS relationships that need to be synchronized with Active Directory (see page 189).

**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()

### ADUser.GetCMSUsersToSync Method

Loads all the CMS users that need to be synchronized with Active Directory (see page 189).

**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 (see page 1419)()

### ADUser.GetDomains Method

Loads a list of domains from Active Directory (see page 189).

**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 (see page 189). Enter **1** to check integration. Otherwise, the method continues without checking. |

**Returns**

Ektron.Cms.DomainData()

### ADUser.MapCMSUserGroupToAD Method

Updates the given CMS group corresponding to the Active Directory (see page 189) group.

**C#**

```
public MapCMSUserGroupToAD(Long GroupId, String GroupName, String Domain);
```

**Visual Basic**

```
Public Sub MapCMSUserGroupToAD(ByVal GroupId As Long, ByVal GroupName As String, ByVal Domain
As String)
```

**Parameters**

| Parameters | Description |
|---|---|
| GroupId | The CMS group ID to update. |
| GroupName | The CMS group name to update. |
| Domain | The Active Directory (see page 189) domain. |

### ADUser.MapCMSUserToAD Method

Maps the given user ID to the corresponding Active Directory (see page 189) entry.

**C#**

```
public MapCMSUserToAD(Long UserId, String UserName, String Domain, int PullInfo);
```

**Visual Basic**

```
Public Sub MapCMSUserToAD(ByVal UserId As Long, ByVal UserName As String, ByVal Domain As
String, ByVal PullInfo As Integer)
```

**Parameters**

| Parameters | Description |
|---|---|
| UserId | The user's ID to map. |
| UserName | The user's name to map. |
| Domain | The user's domain. |
| PullInfo | Flag (see page 1044) to load Active Directory (see page 189) information. (1 = Load Active Directory (see page 189) information, otherwise none.) |

## ADUser.SynchCMSGroupsToAD Method

Synchronizes the CMS group data to the Active Directory (see page 189) group data. This updates the CMS data with the corresponding to Active Directory (see page 189) 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”. |

## ADUser.SynchCMSRelationshipsToAD Method

Synchronizes the CMS relations, such as groups, to the Active Directory (see page 189) 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”). |

## ADUser.SynchCMSUsersToAD Method

Synchronize the CMS user data to the Active Directory (see page 189) 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 (see page 189) information when the value is 1, otherwise none. |

### ADUser.UpdateADMapping Method

Updates the Active directory information into the CMS based on the user ID and user data.

**C#**

```
public UpdateADMapping(Long UserId, Object pagedata);
```

**Visual Basic**

```
Public Sub UpdateADMapping(ByVal UserId As Long, ByVal pagedata As Object)
```

**Parameters**

| Parameters | Description |
|---|---|
| UserId | The user's ID. |
| pagedata | Key, value pair for given user ID. |

### ADUser.UpdateADUsersGroups Method

Updates Active Directory (see page 189) 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. |

### ADUser.UpdateUsersInfoFromAD Method

Updates the user details from the Active Directory (see page 189) details.

**C#**

```
public UpdateUsersInfoFromAD(String UserName, String Domain);
```

**Visual Basic**

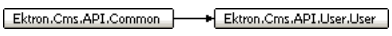
```
Public Sub UpdateUsersInfoFromAD(ByVal UserName As String, ByVal Domain As String)
```

**Parameters**

| Parameters | Description |
|---|---|
| UserName | The user's username. |
| Domain | The user's Domain. |

## User Class

**Class Hierarchy**




**C#**

```
public class User : Ektron.Cms.API.Common;
```

**Visual Basic**

```
Public Class User
Inherits Ektron.Cms.API.Common
```

**File**

User.vb

**Description**

This is class Ektron.Cms.API.User.User.

## User Methods

The methods of the User class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | ActivateUserAccount (see page 922) | This is the overview for the ActivateUserAccount method overload. |
| | AddMembershipUser (see page 923) | Adds a new membership user to CMS. |
| | AddNewUser (see page 925) | **This method is obsolete. Please use the AddMembershipUser (see page 923) method.** Adds a new membership user to CMS. |
| | AddRolePermission (see page 926) | Adds permission for the given role. |
| | AddUser (see page 926) | This is the overview for the AddUser method overload. |
| | AddUserGroup (see page 927) | Creates new user group. |
| | AddUserToGroup (see page 927) | Adds an existing user to an existing group. |
| | AutoLogInUser (see page 928) | Login (see page 337) the given user credentials. |
| | CreateBlankUserObject (see page 928) | This is the overview for the CreateBlankUserObject method overload. |
| | DeleteUserByID (see page 929) | 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 929) | 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 929) | Removes a user from the designated group. |
| | DeleteUserGroup (see page 930) | Removes the group from the system permanently. |
| | EditUserCustomProperties (see page 930) | This method creates the editable fields(UI) for user's custom properties. |
| | GetActiveUser (see page 930) | Returns all of the user properties with or without preference. |
| | GetADMapping (see page 931) | Loads the CMS and Active Directory (see page 189) name. |
| | GetAllCmsUsers (see page 931) | Loads all the CMS users. |
| | GetAllCustomProperties (see page 931) | Returns array of all the custom properties defined in the system. |
| | GetAllUserByCustomProperty (see page 932) | Returns all the users based on the given properties. |
| | GetAllUserGroups (see page 932) | This is the overview for the GetAllUserGroups method overload. |
| | GetAllUsers (see page 933) | This is the overview for the GetAllUsers method overload. |
| | GetGroupsUserIsIn (see page 934) | Loads all the groups by user. |
| | GetMembershipUser (see page 934) | Returns user data based on the user's login name (username). |
| | GetUser (see page 935) | Loads the user details for the given user. |
| | GetUserByUsername (see page 935) | Loads the user details for the given user. |

| | | |
|---|---|---|
| | GetUserCustomProperties (see page 937) | Returns an array of custom properties for a given user's ID. |
| | GetUserEmailInfo (see page 937) | Returns user data for a given user's ID. |
| | GetUserForFolderAdmin (see page 937) | |
| | GetUserGroup (see page 937) | Loads a given group's details. |
| | GetUserGroupByName (see page 938) | Loads a given group's details. |
| | GetUserGroupForFolderAdmin (see page 938) | Returns all the group's data for the administrator. |
| | GetUserPermissions (see page 938) | Loads the user permissions as a array of UserPermissionData. |
| | GetUserPreference (see page 939) | Loads the preferences for the given user. |
| | GetUsers (see page 939) | This is the overview for the GetUsers method overload. |
| | GetUsersForTask (see page 940) | Loads the users that have permissions to the given content. |
| | GetUsersNotInGroup (see page 941) | Loads all the users that are not in the given group. |
| | IsAdmin (see page 941) | Determines the if the current user is an Administrator. |
| | IsAGroupMember (see page 942) | For a given userId, returns whether that user is a member of the group. |
| | IsARoleMember (see page 942) | This is the overview for the IsARoleMember method overload. |
| | IsARoleMemberForContent (see page 945) | Determines whether a Role Member is allowed to work with a given piece of content as an administrator or not. Administrations privileges are for the given piece of content only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. |
| | IsARoleMemberForContent_FolderUserAdmin (see page 945) | 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 946) | 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 946) | 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 946) | Determines whether a Role Member is allowed to work with a given folder as an meta data administrator or not. |
| | LogInUser (see page 947) | This is the overview for the LogInUser method overload. |
| | ReadCustomProperties (see page 950) | Returns the custom properties as a Visual Basic collection. Importing the Microsoft.VisualBasic name space might be required. |
| | ResetMembershipUserPassword (see page 950) | This is the overview for the ResetMembershipUserPassword method overload. |
| | SetAuthenticationCookie (see page 951) | This method sets the authentication cookie. |
| | UnsubscribeUser (see page 951) | Unsubscribes a user from the system. |
| | UpdateCustomPropertiesItemOrder (see page 952) | Updates the order of available custom properties. |
| | UpdateUser (see page 952) | This is the overview for the UpdateUser method overload. |
| | UpDateUserGroup (see page 953) | This is the overview for the UpDateUserGroup method overload. |
| | UpdateUserPreferences (see page 954) | Updates the given user's preference data. |
| | UserObject (see page 954) | This is the overview for the UserObject method overload. |

## ActivateUserAccount Method

This is the overview for the ActivateUserAccount method overload.

### Description

Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID)

parameters.

This method is overloaded

**Overload List**

| | Name | Description |
|---|---|---|
| | User.ActivateUserAccount (Long) (see page 923) | Activates a deleted (deactivated) user into the current/active state using the UserId.<br>This function is overloaded. See Also: ActivateUserAccount (String String) |
| | User.ActivateUserAccount (String, String) (see page 923) | Activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter.<br>This function is overloaded. See Also: ActivateUserAccount (Long) |

## User.ActivateUserAccount Method (Long)

Activates a deleted (deactivated) user into the current/active state using the UserId.

This function is overloaded. See Also: ActivateUserAccount (String String)

**C#**

```
public Boolean ActivateUserAccount(Long UserId);
```

**Visual Basic**

```
Public Function ActivateUserAccount(ByVal UserId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (Long)

**C#**

```
public Boolean ActivateUserAccount(String Username, String AccountID);
```

**Visual Basic**

```
Public Function ActivateUserAccount(ByVal Username As String, ByVal AccountID As String) As
Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 1419)

**Example**

The following example shows how to create a Web page from which a new membership user can be added. This example uses some standard drag and drop controls and a small section of VB code utilizing the AddMembershipUser method. This method uses the InternalAdmin to add the Membership (see page 350) User (see page 920). In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, create a table with four rows and three columns.
2. In the first row, first column, add a label that shows where to enter the membership user's Email address.
```
<asp:Label ID="Label6" runat="server" Text="Email: " Width="76px"></asp:Label>
```
3. In the first row, second column, add a text box to enter the Email address.
```
<asp:TextBox ID="txtEmail" runat="server" Width="144px"></asp:TextBox>
```
4. In the first row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the Email address box.
```
<asp:RequiredFieldValidator ID="rfvEmail" runat="server" ErrorMessage="Please enter an
Email address"
  ControlToValidate="txtEmail" Display="Dynamic">Please enter email
address</asp:RequiredFieldValidator>
```
5. In the same table cell, add a RegularExpressionValidator control. This validates the information entered is in the format of an Email address.
```
<asp:RegularExpressionValidator ID="rfvEmailFormat" ControlToValidate="txtEmail"
runat="server" ErrorMessage="Please enter a valid email address"
  ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*" Width="236px"
></asp:RegularExpressionValidator>
```
6. In the second row, first column, add a label to show where to enter a first name.
```
<asp:Label ID="Label4" runat="server" Text="First name: " Width="76px"></asp:Label>
```
7. In the second row, second column, add a text box to enter the fist name.
```
<asp:TextBox ID="txtFirstName" runat="server" Width="144px"></asp:TextBox>
```
8. In the second row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the first name box.
```
<asp:RequiredFieldValidator ID="rfvFirstName" runat="server" ErrorMessage="Please enter
first name"
  ControlToValidate="txtFirstName">Please enter first name</asp:RequiredFieldValidator>
```
9. In the third row, first column, add a label to show where to enter a last name.
```
<asp:Label ID="Label7" runat="server" Text="Last Name:" Width="76px"></asp:Label>
```
10. In the third row, second column, add a text box to enter the last name.
```
<asp:TextBox ID="txtLastName" runat="server" Width="144px"></asp:TextBox>
```
11. In the third row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the last name box.
```
<asp:RequiredFieldValidator ID="rfvLastName" runat="server" ErrorMessage="Please enter last
name"
  ControlToValidate="txtLastName">Please enter last name</asp:RequiredFieldValidator>
```
12. In the fourth row, first column, add a label to show where to enter a password.
```
<asp:Label ID="Label8" runat="server" Text="Password: " Width="76px"></asp:Label>
```
13. In the fourth row, second column, add a text box to enter the password.
```
<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
Width="144px"></asp:TextBox>
```

14. In the fourth row, third column, add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the password box.

```
<asp:RequiredFieldValidator ID="rfvPassword" runat="server" ErrorMessage="Please enter password"
    ControlToValidate="txtPassword">Please enter password</asp:RequiredFieldValidator>
```

15. After the table, add a break and then add a button to submit the membership user's information.

```
<br />
<asp:Button ID="btnAdd" runat="server" Text="Add Membership User" Width="153px" />
```

16. Add two breaks for spacing purposes.

```
<br /><br />
```

17. Add a results label to display a message stating the membership user was added and the new ID.

```
<asp:Label ID="lblResult" runat="server"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add button click event.

```
    Protected Sub btnAdd_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles btnAdd.Click
```

2. Create an object for the user data items.

```
        Dim memDetails As New Ektron.Cms.UserData
```

3. Create an object for the new membership user information.

```
        Dim newMemUser As New Ektron.Cms.UserData
```

4. Create an object for the User (see page 920) API (see page 639).

```
        Dim userAPI As New Ektron.Cms.API.User.User
```

5. Create an integer place holder for the new ID.

```
        Dim newMemUserId As Integer
```

6. Populate memDetails with the information from the text boxes.

```
        memDetails.Username = txtEmail.Text
        memDetails.FirstName = txtFirstName.Text
        memDetails.LastName = txtLastName.Text
        memDetails.Password = txtPassword.Text
```

7. Create a Try/Catch around the AddMembershipUser method. This allows you to catch any exceptions and helps when debugging.

```
        Try
            newMemUser = userAPI.AddMembershipUser(memDetails)
```

8. Assign newMemUserId a new ID.

```
            newMemUserId = newMemUser.Id
```

9. Check to see if there is an ID returned. If so, display a message with the membership username and ID.

```
        If (newMemUserId > 0) Then
            lblResult.Text = "The membership user " & txtEmail.Text & " was added into CMS400.Net with UserID = " & newMemUserId
        End If
```

10. If there is an error, let the user know what happened.

```
        Catch ex As Exception
            lblResult.Text = ex.Message
        End Try

    End Sub
```

### User.AddNewUser Method

**This method is obsolete. Please use the AddMembershipUser (see page 923) method.**

Adds a new membership user to CMS.

**C#**

```
[Obsolete("Use ""AddMembershipUser"" instead.")]
public UserData AddNewUser(UserData udUserData);
```

**Visual Basic**

```
<Obsolete("Use ""AddMembershipUser"" instead.")>
Public Function AddNewUser(ByVal udUserData As UserData) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| udUserData | An object of class which represents the corresponding user elements. |

**Returns**

Ektron.Cms.UserData (see page 1419)

## 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

This is the overview for the AddUser method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.AddUser (Collection) (see page 926) | Adds a new user to CMS. |
| | User.AddUser (UserData) (see page 927) | 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 (see page 920) details as a value, key pair. |

**Returns**

Ektron.Cms.UserData (see page 1419)

## 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 |
|---|---|
| udUserData | User (see page 920) details as a value strongly typed UserData (see page 1419) object. |

**Returns**

Ektron.Cms.UserData (see page 1419)

## User.AddUserGroup Method

Creates new user group.

**C#**

```
public Collection AddUserGroup(String GroupName, Optional ByVal String GroupPath, Optional
ByVal String GroupDomain);
```

**Visual Basic**

```
Public Function AddUserGroup(ByVal GroupName As String, Optional ByVal GroupPath As String =
"", Optional ByVal GroupDomain As String = "") As Collection
```

**Parameters**

| Parameters | Description |
|---|---|
| GroupName | The name of the group. |
| GroupPath | Group path, usually required for Active Directory (see page 189), otherwise CMS group assumed. |
| GroupDomain | Group domain, usually required for Active Directory (see page 189). |

## User.AddUserToGroup Method

Adds an existing user to an existing group.

**C#**

```
public AddUserToGroup(Long UserId, Long GroupId);
```

**Visual Basic**

```
Public Sub AddUserToGroup(ByVal UserId As Long, ByVal GroupId As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| UserId | The user's ID. |
| GroupId | The required group's ID. |

## User.AutoLogInUser Method

Login (see page 337) 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 is the overview for the CreateBlankUserObject method overload.

**Description**

This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.CreateBlankUserObject () (see page 928) | Returns the instance of the user data.<br>This function is overloaded. See Also: CreateBlankUserObject (UserData (see page 1419)) |
| | User.CreateBlankUserObject (UserData) (see page 929) | Sets the given user data values into the UserData (see page 1419) class.<br>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 (see page 1419))

**C#**

```
public UserData CreateBlankUserObject();
```

**Visual Basic**

```
Public Function CreateBlankUserObject() As UserData
```

**Returns**

Ektron.Cms.UserData (see page 1419)

### User.CreateBlankUserObject Method (UserData)

Sets the given user data values into the UserData (see page 1419) class.

This function is overloaded. See Also: CreateBlankUserObject (see page 928)

**C#**

```
public CreateBlankUserObject(ref UserData udUserData);
```

**Visual Basic**

```
Public Sub CreateBlankUserObject(ByRef udUserData As UserData)
```

**Parameters**

| Parameters | Description |
|---|---|
| udUserData | An Object of user properties. |

**Remarks**

### User.DeleteUserByID Method

Removes the user from the system. This will not purge the data, but it disables the user from further accessing the system.

**C#**

```
public DeleteUserByID(Long Id);
```

**Visual Basic**

```
Public Sub DeleteUserByID(ByVal Id As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| 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(Long UserId, Long GroupId, Optional ByVal Boolean ReturnDeleted);
```

**Visual Basic**

```
Public Sub DeleteUserFromGroup(ByVal UserId As Long, ByVal GroupId As Long, Optional ByVal
ReturnDeleted As Boolean = False)
```

**Parameters**

| Parameters | Description |
|---|---|
| UserId | The user's ID. |
| GroupId | The group's ID. |

## User.DeleteUserGroup Method

Removes the group from the system permanently.

**C#**

```
public DeleteUserGroup(Long Id);
```

**Visual Basic**

```
Public Sub DeleteUserGroup(ByVal Id As Long)
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The group's ID. |

## User.EditUserCustomProperties Method

This method creates the editable fields(UI) for user's custom properties.

**C#**

```
public String EditUserCustomProperties(Optional ByVal Long UserId, Optional ByVal Boolean
bReadOnly);
```

**Visual Basic**

```
Public Function EditUserCustomProperties(Optional ByVal UserId As Long = 0, Optional ByVal
bReadOnly As Boolean = False) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| UserId | User (see page 920)'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(Long Id, Optional ByVal Boolean Preference);
```

**Visual Basic**

```
Public Function GetActiveUser(ByVal Id As Long, Optional ByVal Preference As Boolean = False)
As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The user's ID. |
| Preference | Preference is returned only if the value true, otherwise nothing is returned for preference. |

**Returns**

Ektron.Cms.UserData (see page 1419)

## User.GetADMapping Method

Loads the CMS and Active Directory (see page 189) name.

**C#**

```
public AdMappingData[] GetADMapping(Long CallerId, String MapType, int Map, int Exclude, int Editable);
```

**Visual Basic**

```
Public Function GetADMapping(ByVal CallerId As Long, ByVal MapType As String, ByVal Map As Integer, ByVal Exclude As Integer, ByVal Editable As Integer) As AdMappingData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 1044) (0 – false, 1-true). |
| Editable | Flag (see page 1044) (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.<br>The string value can be:<br>• username<br>• firstname<br>• lastname<br>• userid<br>• language<br>• datecreated<br>• lastlogindate |

**Returns**

Ektron.Cms.UserData (see page 1419)()

**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

This is the overview for the GetAllUserGroups method overload.

**Description**

Loads all of the user groups from the system. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.GetAllUserGroups (Ektron.Cms.Common.EkEnumeration.UserTypes, String) (see page 932) | Loads all of the user groups from the system.<br>This function is overloaded. See Also: GetAllUserGroups (String) |
| | User.GetAllUserGroups (String) (see page 933) | Loads all of the user groups from the system.<br>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 350); 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 is the overview for the GetAllUsers method overload.

**Description**

This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.GetAllUsers (Long, Long) (see page 933) | Gets the User (see page 920) Group Data of all the allowed users in the folder specified. |
| | User.GetAllUsers (UserRequestData) (see page 934) | Returns the list of users from the database based on the UserRequestData (see page 1424) class. |

## User.GetAllUsers Method (Long, Long)

Gets the User (see page 920) Group Data of all the allowed users in the folder specified.

**C#**

```
public UserGroupData[] GetAllUsers(Long CallerID, Long FolderId);
```

**Visual Basic**

```
Public Function GetAllUsers(ByVal CallerID As Long, ByVal FolderId As Long) As UserGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 1424) class.

**C#**

```
public UserData[] GetAllUsers(ref 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 (see page 1419).

### User.GetGroupsUserIsIn Method

Loads all the groups by user.

**C#**

```
public GroupData[] GetGroupsUserIsIn(Long UserId, String OrderBy);
```

**Visual Basic**

```
Public Function GetGroupsUserIsIn(ByVal UserId As Long, ByVal OrderBy As String) As GroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 920)'s login name (username). This is required |

### Returns

Ektron.Cms.UserData (see page 1419)

## User.GetUser Method

Loads the user details for the given user.

### C#

```
public UserData GetUser(Long Id, Optional ByVal Boolean Preference, Optional ByVal Boolean
ReturnDeleted);
```

### Visual Basic

```
Public Function GetUser(ByVal Id As Long, Optional ByVal Preference As Boolean = False,
Optional ByVal ReturnDeleted As Boolean = False) As UserData
```

### Parameters

| Parameters | Description |
|---|---|
| Id | The user ID. |
| Preference | Loads preference only if required (true = load preferences as a part of user data). |

### Returns

Ektron.Cms.UserData (see page 1419)

## 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 (see page 1419)

### Example

The following example shows how to create a Web page that retrieves user details for a given username. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetUserByUsername method. This method uses the InternalAdmin to retrieve the user details. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a label to show where to enter a username.
   `<asp:Label ID="Label1" runat="server" Text="Username: "></asp:Label>`
2. Add a text box to enter the username.
   `<asp:TextBox ID="txtUserName" runat="server" Width="63px"></asp:TextBox>`
3. add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the username box.

```
<asp:RequiredFieldValidator ID="rfvUserId" runat="server" ControlToValidate="txtUserName"
  ErrorMessage="Please enter Username"></asp:RequiredFieldValidator>
```

4. Add two breaks for spacing purposes.
   ```
   <br /><br />
   ```
5. Add a button to submit the username.
   ```
   <asp:Button ID="btnGetUser" runat="server" Text="Get Custom Properties" Width="148px" />
   ```
6. Add two more breaks for spacing purposes.
   ```
   <br /><br />
   ```
7. Add a label to display the given user's details.
   ```
   <asp:Label ID="lblResult" runat="server"></asp:Label>
   ```

Add the following information to the code behind of your aspx.vb page.

1. Add button click event.

```
    Protected Sub btnGetUser_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnGetUser.Click
```

2. Create an object for the user API (see page 639).

```
        Dim userAPI As New Ektron.Cms.API.User.User
```

3. Create an object for the user data.

```
        Dim userObj As Ektron.Cms.UserData
```

4. Create a Try/Catch around the GetUserByUsername method. This allows you to catch any exceptions and helps when debugging.

```
        Try
            userObj = userAPI.GetUserByUsername(txtUserName.Text.Trim)
```

5. Check to make sure the a user ID exists.

```
            If userObj.Id = 0 Then
               lblResult.Text = "Invalid Username"
               Exit Sub
            End If
```

6. Create an integer place holder.

```
            Dim i As Integer
```

7. Check to make sure the attribute list is populated.

```
            If Not userObj.AttributeList Is Nothing Then
```

8. If the list is populated, display the results.

```
               lblResult.Text = "<table> <th>Name</th> <th>Value</th> <th>Type</th>"
               For i = 0 To userObj.AttributeList.Length - 1 Step 1
                   lblResult.Text &= "<tr>"
                   lblResult.Text &= "<td>" & userObj.AttributeList(i).Name & "</td>"
                   lblResult.Text &= "<td>" & userObj.AttributeList(i).Value & "</td>"
                   lblResult.Text &= "<td>" &
userObj.AttributeList(i).ValueType.ToString() & "</td>"
                   lblResult.Text &= "</tr>"
               Next
               lblResult.Text &= "</table>"
```

9. If nothing is contained in the list, display a message.

```
            Else
               lblResult.Text &= "No custom properties set-up for this user"
           End If
```

10. If there is an error, let the user know what happened.

```
        Catch ex As Exception
           lblResult.Text = ex.Message
        End Try
    End Sub
```

### User.GetUserCustomProperties Method

Returns an array of custom properties for a given user's ID.

**C#**

```
public CustomAttribute[] GetUserCustomProperties(Long UserId);
```

**Visual Basic**

```
Public Function GetUserCustomProperties(ByVal UserId As Long) As CustomAttribute()
```

**Parameters**

| Parameters | Description |
|---|---|
| UserId | User (see page 920)'s ID |

**Returns**

Ektron.Cms.CustomAttribute()

### User.GetUserEmailInfo Method

Returns user data for a given user's ID.

**C#**

```
public UserData GetUserEmailInfo(Long Id);
```

**Visual Basic**

```
Public Function GetUserEmailInfo(ByVal Id As Long) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The user's ID. |

**Returns**

Object of UserData (see page 1419)

### User.GetUserForFolderAdmin Method

**C#**

```
public UserData GetUserForFolderAdmin(Long FolderId, Long Id, Optional ByVal Boolean
Preference, Optional ByVal Boolean ReturnDeleted);
```

**Visual Basic**

```
Public Function GetUserForFolderAdmin(ByVal FolderId As Long, ByVal Id As Long, Optional ByVal
Preference As Boolean = False, Optional ByVal ReturnDeleted As Boolean = False) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderId | |
| Id | |
| Preference | |

### User.GetUserGroup Method

Loads a given group's details.

**C#**

```
public UserGroupData GetUserGroup(Long Id);
```

**Visual Basic**

```
Public Function GetUserGroup(ByVal Id As Long) As UserGroupData
```

**Parameters**

| Parameters | Description |
|---|---|
| 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(Long FolderId, Long Id);
```

**Visual Basic**

```
Public Function GetUserGroupForFolderAdmin(ByVal FolderId As Long, ByVal Id As Long) As
UserGroupData
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderId | Folder (see page 652) ID |
| Id | User (see page 920) ID |

**Returns**

UserGroupData

### User.GetUserPermissions Method

Loads the user permissions as a array of UserPermissionData.

**C#**

```
public UserPermissionData[] GetUserPermissions(Long Id, String ItemType, Long UserId, String
UserGroupList, Optional ByVal PermissionUserType PermissionType, Optional ByVal
PermissionRequestType PermissionRequest);
```

**Visual Basic**

```
Public Function GetUserPermissions(ByVal Id As Long, ByVal ItemType As String, ByVal UserId As
Long, ByVal UserGroupList As String, Optional ByVal PermissionType As PermissionUserType =
PermissionUserType.All, Optional ByVal PermissionRequest As PermissionRequestType =
PermissionRequestType.All) As UserPermissionData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The item's ID (content or folder id whose permissions you are seeking). |
| ItemType | The item's type (valid values - content or folder). |
| UserId | The user's ID (when you are getting permissions for a group set this to zero). |
| UserGroupList | The group list (comma separated list of group ids, blank when you are getting permissions for a specific user). |
| PermissionType | PermissionUserType (Use a value from enum Ektron.Cms.Content.EkContent.PermissionResultType). |
| PermissionRequest | PermissionRequestType (Use a value from enum Ektron.Cms.Content.EkContent.PermissionRequestType). |

**Returns**

Ektron.Cms.UserPermissionData()

## User.GetUserPreference Method

Loads the preferences for the given user.

**C#**

```
public UserPreferenceData GetUserPreference(Long Id);
```

**Visual Basic**

```
Public Function GetUserPreference(ByVal Id As Long) As UserPreferenceData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The user's ID. |

**Returns**

Ektron.Cms.UserPreferenceData

## GetUsers Method

This is the overview for the GetUsers method overload.

**Description**

An Overloaded function that allows you to get the users from a specific group or load and array of user group data. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.GetUsers (Long, String) (see page 940) | Loads the all the users from the given group. |
| | User.GetUsers (String, String) (see page 940) | Returns an array of user group data based on the userName and userNameType. |

## User.GetUsers Method (Long, String)

Loads the all the users from the given group.

**C#**

```
public UserData[] GetUsers(Long GroupId, String OrderBy);
```

**Visual Basic**

```
Public Function GetUsers(ByVal GroupId As Long, ByVal OrderBy As String) As UserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| GroupId | The group's ID |
| OrderBy | The order of the users. The string value can be:<br>• username<br>• firstname<br>• lastname<br>• userid<br>• language<br>• datecreated<br>• lastlogindate |

**Returns**

- Ektron.Cms.UserData (see page 1419)()

**Description**

## User.GetUsers Method (String, String)

Returns an array of user group data based on the userName and userNameType.

**C#**

```
public UserGroupData[] GetUsers(String userNames, Optional ByVal String userNameTypes);
```

**Visual Basic**

```
Public Function GetUsers(ByVal userNames As String, Optional ByVal userNameTypes As String =
"") As UserGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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.

**C#**

```
public UserGroupData[] GetUsersForTask(Long CallerID, Long ContentID);
```

**Visual Basic**

```
Public Function GetUsersForTask(ByVal CallerID As Long, ByVal ContentID As Long) As
UserGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| CallerID | The user ID that is logged on to the system. |
| ContentID | The ID of the content. (Required) |

**Returns**

Ektron.Cms.UserGroupData()

## User.GetUsersNotInGroup Method

Loads all the users that are not in the given group.

**C#**

```
public UserData[] GetUsersNotInGroup(Long Id, String OrderBy, Optional ByVal int GroupType);
```

**Visual Basic**

```
Public Function GetUsersNotInGroup(ByVal Id As Long, ByVal OrderBy As String, Optional ByVal
GroupType As Integer = 0) As UserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The group ID. |
| OrderBy | The order of the users. The string value can be:<br>• username<br>• firstname<br>• lastname<br>• userid<br>• language<br>• datecreated<br>• lastlogindate |

**Returns**

Ektron.Cms.UserData (see page 1419)()

**Description**

## User.IsAdmin Method

Determines the if the current user is an Administrator.

**C#**

```
public Boolean IsAdmin();
```

**Visual Basic**

```
Public Function IsAdmin() As Boolean
```

**Returns**

True if the logged in user is administrator. Otherwise, returns false.

## User.IsAGroupMember Method

For a given userId, returns whether that user is a member of the group.

**C#**

```
public Boolean IsAGroupMember(Long userId, Long groupId);
```

**Visual Basic**

```
Public Function IsAGroupMember(ByVal userId As Long, ByVal groupId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | Long value of the user Id. |
| groupId | Long value of the group Id. |

**Returns**

Boolean of whether the user is a member of the group.

## IsARoleMember Method

This is the overview for the IsARoleMember method overload.

**Description**

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) (see page 943) | Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.<br>This function is overloaded. See Also:<br>• IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean)<br>• IsARoleMember (Long Long Boolean)<br>• IsARoleMember (Long) |
| | User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Long, Boolean) (see page 943) | Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.<br>This function is overloaded. See Also:<br>• IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)<br>• IsARoleMember (Long Long Boolean)<br>• IsARoleMember (Long) |

| | | |
|---|---|---|
| | User.IsARoleMember (Long) (see page 944) | Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.<br>This function is overloaded. See Also:<br>• IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)<br>• IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean)<br>• IsARoleMember (Long Long Boolean) |
| | User.IsARoleMember (Long, Long, Boolean) (see page 944) | Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.<br>This function is overloaded. See Also:<br>• IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean)<br>• IsARoleMember (Long)<br>• IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) |

## User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean)
- IsARoleMember (Long Long Boolean)
- IsARoleMember (Long)

**C#**

```
public Boolean IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId);
```

**Visual Basic**

```
Public Function IsARoleMember(ByVal RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds) As
Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | _nt_ |

**Remarks**

## User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Long, Boolean)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- IsARoleMember (Long Long Boolean)
- IsARoleMember (Long)

**C#**

```
public Boolean IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId, Long nUserId,
Optional ByVal Boolean GroupFlag);
```

**Visual Basic**

```
Public Function IsARoleMember(ByVal RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds,
ByVal nUserId As Long, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | _nt_ |
| nUserId | _nt_ |
| GroupFlag | _nt_ |

**Remarks**

## User.IsARoleMember Method (Long)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean)
- IsARoleMember (Long Long Boolean)

**C#**

```
public Boolean IsARoleMember(Long RoleId);
```

**Visual Basic**

```
Public Function IsARoleMember(ByVal RoleId As Long) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | _nt_ |

**Remarks**

## User.IsARoleMember Method (Long, Long, Boolean)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Long Boolean)
- IsARoleMember (Long)
- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)

**C#**

```
public Boolean IsARoleMember(Long RoleId, Long nUserId, Optional ByVal Boolean GroupFlag);
```

**Visual Basic**

```
Public Function IsARoleMember(ByVal RoleId As Long, ByVal nUserId As Long, Optional ByVal
GroupFlag As Boolean = False) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | _nt_ |
| nUserId | _nt_ |
| GroupFlag | _nt_ |

**Remarks**

## User.IsARoleMemberForContent Method

Determines whether a Role Member is allowed to work with a given piece of content as an administrator or not. Administrations privileges are for the given piece of content only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

**C#**

```
public Boolean IsARoleMemberForContent(Long RoleId, Long ContentId, Optional ByVal Long
nUserId, Optional ByVal Boolean GroupFlag);
```

**Visual Basic**

```
Public Function IsARoleMemberForContent(ByVal RoleId As Long, ByVal ContentId As Long,
Optional ByVal nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | |
| ContentId | The numeric content 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(Long ContentId, Optional ByVal Long
nUserId, Optional ByVal Boolean GroupFlag);
```

**Visual Basic**

```
Public Function IsARoleMemberForContent_FolderUserAdmin(ByVal ContentId As Long, Optional
ByVal nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | required content 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(Long RoleId, Long FolderId, Optional ByVal Long nUserId,
Optional ByVal Boolean GroupFlag);
```

**Visual Basic**

```
Public Function IsARoleMemberForFolder(ByVal RoleId As Long, ByVal FolderId As Long, Optional
ByVal nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| 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(Long FolderId, Optional ByVal Long
nUserId, Optional ByVal Boolean GroupFlag);
```

**Visual Basic**

```
Public Function IsARoleMemberForFolder_FolderUserAdmin(ByVal FolderId As Long, Optional ByVal
nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| 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(Long FolderId, Optional ByVal Long
nUserId, Optional ByVal Boolean GroupFlag);
```

**Visual Basic**

```
Public Function IsARoleMemberForFolder_MetadataAdmin(ByVal FolderId As Long, Optional ByVal
nUserId As Long = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderId | required folder id |
| nUserId | user id |
| GroupFlag | group id |

**Returns**

True if granted otherwise false

## LogInUser Method

This is the overview for the LogInUser method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.LogInUser (String, String, String) (see page 947) | This method allows you to log a user into a server by passing their username and password. |
| | User.LogInUser (String, String, String, String) (see page 948) | This method allows you to log a user into a server by passing their username, password, and server name. |
| | User.LogInUser (String, String, String, String, String) (see page 948) | This method allows you to log a user into a server by passing their username, password, and server name. You can also log in an Active Directory (see page 189) user by providing the username, password and domain. Note: When when using Active Directory (see page 189), the ServerName and Protocol parameters are set to empty strings (""). |

## User.LogInUser Method (String, String, String)

This method allows you to log a user into a server by passing their username and password.

**C#**

```
public UserData LogInUser(String username, String Password, String ServerName);
```

**Visual Basic**

```
Public Function LogInUser(ByVal username As String, ByVal Password As String, ByVal ServerName
As String) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| username | The system username. |
| Password | The system password. |
| ServerName | The system server name. When using Active Directory (see page 189), this parameter needs to be an empty string "". |

**Returns**

Ektron.Cms.UserData (see page 1419)

## User.LogInUser Method (String, String, String, String)

This method allows you to log a user into a server by passing their username, password, and server name.

**C#**

```
public UserData LogInUser(String username, String Password, String ServerName, String Domain);
```

**Visual Basic**

```
Public Function LogInUser(ByVal username As String, ByVal Password As String, ByVal ServerName
As String, ByVal Domain As String) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| username | The system username. |
| Password | The system password. |
| ServerName | The system server name. When using Active Directory (see page 189), this parameter needs to be an empty string "". |
| Domain | The user's domain, required only when users login using Active Directory (see page 189). If your domain is **intra.example.com**, it is entered in the following format: **"DC=intra,DC=example,DC=com"**. |

**Returns**

Ektron.Cms.UserData (see page 1419)

## User.LogInUser Method (String, String, String, String, String)

This method allows you to log a user into a server by passing their username, password, and server name. You can also log in an Active Directory (see page 189) user by providing the username, password and domain. Note: When when using Active Directory (see page 189), the ServerName and Protocol parameters are set to empty strings ("").

**C#**

```
public UserData LogInUser(String username, String Password, String ServerName, String Domain,
String Protocol);
```

**Visual Basic**

```
Public Function LogInUser(ByVal username As String, ByVal Password As String, ByVal ServerName
As String, ByVal Domain As String, ByVal Protocol As String) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| username | The system username. |
| Password | The system password. |
| ServerName | The system server name. When using Active Directory (see page 189), this parameter needs to be an empty string "". |
| Domain | The user's domain, required only when users login using Active Directory (see page 189). If your domain is **intra.example.com**, it is entered in the following format: **"DC=intra,DC=example,DC=com"**. |
| Protocol | This property is always set to an empty string "". This parameter is only required when using Active Directory (see page 189). |

**Returns**

Ektron.Cms.UserData (see page 1419)

**Example**

The following example shows how to create a Web page that programmatically logs a user into CMS400.NET. Please note that the code-behind in this example does not login a new user if someone is already logged into the Workarea. This example uses some standard drag and drop controls and a small section of VB code utilizing the LogInUser method. This method uses the InternalAdmin to log the user into CMS400.NET. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add a label to show where the Username is entered.
```
<asp:Label ID="Label6" runat="server" Text="Username: " Width="76px"></asp:Label>
```
2. Add a text box to enter the Username.
```
<asp:TextBox ID="txtUserName" runat="server" Width="134px"></asp:TextBox>
```
3. Add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the Username box.
```
<asp:RequiredFieldValidator ID="rfvUserName" runat="server" ErrorMessage="Please enter user
name"
  ControlToValidate="txtUserName">Please enter user name</asp:RequiredFieldValidator>
```
4. Add two breaks for spacing purposes.
```
<br /><br />
```
5. Add a label to show where the Password is entered.
```
<asp:Label ID="Label8" runat="server" Text="Password: " Width="76px"></asp:Label>
```
6. Add a text box to enter the Password.
```
<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"
Width="134px"></asp:TextBox>
```
7. Add a RequiredFieldValidator control from the Validation control menu. This validates that something appears in the password box.
```
<asp:RequiredFieldValidator ID="rfvPassword" runat="server" ErrorMessage="Please enter
password"
  ControlToValidate="txtPassword">Please enter password</asp:RequiredFieldValidator>
```
8. Add two breaks for spacing purposes.
```
<br /><br />
```
9. Add a button to submit the user's login information. We will add a button click event in the code behind.
```
<asp:Button ID="btnLogin" runat="server" Text="Login" Width="57px" />
```
10. Add two last breaks for spacing purposes.
```
<br /><br />
```
11. Add a results label to display a message that states whether the user is logged in.
```
<asp:Label ID="lblResult" runat="server" Width="432px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a page load event that checks to see if a user is logged into CMS400.NET.
```
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles
Me.Load
        Dim api As New Ektron.Cms.CommonApi
        If (api.RequestInformationRef.CallerId > 0) Then
            lblResult.Text = api.RequestInformationRef.LoggedInUsername & " is logged in."
& <"br/>"
        End If
    End Sub
```
2. Add a button click event.
```
    Protected Sub btnLogin_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnLogin.Click
```
3. Create an object for the user API (see page 639).
```
        Dim userAPI As New Ektron.Cms.API.User.User
```
4. Create an object for the common API (see page 639).
```
        Dim api As New Ektron.Cms.CommonApi
```
5. Create an object to store the user data.
```
        Dim userObj As UserData
```
6. Create a Try/Catch around the LogInUser method. This allows you to catch any exceptions and helps when debugging.
7. Check to see if the user is already logged in.
```
        Try
            If (api.RequestInformationRef.CallerId > 0) Then
                lblResult.Text = api.RequestInformationRef.LoggedInUsername & " is already
logged in."
```

8. If the user is not logged in, proceed with the login.

```
            Else
                userObj = userAPI.LogInUser(txtUserName.Text.Trim, txtPassword.Text.Trim,
"localhost", "", "")
```

9. Check to see whether the login was successful. If it was unsuccessful, display a message.

```
                If userObj.Id = 0 Then
                    lblResult.Text = "Login failed."
                    Exit Sub
                End If
```

10. Set the authentication cookie and Response.Redirect.

```
                userAPI.SetAuthenticationCookie(userObj)
                Response.Redirect(Request.Url.ToString)
            End If
```

11. If there is an error, let the user know what happened.

```
        Catch ex As Exception
            lblResult.Text = ex.Message
        End Try
    End Sub
```

### User.ReadCustomProperties Method

Returns the custom properties as a Visual Basic collection. Importing the Microsoft.VisualBasic name space might be required.

**C#**

```
public Collection ReadCustomProperties(Collections.Specialized.NameValueCollection form);
```

**Visual Basic**

```
Public Function ReadCustomProperties(ByVal form As
Collections.Specialized.NameValueCollection) As Collection
```

**Parameters**

| Parameters | Description |
|---|---|
| form | A collection of user properties. |

**Returns**

Microsoft.VisualBasic.Collection (see page 88)

### ResetMembershipUserPassword Method

This is the overview for the ResetMembershipUserPassword method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.ResetMembershipUserPassword (String) (see page 950) | Reset the password for the given membership user. |
| | User.ResetMembershipUserPassword (String, String) (see page 951) | Reset the password for the given membership user. |

### User.ResetMembershipUserPassword Method (String)

Reset the password for the given membership user.

**C#**

```
public String ResetMembershipUserPassword(String Username);
```

**Visual Basic**

```
Public Function ResetMembershipUserPassword(ByVal Username As String) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| Username | User (see page 920)'s username to be reset. |

**Returns**

String

### User.ResetMembershipUserPassword Method (String, String)

Reset the password for the given membership user.

**C#**

```
public String ResetMembershipUserPassword(String Username, String AccountId);
```

**Visual Basic**

```
Public Function ResetMembershipUserPassword(ByVal Username As String, ByVal AccountId As
String) As String
```

**Parameters**

| Parameters | Description |
|---|---|
| Username | User (see page 920)'s username to be reset. |
| AccountId | If verification is turned on, specify verification id(email to a user). |

**Returns**

String

### User.SetAuthenticationCookie Method

This method sets the authentication cookie.

**C#**

```
public SetAuthenticationCookie(UserData userInfo);
```

**Visual Basic**

```
Public Sub SetAuthenticationCookie(ByVal userInfo As UserData)
```

**Parameters**

| Parameters | Description |
|---|---|
| userInfo | Returned value of the logInUser method. |

**Remarks**

First, you must call the logInUser method to login the user.

### User.UnsubscribeUser Method

Unsubscribes a user from the system.

**C#**

```
public Boolean UnsubscribeUser(String Username, Optional ByVal String Password);
```

**Visual Basic**

```
Public Function UnsubscribeUser(ByVal Username As String, Optional ByVal Password As String =
"") As Boolean
```

### Parameters

| Parameters | Description |
|---|---|
| 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

This is the overview for the UpdateUser method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | User.UpdateUser (Collection) (see page 952) | Updates the user details. |
| | User.UpdateUser (UserData) (see page 953) | Updates the user details. If you do not wish to update the user's password, leave the UserData.Password property blank. |

## User.UpdateUser Method (Collection)

Updates the user details.

### C#

```
public UpdateUser(Collection data);
```

### Visual Basic

```
Public Sub UpdateUser(ByVal data As Collection)
```

### Parameters

| Parameters | Description |
|---|---|
| data | A collection of user data. |

## User.UpdateUser Method (UserData)

Updates the user details. If you do not wish to update the user's password, leave the UserData.Password property blank.

**C#**

```
public UpdateUser(UserData data);
```

**Visual Basic**

```
Public Sub UpdateUser(ByVal data As UserData)
```

**Parameters**

| Parameters | Description |
|---|---|
| data | A UserData (see page 1419) object of user data. |

## UpDateUserGroup Method

This is the overview for the UpDateUserGroup method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.UpDateUserGroup (Collection) (see page 953) | Updates the user group details. |
| | User.UpDateUserGroup (UserData) (see page 953) | 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 (see page 1419)) |

## User.UpdateUserPreferences Method

Updates the given user's preference data.

**C#**

```
public UpdateUserPreferences(Long Id, Collection data);
```

**Visual Basic**

```
Public Sub UpdateUserPreferences(ByVal Id As Long, ByVal data As Collection)
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The user's ID. |
| data | A collection of user preferences. |

## UserObject Method

This is the overview for the UserObject method overload.

**Description**

Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.UserObject (Long) (see page 954) | Returns user data based on the user's ID.<br>This function is overloaded. See Also: UserObject (String) |
| | User.UserObject (String) (see page 954) | Returns user data based on the user's login or username.<br>This function is overloaded. See Also: UserObject (Long) |

## User.UserObject Method (Long)

Returns user data based on the user's ID.

This function is overloaded. See Also: UserObject (String)

**C#**

```
public UserData UserObject(Long Id);
```

**Visual Basic**

```
Public Function UserObject(ByVal Id As Long) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The user's ID. |

**Returns**

Ektron.Cms.UserData (see page 1419)

## User.UserObject Method (String)

Returns user data based on the user's login or username.

This function is overloaded. See Also: UserObject (Long)

**C#**

```
public UserData UserObject(String sUserName);
```

**Visual Basic**

```
Public Function UserObject(ByVal sUserName As String) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| sUserName | User (see page 920)'s login or username. |

**Returns**

Ektron.Cms.UserData (see page 1419)

### User Properties

The properties of the User class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | UserIdAuthenticated (see page 955) | Returns the id of the currently logged in user. |

### User.UserIdAuthenticated Property

Returns the id of the currently logged in user.

**C#**

```
public ReadOnly int UserIdAuthenticated;
```

**Visual Basic**

```
Public ReadOnly Property UserIdAuthenticated() As Integer
```

**Returns**

Boolean

# 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 955) | This is namespace Ektron.Services. |

# Ektron.Services Namespace

This is namespace Ektron.Services.

### Namespaces

| Name | Description |
|---|---|
| Task (see page 956) | This is namespace Ektron.Services.Task. |

### Classes

| | Name | Description |
|---|---|---|
| | ActiveDirectoryUser (see page 974) | Summary description for ActiveDirectoryUser |
| | Asset (see page 982) | Summary description for Asset |
| | Blog (see page 985) | Summary description for Blog |
| | BusinessRules (see page 999) | Summary description for BusinessRules |
| | CommunityGroup (see page 1001) | Summary description for CommunityGroup |
| | Content (see page 1016) | Summary description for ContentService |
| | ContentRating (see page 1029) | Summary description for ContentRating |
| | CustomFields (see page 1031) | Summary description for CustomFields |
| | Favorites (see page 1039) | Summary description for Favorites |
| | Flag (see page 1044) | The summary description for  Taxonomy (see page 1151). |
| | Folder (see page 1048) | Summary description for Folder |
| | Font (see page 1055) | Summary description for Font |
| | Form (see page 1058) | Summary description for Form |
| | Friends (see page 1066) | Summary description for Friends |
| | Library (see page 1078) | Summary description for Library |
| | MessageBoard (see page 1086) | Summary description for MessageBoard |
| | Metadata (see page 1106) | Summary description for Metadata |
| | Permissions (see page 1110) | Summary description for Permissions |
| | SearchManager (see page 1115) | Summary description for SearchManager |
| | Site (see page 1120) | Summary description for Site |
| | SiteMap (see page 1127) | Summary description for SiteMap |
| | Tags (see page 1130) | Summary description for Tags |
| | Taxonomy (see page 1151) | The summary description for Taxonomy. |
| | ThreadedDiscussion (see page 1157) | Summary description for ThreadedDiscussion |
| | User (see page 1167) | Summary description for User |

# Ektron.Services.Task Namespace

This is namespace Ektron.Services.Task.

### Classes

| | Name | Description |
|---|---|---|
| | Task (see page 957) | Summary description for Task |
| | TaskCategory (see page 968) | Summary description for TaskCategory |
| | TaskCategoryType (see page 971) | Summary description for TaskCategoryType |

# Classes

The following table lists classes in this documentation.

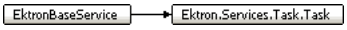
### Classes

| | Name | Description |
|---|---|---|
| | Task (see page 957) | Summary description for Task |
| | TaskCategory (see page 968) | Summary description for TaskCategory |
| | TaskCategoryType (see page 971) | Summary description for TaskCategoryType |

## Task Class

Summary description for Task

### Class Hierarchy




### C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Task : EktronBaseService;
```

### Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Task
Inherits EktronBaseService
```

### File

Task.cs

## Task Methods

The methods of the Task class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddTask (see page 957) | Adds a task in the system. |
| | AssignTaskToUser (see page 958) | This is the overview for the AssignTaskToUser method overload. |
| | AssignTaskToUsergroup (see page 959) | This is the overview for the AssignTaskToUsergroup method overload. |
| | DeleteTask (see page 960) | This is the overview for the DeleteTask method overload. |
| | DeleteTasks (see page 961) | This is the overview for the DeleteTasks method overload. |
| | GetTask (see page 962) | Get the task information. |
| | GetTasks (see page 963) | Get All tasks from the system. |
| | GetTasksAssignedByUser (see page 963) | This is the overview for the GetTasksAssignedByUser method overload. |
| | GetTasksAssignedToUser (see page 964) | This is the overview for the GetTasksAssignedToUser method overload. |
| | GetTasksAssignedToUsergroup (see page 965) | This is the overview for the GetTasksAssignedToUsergroup method overload. |
| | GetTaskState (see page 967) | Get the task state. |
| | SetStateForTask (see page 967) | Set the task state. |
| | UpdateTask (see page 967) | Updates the task |

## Task.AddTask Method

Adds a task in the system.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTask", Description = "Adds a task")]
public bool AddTask(ref TaskData data);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTask", Description = "Adds a task")>
Public Function AddTask(ByRef data As TaskData) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| data | TaskData |

**Returns**

False=if successfully added otherwise True

## AssignTaskToUser Method

This is the overview for the AssignTaskToUser method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.AssignTaskToUser (long, long) (see page 958) | Assign a task to a user. |
| | Task.AssignTaskToUser (long, long, long) (see page 959) | Assign a task to a user by a specified user. |

## Task.AssignTaskToUser Method (long, long)

Assign a task to a user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUser", Description = "Assign a task to a user.")]
public bool AssignTaskToUser(long taskId, long assignToUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUser", Description = "Assign a task to a user.")>
Public Function AssignTaskToUser(taskId As long, assignToUserId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 957) ID |
| assignToUserId | User (see page 1167)'s ID |

**Returns**

False=if successfully assigned to a user otherwise True

### Task.AssignTaskToUser Method (long, long, long)

Assign a task to a user by a specified user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUserByUser", Description = "Assign a task to a user.")]
public bool AssignTaskToUser(long taskId, long assignToUserId, long assignByUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUserByUser", Description = "Assign a task to a user.")>
Public Function AssignTaskToUser(taskId As long, assignToUserId As long, assignByUserId As
long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 957) ID |
| assignToUserId | Assign to User (see page 1167)'s ID |
| assignByUserId | Assign by User (see page 1167)'s ID |

**Returns**

False=if successfully assigned to a user otherwise True

### AssignTaskToUsergroup Method

This is the overview for the AssignTaskToUsergroup method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.AssignTaskToUsergroup (long, long) (see page 959) | Assign a task to a usergroup. |
| | Task.AssignTaskToUsergroup (long, long, long) (see page 960) | Assign a task to a usergroup by a specified user. |

### Task.AssignTaskToUsergroup Method (long, long)

Assign a task to a usergroup.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUsergroup", Description = "Assign a task to a
usergroup.")]
public bool AssignTaskToUsergroup(long taskId, long assignToUsergroup);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUsergroup", Description = "Assign a task to a
usergroup.")>
Public Function AssignTaskToUsergroup(taskId As long, assignToUsergroup As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| taskId | Task (see page 957) ID |
| assignToUsergroup | Assign to Usergroup's ID |

### Returns

False=if successfully assigned to a usergroup otherwise True

## Task.AssignTaskToUsergroup Method (long, long, long)

Assign a task to a usergroup by a specified user.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUsergroupByUser", Description = "Assign a task to a
usergroup by a specified user.")]
public bool AssignTaskToUsergroup(long taskId, long assignToUsergroup, long assignByUserId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUsergroupByUser", Description = "Assign a task to a
usergroup by a specified user.")>
Public Function AssignTaskToUsergroup(taskId As long, assignToUsergroup As long,
assignByUserId As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| taskId | Task (see page 957) ID |
| assignToUsergroup | Assign to Usergroup's ID |
| assignByUserId | Assign by User (see page 1167)'s ID |

### Returns

False=if successfully assigned to a usergroup otherwise True

## DeleteTask Method

This is the overview for the DeleteTask method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | Task.DeleteTask (long) (see page 960) | Mark the task deleted. |
| | Task.DeleteTask (long, bool) (see page 961) | Mark the task deleted or delete the task permanently. |

## Task.DeleteTask Method (long)

Mark the task deleted.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTask", Description = "Mark the task deleted.")]
```

```
public bool DeleteTask(long taskId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTask", Description = "Mark the task deleted.")>
Public Function DeleteTask(taskId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 957) ID |

**Returns**

False=if successfully deleted otherwise True

### Task.DeleteTask Method (long, bool)

Mark the task deleted or delete the task permanently.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaskPermanently", Description = "Mark the task deleted or
delete the task permanently.")]
public bool DeleteTask(long taskId, bool permanently);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaskPermanently", Description = "Mark the task deleted or
delete the task permanently.")>
Public Function DeleteTask(taskId As long, permanently As bool) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 957) ID |
| Permanently | True=Delete the task |

**Returns**

False=if successfully deleted otherwise True

### DeleteTasks Method

This is the overview for the DeleteTasks method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.DeleteTasks (long[]) (see page 961) | Mark tasks deleted. |
| | Task.DeleteTasks (long[], bool) (see page 962) | Mark tasks deleted or delete tasks permanently. |

### Task.DeleteTasks Method (long[])

Mark tasks deleted.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTasks", Description = "Mark tasks deleted.")]
public bool DeleteTasks(long[] taskId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTasks", Description = "Mark tasks deleted.")>
Public Function DeleteTasks(taskId As long[]) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Array of Task (see page 957) ID |

**Returns**

False=if successfully deleted otherwise True

**Remarks**

When an error occurs the method will quite without deleting rest of the tasks.

### Task.DeleteTasks Method (long[], bool)

Mark tasks deleted or delete tasks permanently.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTasksPermanently", Description = "Mark tasks deleted or delete
tasks permanently.")]
public bool DeleteTasks(long[] taskId, bool permanently);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTasksPermanently", Description = "Mark tasks deleted or delete
tasks permanently.")>
Public Function DeleteTasks(taskId As long[], permanently As bool) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Array of Task (see page 957) ID |
| Permanently | True=Delete tasks |

**Returns**

False=if successfully deleted otherwise True

**Remarks**

When an error occurs the method will quite without deleting rest of the tasks.

### Task.GetTask Method

Get the task information.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "GetTask", Description = "Get the task information.")]
public TaskData GetTask(long taskId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTask", Description = "Get the task information.")>
Public Function GetTask(taskId As long) As TaskData
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 957) Id |

**Returns**

Task (see page 957) Data

### Task.GetTasks Method

Get All tasks from the system.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasks", Description = "")]
public TaskData[] GetTasks();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasks", Description = "")>
Public Function GetTasks() As TaskData()
```

**Returns**

Array of task data.

### GetTasksAssignedByUser Method

This is the overview for the GetTasksAssignedByUser method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.GetTasksAssignedByUser (long) (see page 963) | Get all tasks assigned by the user. |
| | Task.GetTasksAssignedByUser (long, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 964) | Get all task assign by the user by a state of tasks specified. |

### Task.GetTasksAssignedByUser Method (long)

Get all tasks assigned by the user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedByUser", Description = "Get all tasks assigned by
the user.")]
```

```
public TaskData[] GetTasksAssignedByUser(long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedByUser", Description = "Get all tasks assigned by
the user.")>
Public Function GetTasksAssignedByUser(userId As long) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | User (see page 1167)'s ID |

**Returns**

Array of tasks.

### Task.GetTasksAssignedByUser Method (long, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all task assign by the user by a state of tasks specified.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedByUserByState", Description = "")]
public TaskData[] GetTasksAssignedByUser(long userId,
Ektron.Cms.Common.EkEnumeration.TaskState taskState);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedByUserByState", Description = "")>
Public Function GetTasksAssignedByUser(userId As long, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | User (see page 1167)'s ID |
| taskState | Task (see page 957) State |

**Returns**

Array of tasks.

### GetTasksAssignedToUser Method

This is the overview for the GetTasksAssignedToUser method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.GetTasksAssignedToUser (long) (see page 965) | Get all tasks assigned to the user. |
| | Task.GetTasksAssignedToUser (long, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 965) | Get all tasks assigned to the user by a state of the task specified. |

### Task.GetTasksAssignedToUser Method (long)

Get all tasks assigned to the user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUser", Description = "Get all tasks assigned to
the user.")]
public TaskData[] GetTasksAssignedToUser(long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUser", Description = "Get all tasks assigned to
the user.")>
Public Function GetTasksAssignedToUser(userId As long) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | User (see page 1167)'s ID |

**Returns**

Array of tasks.

### Task.GetTasksAssignedToUser Method (long, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all tasks assigned to the user by a state of the task specified.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUserByState", Description = "Get all tasks
assigned to the user by a state of the task specified.")]
public TaskData[] GetTasksAssignedToUser(long userId,
Ektron.Cms.Common.EkEnumeration.TaskState taskState);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUserByState", Description = "Get all tasks
assigned to the user by a state of the task specified.")>
Public Function GetTasksAssignedToUser(userId As long, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | User (see page 1167)'s ID |
| taskState | Task (see page 957) State |

**Returns**

Array of tasks.

### GetTasksAssignedToUsergroup Method

This is the overview for the GetTasksAssignedToUsergroup method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Task.GetTasksAssignedToUsergroup (long) (see page 966) | Get all tasks assigned to the usergroup. |
| | Task.GetTasksAssignedToUsergroup (long, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 966) | Get all tasks assigned to the usergroup by a state of tasks specified. |

## Task.GetTasksAssignedToUsergroup Method (long)

Get all tasks assigned to the usergroup.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUsergroup", Description = "Get all tasks assigned
to the usergroup.")]
public TaskData[] GetTasksAssignedToUsergroup(long groupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUsergroup", Description = "Get all tasks assigned
to the usergroup.")>
Public Function GetTasksAssignedToUsergroup(groupId As long) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | Usergroup's ID |

**Returns**

Array of tasks.

## Task.GetTasksAssignedToUsergroup Method (long, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all tasks assigned to the usergroup by a state of tasks specified.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUsergroupByState", Description = "Get all tasks
assigned to the usergroup by a state of tasks specified.")]
public TaskData[] GetTasksAssignedToUsergroup(long groupId,
Ektron.Cms.Common.EkEnumeration.TaskState taskState);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUsergroupByState", Description = "Get all tasks
assigned to the usergroup by a state of tasks specified.")>
Public Function GetTasksAssignedToUsergroup(groupId As long, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData()
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | Usergroup's ID |

| | |
|---|---|
| taskState | Task (see page 957) State |

**Returns**

Array of tasks.

### Task.GetTaskState Method

Get the task state.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskState", Description = "Get the task state.")]
public Ektron.Cms.Common.EkEnumeration.TaskState GetTaskState(long taskId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskState", Description = "Get the task state.")>
Public Function GetTaskState(taskId As long) As Ektron.Cms.Common.EkEnumeration.TaskState
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 957) ID |

**Returns**

If successful returns task state otherwise it will throw an error.

### Task.SetStateForTask Method

Set the task state.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetStateForTask", Description = "Set the task state.")]
public bool SetStateForTask(long taskId, Ektron.Cms.Common.EkEnumeration.TaskState
newTaskState);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SetStateForTask", Description = "Set the task state.")>
Public Function SetStateForTask(taskId As long, newTaskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| taskId | Task (see page 957) ID |
| newTaskState | New Task (see page 957) State |

**Returns**

False=if successfully state of the changed otherwise True

### Task.UpdateTask Method

Updates the task

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateTask", Description = "Updates the task")]
public bool UpdateTask(ref TaskData data);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateTask", Description = "Updates the task")>
Public Function UpdateTask(ByRef data As TaskData) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| data | Task (see page 957) Data |

**Returns**

False=if successfully updated otherwise True

## TaskCategory Class

Summary description for TaskCategory

**Class Hierarchy**

**C#**

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class TaskCategory : EktronBaseService;
```

**Visual Basic**

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class TaskCategory
Inherits EktronBaseService
```

**File**

TaskCategory.cs

### TaskCategory Methods

The methods of the TaskCategory class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddTaskCategory (see page 969) | Adds a task category. |
| | DeleteTaskCategory (see page 969) | Deletes the task category. |
| | GetAllCategories (see page 969) | Gets All Task (see page 957) categories and task types belongs to those categories. |
| | GetTaskCategory (see page 970) | Gets the specified task category and task types belongs to the category. |
| | UpdateTaskCategory (see page 970) | Updates the category title. |

### TaskCategory.AddTaskCategory Method

Adds a task category.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaskCategory", Description = "Adds a task category.")]
public long AddTaskCategory(string title);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaskCategory", Description = "Adds a task category.")>
Public Function AddTaskCategory(title As string) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| title | Category Title |

**Returns**

Category Id

### TaskCategory.DeleteTaskCategory Method

Deletes the task category.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaskCategory", Description = "Deletes the task category.")]
public bool DeleteTaskCategory(long categoryId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaskCategory", Description = "Deletes the task category.")>
Public Function DeleteTaskCategory(categoryId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| categoryId | category id |

**Returns**

False=if successfully deleted otherwise True

### TaskCategory.GetAllCategories Method

Gets All Task (see page 957) categories and task types belongs to those categories.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCategories", Description = "Gets All Task categories and task
types belongs to those categories.")]
public TaskCategoryData[] GetAllCategories();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCategories", Description = "Gets All Task categories and task
types belongs to those categories.")>
Public Function GetAllCategories() As TaskCategoryData()
```

**Returns**

Array of TaskCategoryData

### TaskCategory.GetTaskCategory Method

Gets the specified task category and task types belongs to the category.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskCategory", Description = "Gets the specified task category
and task types belongs to the category.")]
public TaskCategoryData GetTaskCategory(long categoryID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskCategory", Description = "Gets the specified task category
and task types belongs to the category.")>
Public Function GetTaskCategory(categoryID As long) As TaskCategoryData
```

**Parameters**

| Parameters | Description |
|---|---|
| categoryID | category id |

**Returns**

TaskCategoryData

### TaskCategory.UpdateTaskCategory Method

Updates the category title.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateTaskCategory", Description = "Updates the category title.")]
public bool UpdateTaskCategory(long categoryId, string title);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateTaskCategory", Description = "Updates the category title.")>
Public Function UpdateTaskCategory(categoryId As long, title As string) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| 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

## TaskCategoryType Methods

The methods of the TaskCategoryType class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddTaskType (see page 971) | Adds a task type to a category. |
| | DeleteTaskType (see page 972) | Deletes the task type. |
| | GetTaskType (see page 972) | Gets the task type information. |
| | UpdateTaskType (see page 972) | Updates the task type. |

## TaskCategoryType.AddTaskType Method

Adds a task type to a category.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaskType", Description = "Adds a task type to a category.")]
public long AddTaskType(string title, string description, long categoryId, bool active);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaskType", Description = "Adds a task type to a category.")>
Public Function AddTaskType(title As string, description As string, categoryId As long, active
As bool) As long
```

### Parameters

| Parameters | Description |
|---|---|
| title | Title |
| description | Description |
| categoryId | Task (see page 957) Category ID |

| | |
|---|---|
| active | Set as active |

**Returns**

Task (see page 957) Type ID

### TaskCategoryType.DeleteTaskType Method

Deletes the task type.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaskType", Description = "Deletes the task type.")]
public bool DeleteTaskType(long taskTypeId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaskType", Description = "Deletes the task type.")>
Public Function DeleteTaskType(taskTypeId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| taskTypeId | task type id to be deleted. |

**Returns**

False=if successfully deleted otherwise True

### TaskCategoryType.GetTaskType Method

Gets the task type information.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskType", Description = "Gets a task type")]
public TaskTypeData GetTaskType(long taskTypeId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskType", Description = "Gets a task type")>
Public Function GetTaskType(taskTypeId As long) As TaskTypeData
```

**Parameters**

| Parameters | Description |
|---|---|
| taskTypeId | TaskTypeId |

**Returns**

TaskTypeData

### TaskCategoryType.UpdateTaskType Method

Updates the task type.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "UpdateTaskType", Description = "Updates the task type.")]
public bool UpdateTaskType(long taskTypeId, string title, string description, bool active);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateTaskType", Description = "Updates the task type.")>
Public Function UpdateTaskType(taskTypeId As long, title As string, description As string,
active As bool) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| taskTypeId | Task (see page 957) 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 974) | Summary description for ActiveDirectoryUser |
| | Asset (see page 982) | Summary description for Asset |
| | Blog (see page 985) | Summary description for Blog |
| | BusinessRules (see page 999) | Summary description for BusinessRules |
| | CommunityGroup (see page 1001) | Summary description for CommunityGroup |
| | Content (see page 1016) | Summary description for ContentService |
| | ContentRating (see page 1029) | Summary description for ContentRating |
| | CustomFields (see page 1031) | Summary description for CustomFields |
| | Favorites (see page 1039) | Summary description for Favorites |
| | Flag (see page 1044) | The summary description for Taxonomy (see page 1151). |
| | Folder (see page 1048) | Summary description for Folder |
| | Font (see page 1055) | Summary description for Font |
| | Form (see page 1058) | Summary description for Form |
| | Friends (see page 1066) | Summary description for Friends |
| | Library (see page 1078) | Summary description for Library |
| | MessageBoard (see page 1086) | Summary description for MessageBoard |
| | Metadata (see page 1106) | Summary description for Metadata |
| | Permissions (see page 1110) | Summary description for Permissions |
| | SearchManager (see page 1115) | Summary description for SearchManager |
| | Site (see page 1120) | Summary description for Site |
| | SiteMap (see page 1127) | Summary description for SiteMap |
| | Tags (see page 1130) | Summary description for Tags |
| | Taxonomy (see page 1151) | The summary description for Taxonomy. |

| | | |
|---|---|---|
| | ThreadedDiscussion (see page 1157) | Summary description for ThreadedDiscussion |
| | User (see page 1167) | Summary description for User |

# 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

## ActiveDirectoryUser Methods

The methods of the ActiveDirectoryUser class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddADGroupToCMS (see page 975) | This function allows you add Active Directory (see page 189) groups to the CMS. |
| | DeSynchUserGroups (see page 975) | Updates the group with a new group name. |
| | DeSynchUsers (see page 975) | Update the user with new user details. |
| | GetADMapping (see page 976) | Loads the CMS and Active Directory (see page 189) name. |
| | GetADStatus (see page 976) | Used to load required Synch, Desynch and relationship flags of the configured Active Directory (see page 189). |
| | GetAvailableADGroups (see page 977) | Loads all of the Active Directory (see page 189) groups by a given domain and filter. |
| | GetAvailableADUsers (see page 977) | Loads all available Active Directory (see page 189) users. |
| | GetCMSGroupsToSync (see page 978) | Loads all the CMS groups that need to be synchronized. |
| | GetCMSRelationshipsToSync (see page 978) | Loads all the CMS relationships that need to be synchronized with Active Directory (see page 189). |
| | GetCMSUsersToSync (see page 979) | Loads all the CMS users that need to be synchronized with Active Directory (see page 189). |
| | GetDomains (see page 979) | Loads a list of domains from Active Directory (see page 189). |
| | MapCMSUserGroupToAD (see page 980) | Updates the given CMS group corresponding to the Active Directory (see page 189) group. |
| | MapCMSUserToAD (see page 980) | Maps the given user ID to the corresponding Active Directory (see page 189) entry. |
| | UpdateADMapping (see page 981) | Updates the Active directory information into the CMS based on the user ID and user data. |
| | UpdateADUsersGroups (see page 981) | Updates Active Directory (see page 189) users in a CMS group. |

| | | |
|---|---|---|
| | UpdateUsersInfoFromAD (see page 981) | Updates the user details from the Active Directory (see page 189) details. |

### ActiveDirectoryUser.AddADGroupToCMS Method

This function allows you add Active Directory (see page 189) groups to the CMS.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddADGroupToCMS", Description = "")]
public void AddADGroupToCMS(object GroupPaths);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddADGroupToCMS", Description = "")>
Public Function AddADGroupToCMS(GroupPaths As object) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| GroupPaths | A collection of Active Directory (see page 189) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeSynchUserGroups", Description = "")]
public AdDeSyncGroupData[] DeSynchUserGroups(bool Flag);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeSynchUserGroups", Description = "")>
Public Function DeSynchUserGroups(Flag As bool) As AdDeSyncGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "DeSynchUsers", Description = "")]
public AdDeSyncUserData[] DeSynchUsers(bool Flag);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeSynchUsers", Description = "")>
Public Function DeSynchUsers(Flag As bool) As AdDeSyncUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Flag | If true, update the user with new user information, otherwise none. |

**Returns**

Ektron.Cms.AdDeSyncUserData()

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

### ActiveDirectoryUser.GetADMapping Method

Loads the CMS and Active Directory (see page 189) name.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetADMapping", Description = "")]
public AdMappingData[] GetADMapping(long CallerId, string MapType, int Map, int Exclude, int
Editable);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetADMapping", Description = "")>
Public Function GetADMapping(CallerId As long, MapType As string, Map As Integer, Exclude As
Integer, Editable As Integer) As AdMappingData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 1044) (0 – false, 1-true). |
| Editable | Flag (see page 1044)(0-false,1-true). |

**Returns**

Ektron.Cms.AdMappingData()

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

### ActiveDirectoryUser.GetADStatus Method

Used to load required Synch, Desynch and relationship flags of the configured Active Directory (see page 189).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "GetADStatus", Description = "")]
public AdSyncData GetADStatus();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetADStatus", Description = "")>
Public Function GetADStatus() As AdSyncData
```

**Returns**

Ektron.Cms.AdSyncData

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

### ActiveDirectoryUser.GetAvailableADGroups Method

Loads all of the Active Directory (see page 189) groups by a given domain and filter.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAvailableADGroups", Description = "")]
public GroupData[] GetAvailableADGroups(string Filter, string Domain);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAvailableADGroups", Description = "")>
Public Function GetAvailableADGroups(Filter As string, Domain As string) As GroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 189) users.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAvailableADUsers", Description = "")]
public UserData[] GetAvailableADUsers(object Attributes, object Filter, string OrderBy, string
Domain);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAvailableADUsers", Description = "")>
Public Function GetAvailableADUsers(Attributes As object, Filter As object, OrderBy As string,
Domain As string) As UserData()
```

### Parameters

| Parameters | Description |
|---|---|
| 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 (see page 1419)()

### 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCMSGroupsToSync", Description = "")]
public GroupData[] GetCMSGroupsToSync(int MaxReturn);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCMSGroupsToSync", Description = "")>
Public Function GetCMSGroupsToSync(MaxReturn As Integer) As GroupData()
```

### Parameters

| Parameters | Description |
|---|---|
| MaxReturn | The maximum number of groups that should be displayed. |

### Returns

Ektron.Cms.GroupData()

### Remarks

A user with administrator privileges must be logged in to perform this operation.

## ActiveDirectoryUser.GetCMSRelationshipsToSync Method

Loads all the CMS relationships that need to be synchronized with Active Directory (see page 189).

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCMSRelationshipsToSync", Description = "")]
public UserGroupData[] GetCMSRelationshipsToSync(int MaxReturn);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCMSRelationshipsToSync", Description = "")>
Public Function GetCMSRelationshipsToSync(MaxReturn As Integer) As UserGroupData()
```

### Parameters

| Parameters | Description |
|---|---|
| MaxReturn | The maximum number of results that should be displayed. |

### Returns

Ektron.Cms.UserGroupData()

### Remarks

A user with administrator privileges must be logged in to perform this operation.

## ActiveDirectoryUser.GetCMSUsersToSync Method

Loads all the CMS users that need to be synchronized with Active Directory (see page 189).

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCMSUsersToSync", Description = "")]
public UserData[] GetCMSUsersToSync(int MaxReturn);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCMSUsersToSync", Description = "")>
Public Function GetCMSUsersToSync(MaxReturn As Integer) As UserData()
```

### Parameters

| Parameters | Description |
|---|---|
| MaxReturn | The maximum number of users that should be displayed. |

### Returns

Ektron.Cms.UserData (see page 1419)()

### Remarks

A user with administrator privileges must be logged in to perform this operation.

## ActiveDirectoryUser.GetDomains Method

Loads a list of domains from Active Directory (see page 189).

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDomains", Description = "")]
public DomainData[] GetDomains(int ShowAll, int CheckIntegration);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDomains", Description = "")>
Public Function GetDomains(ShowAll As Integer, CheckIntegration As Integer) As DomainData()
```

### Parameters

| Parameters | Description |
|---|---|
| ShowAll | flag(1 = show all). |
| CheckIntegration | Check whether CMS is configured to integrate Active Directory (see page 189) (1 = check otherwise resumed). |

### Returns

Ektron.Cms.DomainData()

## ActiveDirectoryUser.MapCMSUserGroupToAD Method

Updates the given CMS group corresponding to the Active Directory (see page 189) group.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MapCMSUserGroupToAD", Description = "")]
public void MapCMSUserGroupToAD(long GroupId, string GroupName, string Domain);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MapCMSUserGroupToAD", Description = "")>
Public Function MapCMSUserGroupToAD(GroupId As long, GroupName As string, Domain As string) As
void
```

### Parameters

| Parameters | Description |
|---|---|
| GroupId | CMS group ID. |
| GroupName | CMS group name. |
| Domain | The Active Directory (see page 189) domain. |

### Remarks

A user with administrator privileges must be logged in to perform this operation.

## ActiveDirectoryUser.MapCMSUserToAD Method

Maps the given user ID to the corresponding Active Directory (see page 189) entry.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MapCMSUserToAD", Description = "")]
public void MapCMSUserToAD(long UserId, string UserName, string Domain, int PullInfo);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MapCMSUserToAD", Description = "")>
Public Function MapCMSUserToAD(UserId As long, UserName As string, Domain As string, PullInfo
As Integer) As void
```

### Parameters

| Parameters | Description |
|---|---|
| UserId | The user's ID. |
| UserName | The user's name. |
| Domain | The user's domain. |
| PullInfo | Flag (see page 1044) to load Active Directory (see page 189) information. (1 = Load Active Directory (see page 189) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateADMapping", Description = "")]
public void UpdateADMapping(long UserId, object pagedata);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateADMapping", Description = "")>
Public Function UpdateADMapping(UserId As long, pagedata As object) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 189) users in a CMS group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateADUsersGroups", Description = "")]
public void UpdateADUsersGroups(string UserName, string Domain);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateADUsersGroups", Description = "")>
Public Function UpdateADUsersGroups(UserName As string, Domain As string) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 189) details.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateUsersInfoFromAD", Description = "")]
public void UpdateUsersInfoFromAD(string UserName, string Domain);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateUsersInfoFromAD", Description = "")>
Public Function UpdateUsersInfoFromAD(UserName As string, Domain As string) As void
```

### Parameters

| Parameters | Description |
|---|---|
| UserName | The user's username. |
| Domain | The user's Domain. |

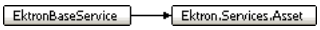
### Remarks

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

## Asset Methods

The methods of the Asset class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | GetAssetMgtConfigInfo (see page 983) | Loads all of the registered plug-in configuration details. |
| | GetAssetMgtInfo (see page 983) | Loads all of the registered plug-in type details. |
| | GetAssetSupertypes (see page 983) | Loads all of the asset supertypes supported by documents add-in. |
| | GetEditSnippet (see page 984) | Loads the edit snippet for the given document. Used with the Document Management functionality. |
| | GetMyAssetMgtID (see page 984) | Retrieves the Document Management functionality client ID. |
| | MakeCopyDMSAsset (see page 984) | Create a copy of DMS Asset (see page 982). |
| | SetAssetMgtConfigInfo (see page 985) | Updates the Document Management configuration information. |

### Asset.GetAssetMgtConfigInfo Method

Loads all of the registered plug-in configuration details.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAssetMgtConfigInfo", Description = "")]
public AssetConfigInfo[] GetAssetMgtConfigInfo();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAssetMgtConfigInfo", Description = "")>
Public Function GetAssetMgtConfigInfo() As AssetConfigInfo()
```

**Returns**

Ektron.Cms.AssetConfigInfo()

### Asset.GetAssetMgtInfo Method

Loads all of the registered plug-in type details.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAssetMgtInfo", Description = "")]
public AssetInfoData[] GetAssetMgtInfo();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAssetMgtInfo", Description = "")>
Public Function GetAssetMgtInfo() As AssetInfoData()
```

**Returns**

Ektron.Cms.AssetInfoData()

### Asset.GetAssetSupertypes Method

Loads all of the asset supertypes supported by documents add-in.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAssetSupertypes", Description = "")]
public AssetInfoData[] GetAssetSupertypes();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAssetSupertypes", Description = "")>
Public Function GetAssetSupertypes() As AssetInfoData()
```

**Returns**

Ektron.Cms.AssetInfoData()

### Asset.GetEditSnippet Method

Loads the edit snippet for the given document. Used with the Document Management functionality.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetEditSnippet", Description = "")]
public string GetEditSnippet(string AssetId, int ContentType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetEditSnippet", Description = "")>
Public Function GetEditSnippet(AssetId As string, ContentType As Integer) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| AssetId | The document's ID (Required) |
| ContentType | The type of the content, such as office documents or managed files or * (all). |

**Returns**

String

**Remarks**

The Document Management functionality with a valid license is required to perform this operation.

### Asset.GetMyAssetMgtID Method

Retrieves the Document Management functionality client ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMyAssetMgtID", Description = "")]
public string GetMyAssetMgtID();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMyAssetMgtID", Description = "")>
Public Function GetMyAssetMgtID() As string
```

**Returns**

String

### Asset.MakeCopyDMSAsset Method

Create a copy of DMS Asset (see page 982).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MakeCopyDMSAsset", Description = "")]
public bool MakeCopyDMSAsset(string AssetID, string AssetVersion);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "MakeCopyDMSAsset", Description = "")>
Public Function MakeCopyDMSAsset(AssetID As string, AssetVersion As string) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| strAssetID | ID associated with the asset |
| strAssetVersion | Version of the asset |

**Returns**

boolean

**Remarks**

User (see page 1167) login with create content privileges are required to perform this operation.

### Asset.SetAssetMgtConfigInfo Method

Updates the Document Management configuration information.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetAssetMgtConfigInfo", Description = "")]
public bool SetAssetMgtConfigInfo(AssetConfigInfo[] AssetConfig);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SetAssetMgtConfigInfo", Description = "")>
Public Function SetAssetMgtConfigInfo(AssetConfig As AssetConfigInfo[]) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| asset_config | The Document Management configuration data |

**Returns**

Boolean

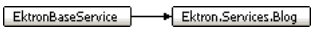
**Remarks**

User (see page 1167) 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

## Blog Methods

The methods of the Blog class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddBlog (see page 986) | Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog. |
| | AddCommentForPost (see page 987) | 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 987) | Adds a post to a given blog. Accepts a ContentData (see page 1378) class as the post content. Return new post ID. |
| | DeleteBlog (see page 988) | Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0. |
| | DeleteBlogPostComment (see page 988) | Deletes the blog post comment by it's id. |
| | DeletePost (see page 988) | 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 989) | Returns a blank BlogPostData class. Used Primarily in the Workarea. |
| | GetBlog (see page 989) | This is the overview for the GetBlog method overload. |
| | GetBlogData (see page 991) | This is the overview for the GetBlogData method overload. |
| | GetBlogDataForPost (see page 995) | 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 995) | 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 995) | This is the overview for the GetBlogRoll method overload. |
| | GetPost (see page 997) | Gets a blog post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData (see page 1378) Class. |
| | GetPostbyID (see page 997) | Gets a blog post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class. |
| | GetUserBlog (see page 998) | Gets a blog based on a user's ID. |
| | UpdateBlogProperties (see page 998) | This updates the blog data. |

### Blog.AddBlog Method

Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddBlog", Description = "")]
public long AddBlog(long ParentFolderID, string BlogName, string BlogTitle, string
BlogDescription, Ektron.Cms.Common.EkEnumeration.BlogVisibility Visibility, bool
EnableComments, bool ModerateComments, bool RequireAuthentication, string[] Categories,
BlogRoll RollLinks);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddBlog", Description = "")>
Public Function AddBlog(ParentFolderID As long, BlogName As string, BlogTitle As string,
BlogDescription As string, Visibility As Ektron.Cms.Common.EkEnumeration.BlogVisibility,
EnableComments As bool, ModerateComments As bool, RequireAuthentication As bool, Categories As
string[], RollLinks As BlogRoll) As long
```

### Parameters

| Parameters | Description |
|---|---|
| ParentFolderID | |
| BlogName | Name of the Blog (see page 985). |
| BlogTitle | Title of the Blog (see page 985). |
| BlogDescription | Description of the Blog (see page 985). |
| 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 (see page 985)'s roll links. |

### Returns

The numeric value of the blog ID that is greater than 0.

## Blog.AddCommentForPost Method

Adds a post comment to a blog post. Accepts a BlogComment class as the input. Will return the result of that add action. False indicates the action was successful.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddCommentForPost", Description = "")]
public bool AddCommentForPost(BlogComment bcComment);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddCommentForPost", Description = "")>
Public Function AddCommentForPost(bcComment As BlogComment) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| 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 (see page 1378) class as the post content. Return new post ID.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddPost", Description = "")]
public long AddPost(long blogID, ContentData PostContent, string[] Categories, bool Pingback,
string Tags, string TrackbackURL);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddPost", Description = "")>
Public Function AddPost(blogID As long, PostContent As ContentData, Categories As string[],
Pingback As bool, Tags As string, TrackbackURL As string) As long
```

### Parameters

| Parameters | Description |
|---|---|
| blogID | Numeric ID of the blog to add the post to. |
| PostContent | ContentData (see page 1378) 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

Numeric ID of the post created.

## Blog.DeleteBlog Method

Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteBlog", Description = "")]
public void DeleteBlog(long BlogID);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteBlog", Description = "")>
Public Function DeleteBlog(BlogID As long) As void
```

### Parameters

| Parameters | Description |
|---|---|
| BlogID | The numeric ID of the blog. This corresponds to the folder ID of the blog. |

## Blog.DeleteBlogPostComment Method

Deletes the blog post comment by it's id.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteBlogPostComment", Description = "")]
public void DeleteBlogPostComment(Int64 postComment_id);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteBlogPostComment", Description = "")>
Public Function DeleteBlogPostComment(postComment_id As Int64) As void
```

### Parameters

| Parameters | Description |
|---|---|
| postComment_id | post comment id |

## Blog.DeletePost Method

Deletes a post by the post's ID This method will delete all comments associated with the post as well. Returns boolean value.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeletePost", Description = "")]
public bool DeletePost(long post_id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeletePost", Description = "")>
Public Function DeletePost(post_id As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| post_id | The numeric value of the post ID you wish to delete. |

**Returns**

Boolean

### Blog.GetBlankBlogPostData Method

Returns a blank BlogPostData class. Used Primarily in the Workarea.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlankBlogPostData", Description = "")]
public BlogPostData GetBlankBlogPostData();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlankBlogPostData", Description = "")>
Public Function GetBlankBlogPostData() As BlogPostData
```

**Returns**

A blank, but not null, BlogPostData class.

### GetBlog Method

This is the overview for the GetBlog method overload.

**Description**

Returns a BlogData class for a given blog id, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This method is overloaded.

- GetBlog (Long, String, Int, Int) - returns a BlogData class for a given blog by providing the blog's ID, the date to start getting posts, the maximum number of entries to return and the ID of the Language.
- GetBlog (String, String, Int, Int) - returns a BlogData class for a given blog by providing the blog's folder path in CMS400.NET, the date to start getting posts, the maximum number of entries to return and the ID of the Language.

**Overload List**

| | Name | Description |
|---|---|---|
| | Blog.GetBlog (long, string, int, int) (see page 990) | 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.<br>This function is overloaded. See Also:<br>GetBlog (int, string, int, int) (see page 991) |
| | Blog.GetBlog (string, string, int, int) (see page 991) | Returns a BlogData class for a given path, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.<br>This function is overloaded. See Also:<br>GetBlog (string, string, int, int) |

## Blog.GetBlog Method (long, string, int, int)

Returns a BlogData class for a given blog ID, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

This function is overloaded. See Also:

GetBlog (int, string, int, int) (see page 991)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlog by ID, StartDate, MaxEntries and LanguageID", Description =
"")]
public BlogData GetBlog(long ID, string StartDate, int MaxEntries, int LanguageID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlog by ID, StartDate, MaxEntries and LanguageID", Description =
"")>
Public Function GetBlog(ID As long, StartDate As string, MaxEntries As Integer, LanguageID As
Integer) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| ID | The ID of the blog. This is a numeric 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.<br>If set to 0, the `postsvisible` property of the blog is used.<br>If set to -1, the content is not returned, only the post info for the content. |
| LanguageID | The language ID for the content to get. |

**Returns**

For a given blog ID, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

**Remarks**

Developers are advised to check for null prior to using.

## Blog.GetBlog Method (string, string, int, int)

Returns a BlogData class for a given path, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

This function is overloaded. See Also:

GetBlog (string, string, int, int)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlog By Path, StartDate, MaxEntries and LanguageID", Description
= "")]
public BlogData GetBlog(string path, string StartDate, int MaxEntries, int LanguageID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlog By Path, StartDate, MaxEntries and LanguageID", Description
= "")>
Public Function GetBlog(path As string, StartDate As string, MaxEntries As Integer, LanguageID
As Integer) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| 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.<br>If set to 0, the `postsvisible` property of the blog is used.<br>If set to -1, the content is not returned, only the post info for the content. |
| LanguageID | The language ID for the content to get. |

**Returns**

For a given path, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

**Remarks**

Developers are advised to check for null prior to using.

## GetBlogData Method

This is the overview for the GetBlogData method overload.

**Description**

Overloaded Method to obtain a BlogData Class.

- GetBlogData (FolderData) - returns the corresponding blog (if applicable) for a complete folderData object.
- GetBlogData (Long) - returns either a populated BlogData class or null (nothing) by providing the ID of the blog.
- GetBlogData (Long, DateTime)- returns either a populated BlogData class or null (nothing) by providing the ID of the blog and date to start returning data.
- GetBlogData (Long, DateTime, Int, Int)- returns either a populated BlogData class or null (nothing) by providing the integer ID of the blog, the date to start returning data, the number of posts to return and the language ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Blog.GetBlogData (FolderData) (see page 992) | 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.**<br>This function is overloaded. See Also:<br>GetBlogData (long) (see page 993)<br>GetBlogData (long, DateTime) (see page 993)<br>GetBlogData (long, DateTime, int, int) (see page 994) |
| | Blog.GetBlogData (long) (see page 993) | Overloaded Method to obtain a BlogData Class. Accepts the Numeric ID of the blog. Returns either a populated BlogData class or null (nothing).<br>This function is overloaded. See Also:<br>GetBlogData (FolderData) (see page 992)<br>GetBlogData (long, DateTime) (see page 993)<br>GetBlogData (long, DateTime, int, int,) (see page 994) |
| | Blog.GetBlogData (long, DateTime) (see page 993) | Overloaded Method to obtain a BlogData Class. Accepts the numeric ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing).<br>This function is overloaded. See Also:<br>GetBlogData (FolderData) (see page 992)<br>GetBlogData (Long) (see page 993)<br>GetBlogData (Long, DateTime, int, int,) (see page 994) |
| | Blog.GetBlogData (long, DateTime, int, int) (see page 994) | Overloaded Method to obtain a BlogData Class. Accepts the numeric ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing).<br>This function is overloaded. See Also:<br>GetBlogData (FolderData) (see page 992)<br>GetBlogData (Long) (see page 993)<br>GetBlogData (Long, DateTime) (see page 993) |

## 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:

GetBlogData (long) (see page 993)

GetBlogData (long, DateTime) (see page 993)

GetBlogData (long, DateTime, int, int) (see page 994)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData By FolderData", Description = "")]
public BlogData GetBlogData(FolderData folder_data);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "GetBlogData By FolderData", Description = "")>
Public Function GetBlogData(folder_data As FolderData) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| folder_data | A populated FolderData class. (see FolderData) |

**Returns**

A populated BlogData class or nothing

**Remarks**

## Blog.GetBlogData Method (long)

Overloaded Method to obtain a BlogData Class. Accepts the Numeric ID of the blog. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

GetBlogData (FolderData) (see page 992)

GetBlogData (long, DateTime) (see page 993)

GetBlogData (long, DateTime, int, int,) (see page 994)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData By ID", Description = "")]
public BlogData GetBlogData(long BlogID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData By ID", Description = "")>
Public Function GetBlogData(BlogID As long) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| BlogID | Numeric ID of the blog. This corresponds to the ID of the blog folder. |

**Returns**

Either a populated BlogData class or null (nothing).

**Remarks**

## Blog.GetBlogData Method (long, DateTime)

Overloaded Method to obtain a BlogData Class. Accepts the numeric ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

GetBlogData (FolderData) (see page 992)

GetBlogData (Long) (see page 993)

GetBlogData (Long, DateTime, int, int,) (see page 994)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData By ID and StartDate", Description = "")]
public BlogData GetBlogData(long BlogID, DateTime StartDate);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData By ID and StartDate", Description = "")>
Public Function GetBlogData(BlogID As long, StartDate As DateTime) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| BlogID | Numeric 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 (long, DateTime, int, int)

Overloaded Method to obtain a BlogData Class. Accepts the numeric ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

GetBlogData (FolderData) (see page 992)

GetBlogData (Long) (see page 993)

GetBlogData (Long, DateTime) (see page 993)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData by ID, StartDate, PostsVisible and LanguageID",
Description = "")]
public BlogData GetBlogData(long BlogID, DateTime StartDate, int Postsvisible, int LanguageID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData by ID, StartDate, PostsVisible and LanguageID",
Description = "")>
Public Function GetBlogData(BlogID As long, StartDate As DateTime, Postsvisible As Integer,
LanguageID As Integer) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| BlogID | Numeric 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 | The number of posts to return. Leave as 0 to use the blog setting, or -1 to return none. |
| LanguageID | Numeric ID of the language. |

**Returns**

Either a populated BlogData class or null (nothing).

**Remarks**

## Blog.GetBlogDataForPost Method

Returns the blog information for a blog post. This contains information about categories, whether comments are enabled, etc. Accepts the post ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogDataForPost", Description = "")]
public BlogData GetBlogDataForPost(long post_id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogDataForPost", Description = "")>
Public Function GetBlogDataForPost(post_id As long) As BlogData
```

**Parameters**

| Parameters | Description |
|---|---|
| post_id | Numeric value of the post ID. |

**Returns**

BlogData class

## Blog.GetBlogPostData Method

Gets the Post data for a specific blog post. Accepts the ID of the post. Will return tags, trackback, pingback, and categories for the post.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogPostData", Description = "")]
public BlogPostData GetBlogPostData(long id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogPostData", Description = "")>
Public Function GetBlogPostData(id As long) As BlogPostData
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Numeric value of the post ID. |

**Returns**

BlogPostData Class.

## GetBlogRoll Method

This is the overview for the GetBlogRoll method overload.

**Description**

Returns a populated blog roll class. This blog roll is the list of links for the blog. See the BlogRoll class for the properties and usage of the blog roll.

This is an overloaded method.

- GetBlogRoll (Long) - returns a populated blog roll class by providing the ID of the blog.
- GetBlogRoll (String) - returns a populated blog roll class by providing the CMS400.NET folder path of the blog.

**Overload List**

| | Name | Description |
|---|---|---|
| | Blog.GetBlogRoll (long) (see page 996) | Returns a populated blog roll class. Accepts a blog's numeric ID value. See the BlogRoll (see page 52) class for more information.<br>This function is overloaded. See Also:<br>GetBlogRoll (string) (see page 996) |
| | Blog.GetBlogRoll (string) (see page 996) | 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 52) class for the properties and usage of the blog roll.<br>This is an overloaded method. See Also:<br>GetBlogRoll (int). |

## Blog.GetBlogRoll Method (long)

Returns a populated blog roll class. Accepts a blog's numeric ID value. See the BlogRoll (see page 52) class for more information.

This function is overloaded. See Also:

GetBlogRoll (string) (see page 996)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogRoll", Description = "")]
public BlogRoll GetBlogRoll(long blog_id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogRoll", Description = "")>
Public Function GetBlogRoll(blog_id As long) As BlogRoll
```

**Parameters**

| Parameters | Description |
|---|---|
| blog_id | Numeric ID of the blog. This corresponds to the folder ID of the blog. |

**Returns**

A populated BlogRoll (see page 52) 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 52) class for the properties and usage of the blog roll.

This is an overloaded method. See Also:

GetBlogRoll (int).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogRoll By Path", Description = "")]
public BlogRoll GetBlogRoll(string Path);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogRoll By Path", Description = "")>
Public Function GetBlogRoll(Path As string) As BlogRoll
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 52) 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 (see page 1378) Class.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPost", Description = "")]
public ContentData GetPost(long post_id, BlogPostData PostInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPost", Description = "")>
Public Function GetPost(post_id As long, PostInfo As BlogPostData) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| post_id | The numeric ID for the post. |
| PostInfo | BlogPostData class representing the blog specific post info. This is also in the metadata. |

**Returns**

ContentData (see page 1378) class with a ByRef BlogPostData

### Blog.GetPostbyID Method

Gets a blog post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class.

**C#**

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPostbyID", Description = "")]
public BlogPostData GetPostbyID(long post_id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPostbyID", Description = "")>
Public Function GetPostbyID(post_id As long) As BlogPostData
```

**Parameters**

| Parameters | Description |
|---|---|
| post_id | The numeric ID for the post. |

**Returns**

BlogPostData class

### Blog.GetUserBlog Method

Gets a blog based on a user's ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserBlog", Description = "retrieves the user's blog folder")]
public long GetUserBlog(long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserBlog", Description = "retrieves the user's blog folder")>
Public Function GetUserBlog(userId As long) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | A user's ID. |

### Blog.UpdateBlogProperties Method

This updates the blog data.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateBlogProperties", Description = "updates blog data.")]
public bool UpdateBlogProperties(BlogData blog);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateBlogProperties", Description = "updates blog data.")>
Public Function UpdateBlogProperties(blog As BlogData) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| blog | blog data |

# 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

## BusinessRules Methods

The methods of the BusinessRules class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | GetBusinessRules (see page 999) | This is the overview for the GetBusinessRules method overload. |

## GetBusinessRules Method

This is the overview for the GetBusinessRules method overload.

### Description

Method to get the results from a business rule. This method is overloaded.

- GetBusinessRules (string, string, string, string, string, string, string, int) (see page 1000)
- GetBusinessRules (string, string, string, string, string, string, string, string) (see page 1000)

### Overload List

| | Name | Description |
|---|---|---|
| | BusinessRules.GetBusinessRules (string, string, string, string, string, string, string, int) (see page 1000) | 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 1000) | 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBusinessRules By Int RuleID", Description = "")]
public string GetBusinessRules(string httpHost, string urlRest, string cookies, string
formValues, string queryValues, string serverValues, string paramValues, int RuleID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBusinessRules By Int RuleID", Description = "")>
Public Function GetBusinessRules(httpHost As string, urlRest As string, cookies As string,
formValues As string, queryValues As string, serverValues As string, paramValues As string,
RuleID As Integer) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| httpHost | The HTTP host the request is made on. |
| urlRest | Not used. |
| cookies | Cookie Value Collection (see page 88) |
| formValues | Form (see page 1058) Value Collection (see page 88) |
| queryValues | Querystring Value Collection (see page 88) |
| serverValues | Server Variables Collection (see page 88) |
| paramValues | Custom Value Collection (see page 88) |
| RuleID | Numeric value representing the ID of the Ruleset. |

**Returns**

A string representing the output of the evaluated ruleset.

### BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, string)

Method to get the results from a business rule by identifier. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBusinessRules by String RuleID", Description = "")]
public string GetBusinessRules(string httpHost, string urlRest, string cookies, string
formValues, string queryValues, string serverValues, string paramValues, string RuleID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBusinessRules by String RuleID", Description = "")>
Public Function GetBusinessRules(httpHost As string, urlRest As string, cookies As string,
formValues As string, queryValues As string, serverValues As string, paramValues As string,
RuleID As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| httpHost | The HTTP host the request is made on. |

| | |
|---|---|
| urlRest | Not used. |
| cookies | Cookie Value Collection (see page 88) |
| formValues | Form (see page 1058) Value Collection (see page 88) |
| queryValues | Querystring Value Collection (see page 88) |
| serverValues | Server Variables Collection (see page 88) |
| paramValues | Custom Value Collection (see page 88) |
| 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.

# CommunityGroup Class

Summary description for CommunityGroup

### Class Hierarchy

### C#

```
public class CommunityGroup : EktronBaseService;
```

### Visual Basic

```
Public Class CommunityGroup
Inherits EktronBaseService
```

### File

CommunityGroup.cs

## CommunityGroup Methods

The methods of the CommunityGroup class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddCommunityGroup (see page 1002) | Saves a new community group. |
| | AddUserToCommunityGroup (see page 1002) | This is the overview for the AddUserToCommunityGroup method overload. |
| | ApprovePendingGroupUser (see page 1004) | Approves the user's pending group request. |
| | CancelJoinRequestForCommunityGroup (see page 1004) | Cancels a community group join request sent by the user. |
| | DeleteCommunityGroupByID (see page 1004) | Deletes an existing community group. |
| | DeletePendingGroupUser (see page 1005) | Delete or reject a users pending group request. |
| | GetAllCommunityGroups (see page 1005) | Returns an array of Community (see page 718) Groups based upon supplied criteria. |
| | GetAllCommunityGroupsInDirectory (see page 1006) | Retrieves a list of community groups in a Directory (see page 189). |
| | GetAllUnassignedCommunityGroupUsers (see page 1006) | Returns a list of users who are not currently in a community group. |
| | GetCommunityGroupBlogIds (see page 1006) | Gets a list of blog IDs that belong to the community group. |
| | GetCommunityGroupByID (see page 1007) | Retrieves a CommunityGroupData object by ID. |

| | | |
|---|---|---|
| | GetCommunityGroupDiscussionBoardIds (see page 1007) | Gets a list of Discussion Board IDs that belong to the community group. |
| | GetCommunityGroups (see page 1008) | Gets a list of community groups a user belongs to. |
| | GetCommunityGroupUserIDs (see page 1008) | Gets a list of user IDs that belong to the community group. |
| | GetCommunityGroupUsers (see page 1009) | This is the overview for the GetCommunityGroupUsers method overload. |
| | GetFolderIdForCommunityGroup (see page 1010) | Gets the folder ID used by the Community (see page 718) Group in the CMS content tree. |
| | GetGroupMemberStatus (see page 1010) | Gets the Member status of a user for a particular group. |
| | GetInvitedCommunityGroups (see page 1011) | Returns a list of Community (see page 718) Groups that a user has been invited to join. |
| | GetMostPopularCommunityGroups (see page 1011) | This is the overview for the GetMostPopularCommunityGroups method overload. |
| | GetPendingCommunityGroups (see page 1013) | Gets a list of community groups for which the user's membership is pending the Group administrator's approval. |
| | GetPendingCommunityGroupUsers (see page 1013) | Returns a list of users whose community group membership is pending approval. |
| | InviteUsers (see page 1014) | Invites the users to the community group. |
| | IsCommunityGroupAdmin (see page 1014) | Returns true if the user is an administrator of the group. |
| | IsUserInCommunityGroup (see page 1014) | Returns true if the user is a member of the community group. |
| | RemoveUserFromCommunityGroup (see page 1015) | Removes a user from a community group. |
| | UpdateCommunityGroup (see page 1015) | Updates an existing community group. |

## CommunityGroup.AddCommunityGroup Method

Saves a new community group.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddCommunityGroup", Description = "Saves a new community group.")]
public long AddCommunityGroup(CommunityGroupData groupData);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddCommunityGroup", Description = "Saves a new community group.")>
Public Function AddCommunityGroup(groupData As CommunityGroupData) As long
```

### Parameters

| Parameters | Description |
|---|---|
| groupData | CommunityGroupData class containing information about new community group. |

### Returns

ID of the new community group.

## AddUserToCommunityGroup Method

This is the overview for the AddUserToCommunityGroup method overload.

### Description

This method is overloaded.

- AddUserToCommunityGroup (Long, Long) - adds a user to a community group by providing the group's ID and the user's ID.

- AddUserToCommunityGroup (Long, Long, Bool) - adds a user to a community group by providing the group's ID and the user's ID. This method allows you to override the community group's enrollment moderation.

**Overload List**

| | Name | Description |
|---|---|---|
| | CommunityGroup.AddUserToCommunityGroup (long, long) (see page 1003) | Adds a user to a community group by providing the group's ID and the user's ID. |
| | CommunityGroup.AddUserToCommunityGroup (long, long, bool) (see page 1003) | Adds a user to a community group. This method allows you to bypass community group enrollment moderation. |

## CommunityGroup.AddUserToCommunityGroup Method (long, long)

Adds a user to a community group by providing the group's ID and the user's ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUserToCommunityGroup", Description = "Adds a user to a community
group.")]
public void AddUserToCommunityGroup(long groupId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUserToCommunityGroup", Description = "Adds a user to a community
group.")>
Public Function AddUserToCommunityGroup(groupId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of the community group. |
| userId | ID of the user to be added. |

## CommunityGroup.AddUserToCommunityGroup Method (long, long, bool)

Adds a user to a community group. This method allows you to bypass community group enrollment moderation.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUserToCommunityGroup with override", Description = "Adds a user
to a community group.")]
public void AddUserToCommunityGroup(long groupId, long userId, bool overridePending);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUserToCommunityGroup with override", Description = "Adds a user
to a community group.")>
Public Function AddUserToCommunityGroup(groupId As long, userId As long, overridePending As
bool) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of the community group. |
| userId | ID of the user to be added. |

| | |
|---|---|
| overridePending | Set to true if you wish to bypass community group enrollment moderation. |

### CommunityGroup.ApprovePendingGroupUser Method

Approves the user's pending group request.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ApprovePendingGroupUser", Description = "Approves the user's pending
group request.")]
public void ApprovePendingGroupUser(long userId, long groupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ApprovePendingGroupUser", Description = "Approves the user's pending
group request.")>
Public Function ApprovePendingGroupUser(userId As long, groupId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user to approve. |
| groupId | ID of the community group to approve user for. |

### CommunityGroup.CancelJoinRequestForCommunityGroup Method

Cancels a community group join request sent by the user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CancelJoinRequestForCommunityGroup", Description = "Cancels a group
join request sent by the user.")]
public void CancelJoinRequestForCommunityGroup(long groupId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CancelJoinRequestForCommunityGroup", Description = "Cancels a group
join request sent by the user.")>
Public Function CancelJoinRequestForCommunityGroup(groupId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of the community group the join request was sent to. |
| userId | ID of the user whom initially sent join request. |

### CommunityGroup.DeleteCommunityGroupByID Method

Deletes an existing community group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteCommunityGroupByID", Description = "Deletes an existing
community group.")]
public void DeleteCommunityGroupByID(long groupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteCommunityGroupByID", Description = "Deletes an existing
community group.")>
Public Function DeleteCommunityGroupByID(groupId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of the group to delete. |

## CommunityGroup.DeletePendingGroupUser Method

Delete or reject a users pending group request.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeletePendingGroupUser", Description = "Delete or reject a users
pending group request.")]
public void DeletePendingGroupUser(long userId, long groupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeletePendingGroupUser", Description = "Delete or reject a users
pending group request.")>
Public Function DeletePendingGroupUser(userId As long, groupId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user for which to delete a request. |
| groupId | ID of the community group for which to delete a request. |

## CommunityGroup.GetAllCommunityGroups Method

Returns an array of Community (see page 718) Groups based upon supplied criteria.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCommunityGroups", Description = "Returns an array of Community
Groups based upon supplied criteria.")]
public CommunityGroupData[] GetAllCommunityGroups(CommunityGroupRequest request);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCommunityGroups", Description = "Returns an array of Community
Groups based upon supplied criteria.")>
Public Function GetAllCommunityGroups(request As CommunityGroupRequest) As CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| request | Request criteria defining the groups to retrieve. |

**Returns**

An array of CommunityGroupData objects.

### CommunityGroup.GetAllCommunityGroupsInDirectory Method

Retrieves a list of community groups in a Directory (see page 189).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCommunityGroupsInDirectory", Description = "Retrieves a list
of community groups in a Directory.")]
public CommunityGroupData[] GetAllCommunityGroupsInDirectory(ref DirectoryRequest request);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCommunityGroupsInDirectory", Description = "Retrieves a list
of community groups in a Directory.")>
Public Function GetAllCommunityGroupsInDirectory(ByRef request As DirectoryRequest) As
CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| request | Request criteria for group retrieval. |

**Returns**

An array of CommunityGroupData objects.

### CommunityGroup.GetAllUnassignedCommunityGroupUsers Method

Returns a list of users who are not currently in a community group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUnassignedCommunityGroupUsers", Description = "Returns a list
of users who are not currently in a community group.")]
public DirectoryUserData[] GetAllUnassignedCommunityGroupUsers(ref CommunityGroupObjectRequest
request);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUnassignedCommunityGroupUsers", Description = "Returns a list
of users who are not currently in a community group.")>
Public Function GetAllUnassignedCommunityGroupUsers(ByRef request As
CommunityGroupObjectRequest) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| request | Request criteria for user list. |

**Returns**

An array of DirectoryUserData objects.

### CommunityGroup.GetCommunityGroupBlogIds Method

Gets a list of blog IDs that belong to the community group.

**C#**

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupBlogIds", Description = "Gets a list of Blog IDs
that belong to the community group.")]
public long[] GetCommunityGroupBlogIds(long communityGroupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupBlogIds", Description = "Gets a list of Blog IDs
that belong to the community group.")>
Public Function GetCommunityGroupBlogIds(communityGroupId As long) As long()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | ID of community group. |

**Returns**

An array of blog IDs (Long).

### CommunityGroup.GetCommunityGroupByID Method

Retrieves a CommunityGroupData object by ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupByID", Description = "Retrieves a
CommunityGroupData object by ID.")]
public CommunityGroupData GetCommunityGroupByID(long groupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupByID", Description = "Retrieves a
CommunityGroupData object by ID.")>
Public Function GetCommunityGroupByID(groupId As long) As CommunityGroupData
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of community group to retrieve. |

**Returns**

CommunityGroupData if groupId is valid, else null.

### CommunityGroup.GetCommunityGroupDiscussionBoardIds Method

Gets a list of Discussion Board IDs that belong to the community group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupDiscussionBoardIds", Description = "Gets a list of
Discussion Board IDs that belong to the community group.")]
public long[] GetCommunityGroupDiscussionBoardIds(long communityGroupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupDiscussionBoardIds", Description = "Gets a list of
Discussion Board IDs that belong to the community group.")>
```

```
Public Function GetCommunityGroupDiscussionBoardIds(communityGroupId As long) As long()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | ID of community group. |

**Returns**

An array of Discussion Board IDs (Long).

### CommunityGroup.GetCommunityGroups Method

Gets a list of community groups a user belongs to.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroups", Description = "Gets a list of community groups
a user belongs to.")]
public CommunityGroupData[] GetCommunityGroups(long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroups", Description = "Gets a list of community groups
a user belongs to.")>
Public Function GetCommunityGroups(userId As long, pagingInfo As PagingInfo) As
CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of a user to retrieve community groups for. |
| pagingInfo | Paging information for list retrieval. |

**Returns**

An array of CommunityGroupData objects.

### CommunityGroup.GetCommunityGroupUserIDs Method

Gets a list of user IDs that belong to the community group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupUserIDs", Description = "Gets a list of User IDs
that belong to the community group.")]
public long[] GetCommunityGroupUserIDs(long communityGroupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupUserIDs", Description = "Gets a list of User IDs
that belong to the community group.")>
Public Function GetCommunityGroupUserIDs(communityGroupId As long) As long()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | ID of community group. |

### Returns

An array of discussion board IDs (Long).

## GetCommunityGroupUsers Method

This is the overview for the GetCommunityGroupUsers method overload.

### Description

This method is overloaded.

- GetCommunityGroupUsers (Long) - retrieves a list of users in the community group.
- GetCommunityGroupUsers (Long, PagingInfo) - retrieves a list of users in the community group and allows you to page the results.

### Overload List

| | Name | Description |
|---|---|---|
| | CommunityGroup.GetCommunityGroupUsers (long) (see page 1009) | Retrieves a list of users in the community group. |
| | CommunityGroup.GetCommunityGroupUsers (long, PagingInfo) (see page 1009) | Returns a list of users in a community group. |

## CommunityGroup.GetCommunityGroupUsers Method (long)

Retrieves a list of users in the community group.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupUsers", Description = "Retrieves a list of users in
the community group.")]
public DirectoryUserData[] GetCommunityGroupUsers(long communityGroupId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupUsers", Description = "Retrieves a list of users in
the community group.")>
Public Function GetCommunityGroupUsers(communityGroupId As long) As DirectoryUserData()
```

### Parameters

| Parameters | Description |
|---|---|
| communityGroupId | ID of the community group whose users are being retrieved. |

### Returns

An array of DirectoryUserData.

## CommunityGroup.GetCommunityGroupUsers Method (long, PagingInfo)

Returns a list of users in a community group.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCommunityGroupUsers with paging info", Description = "Returns a
```

```
list of users in a community group.")]
public DirectoryUserData[] GetCommunityGroupUsers(long communityGroupId, PagingInfo
pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCommunityGroupUsers with paging info", Description = "Returns a
list of users in a community group.")>
Public Function GetCommunityGroupUsers(communityGroupId As long, pagingInfo As PagingInfo) As
DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | ID of community group. |
| pagingInfo | Paging information for the user list. |

**Returns**

An array of DirectoryUserData objects.

### CommunityGroup.GetFolderIdForCommunityGroup Method

Gets the folder ID used by the Community (see page 718) Group in the CMS content tree.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderIdForCommunityGroup", Description = "Gets the folder ID
used by the Community Group in the CMS content tree.")]
public long GetFolderIdForCommunityGroup(long CommunityGroupID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderIdForCommunityGroup", Description = "Gets the folder ID
used by the Community Group in the CMS content tree.")>
Public Function GetFolderIdForCommunityGroup(CommunityGroupID As long) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| CommunityGroupID | The ID of the community group. |

**Returns**

Retuns a Folder (see page 1048) ID (Long).

### CommunityGroup.GetGroupMemberStatus Method

Gets the Member status of a user for a particular group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetGroupMemberStatus", Description = "Gets the Member status of a
user for a particular group.")]
public EkEnumeration.GroupMemberStatus GetGroupMemberStatus(long communityGroupId, long
userID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "GetGroupMemberStatus", Description = "Gets the Member status of a
user for a particular group.")>
Public Function GetGroupMemberStatus(communityGroupId As long, userID As long) As
EkEnumeration.GroupMemberStatus
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | ID of community group to get user status for. |
| userID | ID of the user. |

**Returns**

GroupMemberStatus enumeration

## CommunityGroup.GetInvitedCommunityGroups Method

Returns a list of Community (see page 718) Groups that a user has been invited to join.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetInvitedCommunityGroups", Description = "Returns a list of
Community Groups that a user has been invited to.")]
public CommunityGroupData[] GetInvitedCommunityGroups(long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetInvitedCommunityGroups", Description = "Returns a list of
Community Groups that a user has been invited to.")>
Public Function GetInvitedCommunityGroups(userId As long, pagingInfo As PagingInfo) As
CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user who was invited to join the community group. |
| pagingInfo | Object to supply paging parameters. |

**Returns**

An array of CommunityGroupData.

## GetMostPopularCommunityGroups Method

This is the overview for the GetMostPopularCommunityGroups method overload.

**Description**

This method is overloaded.

- GetMostPopularCommunityGroups (PagingInfo) (see page 1012) - retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group.
- GetMostPopularCommunityGroups (int) (see page 1012) - retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a group is determined by how many members belong to the group.

**Overload List**

| | Name | Description |
|---|---|---|
| | CommunityGroup.GetMostPopularCommunityGroups (int) (see page 1012) | Retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a community group is determined by how many members belong to the group. |
| | CommunityGroup.GetMostPopularCommunityGroups (PagingInfo) (see page 1012) | Retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group. |

## CommunityGroup.GetMostPopularCommunityGroups Method (PagingInfo)

Retrieves all community groups ranked by popularity with the ability to have paging. The popularity of a group is determined by how many members belong to the group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMostPopularCommunityGroups with paging info", Description =
"Retrieves the most popular community groups.")]
public CommunityGroupData[] GetMostPopularCommunityGroups(PagingInfo pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMostPopularCommunityGroups with paging info", Description =
"Retrieves the most popular community groups.")>
Public Function GetMostPopularCommunityGroups(pagingInfo As PagingInfo) As CommunityGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| pagingInfo | Paging information for paging through large list of groups. |

**Returns**

An array of CommunityGroupData objects.

## CommunityGroup.GetMostPopularCommunityGroups Method (int)

Retrieves a list of the most popular community groups up to the number entered in the maxResults property. The popularity of a community group is determined by how many members belong to the group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMostPopularCommunityGroups", Description = "Retrieves the most
popular community groups.")]
public CommunityGroupData[] GetMostPopularCommunityGroups(int maxResults);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMostPopularCommunityGroups", Description = "Retrieves the most
popular community groups.")>
Public Function GetMostPopularCommunityGroups(maxResults As Integer) As CommunityGroupData()
```

### Parameters

| Parameters | Description |
|---|---|
| maxResults | The number of groups to retrieve. For example, if 10 is supplied, the top 10 most popular groups will be returned. |

### Returns

An array of CommunityGroupData objects.

## CommunityGroup.GetPendingCommunityGroups Method

Gets a list of community groups for which the user's membership is pending the Group administrator's approval.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPendingCommunityGroups", Description = "Gets a list of community
groups for which the user's membership is pending the Group administrator's approval.")]
public CommunityGroupData[] GetPendingCommunityGroups(long userId, PagingInfo pagingInfo);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPendingCommunityGroups", Description = "Gets a list of community
groups for which the user's membership is pending the Group administrator's approval.")>
Public Function GetPendingCommunityGroups(userId As long, pagingInfo As PagingInfo) As
CommunityGroupData()
```

### Parameters

| Parameters | Description |
|---|---|
| userId | ID of user to retrieve pending groups for. |
| pagingInfo | Paging information for retrieval. |

## CommunityGroup.GetPendingCommunityGroupUsers Method

Returns a list of users whose community group membership is pending approval.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPendingCommunityGroupUsers", Description = "Returns a list of
users who's community group membership is pending approval.")]
public DirectoryUserData[] GetPendingCommunityGroupUsers(long communityGroupID, PagingInfo
pagingInfo);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPendingCommunityGroupUsers", Description = "Returns a list of
users who's community group membership is pending approval.")>
Public Function GetPendingCommunityGroupUsers(communityGroupID As long, pagingInfo As
PagingInfo) As DirectoryUserData()
```

### Parameters

| Parameters | Description |
|---|---|
| pagingInfo | Paging information for the user list. |
| communityGroupId | ID of community group. |

### Returns

An array of DirectoryUserData objects.

### CommunityGroup.InviteUsers Method

Invites the users to the community group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "InviteUsers", Description = "Invites the users to the community
group.")]
public Collection<Invitation> InviteUsers(InvitationSendRequestData invitationData, long
groupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "InviteUsers", Description = "Invites the users to the community
group.")>
Public Function InviteUsers(invitationData As InvitationSendRequestData, groupId As long) As
Collection<Invitation>
```

**Parameters**

| Parameters | Description |
|---|---|
| invitationData | InvitationSendRequestData containing information about the GroupInvitation to send. |
| groupId | ID of the community group to invite people to join. |

**Returns**

A collection of invitations that were sent out.

### CommunityGroup.IsCommunityGroupAdmin Method

Returns true if the user is an administrator of the group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsCommunityGroupAdmin", Description = "Returns true if the user is
an admin of the group.")]
public bool IsCommunityGroupAdmin(long userId, long groupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsCommunityGroupAdmin", Description = "Returns true if the user is
an admin of the group.")>
Public Function IsCommunityGroupAdmin(userId As long, groupId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user to verify. |
| groupId | ID of the group to check. |

### CommunityGroup.IsUserInCommunityGroup Method

Returns true if the user is a member of the community group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsUserInCommunityGroup", Description = "Returns true if the  user is
a member of the community group.")]
```

```
public bool IsUserInCommunityGroup(long communityGroupId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsUserInCommunityGroup", Description = "Returns true if the  user is
a member of the community group.")>
Public Function IsUserInCommunityGroup(communityGroupId As long, userId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| communityGroupId | ID of community group to check membership. |
| userId | User (see page 1167) ID to check for. |

**Returns**

True if the user is a member of the community group.

### CommunityGroup.RemoveUserFromCommunityGroup Method

Removes a user from a community group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RemoveUserFromCommunityGroup", Description = "Removes a user from a
community group.")]
public void RemoveUserFromCommunityGroup(long groupId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveUserFromCommunityGroup", Description = "Removes a user from a
community group.")>
Public Function RemoveUserFromCommunityGroup(groupId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | ID of the community group. |
| userId | ID of the user to remove. |

### CommunityGroup.UpdateCommunityGroup Method

Updates an existing community group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateCommunityGroup", Description = "Updates an existing community
group.")]
public void UpdateCommunityGroup(CommunityGroupData groupData);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateCommunityGroup", Description = "Updates an existing community
group.")>
Public Function UpdateCommunityGroup(groupData As CommunityGroupData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| groupData | CommunityGroupData class containing information to update for the group. |

# 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)]

## Content Methods

The methods of the Content class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddContent (see page 1017) | This is the overview for the AddContent method overload. |
| | CheckOutContent (see page 1019) | 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 1020) | Copy the content into required folder. |
| | DeleteContentItem (see page 1020) | Delete the content by the content's ID. |
| | GetAddViewLanguage (see page 1021) | 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 1021) | Loads all of the comments. |
| | GetAllTemplates (see page 1021) | Loads all of the templates. |
| | GetAllUnassignedItemApprovals (see page 1022) | Returns all of the unassigned user/groups for a given content ID. |
| | GetChildContent (see page 1022) | This is the overview for the GetChildContent method overload. |
| | GetContent (see page 1024) | Get Content (see page 1016) By ID. |
| | GetContentByHistory (see page 1024) | Retrieves the corresponding content by the history ID. |
| | GetContentForEditing (see page 1025) | Loads the content details with checkout mode. |
| | GetContentState (see page 1025) | Loads the content's details. |
| | GetContentStatus (see page 1025) | 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 1026) | Gets web domain for a piece of content. |

| | | |
|---|---|---|
| | MoveContentToFolder (see page 1026) | Moves content to a folder. |
| | PublishContent (see page 1026) | 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. |
| | RenameContent (see page 1027) | Renames the content. |
| | SaveContent (see page 1027) | 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 1028) | Delete the content by the content's ID and folder ID. |
| | UndoCheckout (see page 1028) | This undoes the checkout for a piece of content which is checked out to the current user. |
| | UpdateContentMetaData (see page 1029) | Updates a specific metadata item attached to a specified piece of content. |

## AddContent Method

This is the overview for the AddContent method overload.

### Description

Adds Content to the CMS. This method is overloaded.

### Overload List

| | Name | Description |
|---|---|---|
| | Content.AddContent (string, string, string, string, string, string, long, object, object, string) (see page 1017) | Adds Content (see page 1016) to the CMS |
| | Content.AddContent (string, string, string, string, string, string, long, object, object, string, int) (see page 1018) | Adds Content (see page 1016) to the CMS |
| | Content.AddContent (string, string, string, string, string, string, long, object, object, string, int, int) (see page 1019) | Adds Content (see page 1016) to the CMS |

## Content.AddContent Method (string, string, string, string, string, string, long, object, object, string)

Adds Content (see page 1016) to the CMS

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContent", Description = "Adds a piece of content.")]
public long AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, long FolderId, object GoLive, object
EndDate, string MetaInfoXml);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContent", Description = "Adds a piece of content.")>
Public Function AddContent(ContentTitle As string, ContentComment As string, ContentHtml As
string, SearchText As string, SummaryHtml As string, ContentLanguage As string, FolderId As
long, GoLive As object, EndDate As object, MetaInfoXml As string) As long
```

### Parameters

| Parameters | Description |
|---|---|
| ContentTitle | Title of content to add |
| ContentComment | Content (see page 1016) comments |
| ContentHtml | Content (see page 1016) HTML to display |
| SearchText | Search (see page 864) Text |
| SummaryHtml | Summary HTML |
| ContentLanguage | Content (see page 1016) Language |
| FolderId | Folder (see page 1048) to place the content in |
| GoLive | Go Live Date |
| EndDate | Content (see page 1016) End Date |
| MetaInfoXml | Meta Info XML |

### Returns

Returns the numeric ID of newly created content.

## Content.AddContent Method (string, string, string, string, string, string, long, object, object, string, int)

Adds Content (see page 1016) to the CMS

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContent With ContentID", Description = "Adds a piece of
content.")]
public long AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, long FolderId, object GoLive, object
EndDate, string MetaInfoXml, int ContentID);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContent With ContentID", Description = "Adds a piece of
content.")>
Public Function AddContent(ContentTitle As string, ContentComment As string, ContentHtml As
string, SearchText As string, SummaryHtml As string, ContentLanguage As string, FolderId As
long, GoLive As object, EndDate As object, MetaInfoXml As string, ContentID As Integer) As long
```

### Parameters

| Parameters | Description |
|---|---|
| ContentTitle | Title of content to add |
| ContentComment | Content (see page 1016) comments |
| ContentHtml | Content (see page 1016) HTML to display |
| SearchText | Search (see page 864) Text |
| SummaryHtml | Summary HTML |
| ContentLanguage | Content (see page 1016) Language |
| FolderId | Folder (see page 1048) to place the content in |
| GoLive | Go Live Date |
| EndDate | Content (see page 1016) End Date |
| MetaInfoXml | Meta Info XML |
| ContentID | Content (see page 1016) ID |

### Returns

The numeric ID of newly created content.

## Content.AddContent Method (string, string, string, string, string, string, long, object, object, string, int, int)

Adds Content (see page 1016) to the CMS

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContent with XmlID and TemplateID", Description = "Adds a piece
of content.")]
public long AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, long FolderId, object GoLive, object
EndDate, string MetaInfoXml, int XmlId, int TemplateID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContent with XmlID and TemplateID", Description = "Adds a piece
of content.")>
Public Function AddContent(ContentTitle As string, ContentComment As string, ContentHtml As
string, SearchText As string, SummaryHtml As string, ContentLanguage As string, FolderId As
long, GoLive As object, EndDate As object, MetaInfoXml As string, XmlId As Integer, TemplateID
As Integer) As long
```

### Parameters

| Parameters | Description |
|---|---|
| ContentTitle | Title of content to add |
| ContentComment | Content (see page 1016) comments |
| ContentHtml | Content (see page 1016) HTML to display |
| SearchText | Search (see page 864) Text |
| SummaryHtml | Summary HTML |
| ContentLanguage | Content (see page 1016) Language |
| FolderId | Folder (see page 1048) to place the content in |
| GoLive | Go Live Date |
| EndDate | Content (see page 1016) End Date |
| XmlId | XmlId |
| TemplateID | TemplateID |

### Returns

The numeric ID of newly created content.

## Content.CheckOutContent Method

This will check the content out to the current user if the content is currently checked in and the user has privileges to check it out.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CheckOutContent", Description = "Checks the content out to the
current user")]
public bool CheckOutContent(long ContentID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "CheckOutContent", Description = "Checks the content out to the
current user")>
Public Function CheckOutContent(ContentID As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | Id of the content to check out |

**Returns**

A Boolean representing the success of the check out

## Content.CopyContentToFolder Method

Copy the content into required folder.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CopyContentToFolder", Description = "Copy the content into required
folder.")]
public ContentData CopyContentToFolder(string ContentID, long FolderID, string Language, bool
Publish);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CopyContentToFolder", Description = "Copy the content into required
folder.")>
Public Function CopyContentToFolder(ContentID As string, FolderID As long, Language As string,
Publish As bool) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | Content (see page 1016) ID |
| FID | Folder (see page 1048) ID |
| Lang | Language ID |

**Returns**

Ektron.Cms.ContentData (see page 1378)

## Content.DeleteContentItem Method

Delete the content by the content's ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteContentItem", Description = "Delete the content by the
content's ID")]
public bool DeleteContentItem(long ContentID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteContentItem", Description = "Delete the content by the
content's ID")>
Public Function DeleteContentItem(ContentID As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| ContentID | The content's numeric ID. |

## Content.GetAddViewLanguage Method

Loads all the languages from the system for a given content ID and populates the Add/View drop down box with flag set on each item.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAddViewLanguage", Description = "Gets all the languages from the
system for a given content ID ")]
public LanguageData[] GetAddViewLanguage(long ContentID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAddViewLanguage", Description = "Gets all the languages from the
system for a given content ID ")>
Public Function GetAddViewLanguage(ContentID As long) As LanguageData()
```

### Parameters

| Parameters | Description |
|---|---|
| ContentID | The Content (see page 1016)'s numeric ID. |

## Content.GetAllComments Method

Loads all of the comments.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllComments", Description = "Gets all comments")]
public CommentData[] GetAllComments(long KeyID, long CommentID, long RefID, string RefType,
long UserID, string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllComments", Description = "Gets all comments")>
Public Function GetAllComments(KeyID As long, CommentID As long, RefID As long, RefType As
string, UserID As long, OrderBy As string) As CommentData()
```

### Parameters

| Parameters | Description |
|---|---|
| KeyID | The numeric key ID. |
| CommentID | The comment's numeric ID. |
| RefID | Reference ID, (content or forms). |
| RefType | Reference type, C = content or T = Task (see page 956) |
| UserID | The user's numeric ID. |
| OrderBy | The order of the comments. |

## Content.GetAllTemplates Method

Loads all of the templates.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTemplates", Description = "Gets all templates")]
public TemplateData[] GetAllTemplates(string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTemplates", Description = "Gets all templates")>
Public Function GetAllTemplates(OrderBy As string) As TemplateData()
```

**Parameters**

| Parameters | Description |
|---|---|
| OrderBy | Order by clause |

## Content.GetAllUnassignedItemApprovals Method

Returns all of the unassigned user/groups for a given content ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUnassignedItemApprovals", Description = "Returns all of the
unassigned user/groups for a given content ID")]
public ApprovalData[] GetAllUnassignedItemApprovals(long ContentID, string ItemType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUnassignedItemApprovals", Description = "Returns all of the
unassigned user/groups for a given content ID")>
Public Function GetAllUnassignedItemApprovals(ContentID As long, ItemType As string) As
ApprovalData()
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The content's numeric ID. |
| ItemType | The type of user. |

## GetChildContent Method

This is the overview for the GetChildContent method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Content.GetChildContent (long, bool, string) (see page 1022) | Loads all of the content for the given folder. |
| | Content.GetChildContent (long, bool, string, PagingInfo) (see page 1023) | Loads all of the contents for the given folder. |

## Content.GetChildContent Method (long, bool, string)

Loads all of the content for the given folder.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetChildContent", Description = "Loads all of the contents for the
given folder")]
public ContentData[] GetChildContent(long FolderID, bool Recursive, string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetChildContent", Description = "Loads all of the contents for the
given folder")>
Public Function GetChildContent(FolderID As long, Recursive As bool, OrderBy As string) As
ContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The folder's numeric ID. |
| Recursive | True = recursive look up. False = none. |
| OrderBy | An order by clause. Choices are:<br>• **id** - the content ID.<br>• **title** - the title of the content item.<br>• **status** - whether the content is checked-in, published, checked-out, etc.<br>• **language** - the language ID of the content.<br>• **datecreated** - the date the content was created.<br>• **datemodified** - the date the content was last modified.<br>• **userid** - the ID of the user who last worked with the content. |

**Description**

## Content.GetChildContent Method (long, bool, string, PagingInfo)

Loads all of the contents for the given folder.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetChildContentWithPaging", Description = "Loads all of the contents
for the given folder")]
public ContentData[] GetChildContent(long FolderID, bool Recursive, string OrderBy, PagingInfo
pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetChildContentWithPaging", Description = "Loads all of the contents
for the given folder")>
Public Function GetChildContent(FolderID As long, Recursive As bool, OrderBy As string,
pagingInfo As PagingInfo) As ContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Recursive | True = recursive look up. False = none |

| | |
|---|---|
| OrderBy | An order by clause. Choices are:<br>• **id** - the content ID.<br>• **title** - the title of the content item.<br>• **status** - whether the content is checked-in, published, checked-out, etc.<br>• **language** - the language ID of the content.<br>• **datecreated** - the date the content was created.<br>• **datemodified** - the date the content was last modified.<br>• **userid** - the ID of the user who last worked with the content. |
| Id | the folders ID |
| pagignInfo | Paging Information |

**Returns**

Ektron.Cms.ContentData (see page 1378)()

**Description**

## Content.GetContent Method

Get Content (see page 1016) By ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContent", Description = "Gets the content details by content id")]
public ContentData GetContent(long ContentID, Ektron.Cms.Content.EkContent.ContentResultType
ResultType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContent", Description = "Gets the content details by content id")>
Public Function GetContent(ContentID As long, ResultType As
Ektron.Cms.Content.EkContent.ContentResultType) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The content's numeric ID. |
| Preview | Preview |
| Published | Is Published |

## Content.GetContentByHistory Method

Retrieves the corresponding content by the history ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentByHistory", Description = "Retrieves the corresponding
content by the history ID")]
public ContentData GetContentByHistory(long HistoryID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentByHistory", Description = "Retrieves the corresponding
content by the history ID")>
```

```
Public Function GetContentByHistory(HistoryID As long) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| HistoryID | The history's numeric ID. |

### Content.GetContentForEditing Method

Loads the content details with checkout mode.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentForEditing", Description = "Loads the content details with
checkout mode")]
public ContentEditData GetContentForEditing(long ContentID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentForEditing", Description = "Loads the content details with
checkout mode")>
Public Function GetContentForEditing(ContentID As long) As ContentEditData
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The content's numeric ID. |

### Content.GetContentState Method

Loads the content's details.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentState", Description = "Gets the content's details")]
public ContentStateData GetContentState(long ContentID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentState", Description = "Gets the content's details")>
Public Function GetContentState(ContentID As long) As ContentStateData
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The content's numeric ID. |

### Content.GetContentStatus Method

This will return a single letter string which is the status of the piece of content. The letters returned are the same letters used in the Workarea view.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentStatus", Description = "Gets single letter string which is
the status of the piece of content")]
public string GetContentStatus(long ContentID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentStatus", Description = "Gets single letter string which is
the status of the piece of content")>
Public Function GetContentStatus(ContentID As long) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The numeric ID of the content for which to return a status. |

## Content.GetDomain Method

Gets web domain for a piece of content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDomain", Description = "Gets web domain for a piece of content")]
public string GetDomain(long ContentID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDomain", Description = "Gets web domain for a piece of content")>
Public Function GetDomain(ContentID As long) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The content's ID |

## Content.MoveContentToFolder Method

Moves content to a folder.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MoveContentToFolder", Description = "Moves content to a folder")]
public ContentData MoveContentToFolder(string ContentID, string ContentLanguage, long
FolderID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MoveContentToFolder", Description = "Moves content to a folder")>
Public Function MoveContentToFolder(ContentID As string, ContentLanguage As string, FolderID
As long) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The content's ID. |
| ContentLanguage | The content's language. |
| FolderID | The folder's numeric ID. |

## Content.PublishContent Method

Function takes in information about the content to be checked in and published. This goes through the normal approval chain. The content must be in a checked out state to the current logged in user for this to succeed.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "PublishContent", Description = "Function takes in information about
the content to be checked in and published. This goes through the normal approval chain. The
content must be in a checked out state to the current logged in user for this to succeed")]
public bool PublishContent(long ContentID, long FolderID, int ContentLanguage, string
DontCreateTask, long UserID, string TaskTitle);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "PublishContent", Description = "Function takes in information about
the content to be checked in and published. This goes through the normal approval chain. The
content must be in a checked out state to the current logged in user for this to succeed")>
Public Function PublishContent(ContentID As long, FolderID As long, ContentLanguage As
Integer, DontCreateTask As string, UserID As long, TaskTitle As string) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The numeric value of the content's ID. |
| FolderID | The numeric value of the content folder's ID. |
| ContentLanguage | The numeric value of the content's language ID. |
| DontCreateTask | If this parameter is "", a task will be created for this submit. Otherwise, no task is created. |
| UserID | The user ID of task assigned by current user. This is needed only if "dontCreateTask" is "", otherwise it is ignored. |
| TaskTitle | Title for the task. This is needed only if "dontCreateTask" is "", otherwise it is ignored. |

## Content.RenameContent Method

Renames the content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenameContent", Description = "Renames the content by ID")]
public ContentData RenameContent(long ContentID, string Title);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RenameContent", Description = "Renames the content by ID")>
Public Function RenameContent(ContentID As long, Title As string) As ContentData
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The content's numeric ID. |
| Title | The content's new title. |

## Content.SaveContent Method

Updates the checked out content block. Only following properties can be updated: contentData.Title contentData.Teaser contentData.Comment contentData.GoLive contentData.Html contentData.EndDate contentData.EndDateAction contentData.ManualAlias

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SaveContent", Description = "Updates the checked out content
block.")]
```

```
public bool SaveContent(ContentEditData contentEditData);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SaveContent", Description = "Updates the checked out content
block.")>
Public Function SaveContent(contentEditData As ContentEditData) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| contentData | |

**Returns**

False if successful

**Remarks**

Content (see page 1016) must be in the checkedout state.

## Content.SubmitForDelete Method

Delete the content by the content's ID and folder ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SubmitForDelete", Description = "Deletes the content by the
content's Id and folder Id")]
public bool SubmitForDelete(long ContentID, long FolderID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SubmitForDelete", Description = "Deletes the content by the
content's Id and folder Id")>
Public Function SubmitForDelete(ContentID As long, FolderID As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentID | The content's numeric ID. |
| FolderID | The content's numeric folder ID. |

## Content.UndoCheckout Method

This undoes the checkout for a piece of content which is checked out to the current user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UndoCheckout", Description = "Reverts the checkout for a piece of
content which is checked out to the current user")]
public bool UndoCheckout(long ContentID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UndoCheckout", Description = "Reverts the checkout for a piece of
content which is checked out to the current user")>
Public Function UndoCheckout(ContentID As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| ContentID | The numeric ID of the content for which to undo check out. |

### Content.UpdateContentMetaData Method

Updates a specific metadata item attached to a specified piece of content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateContentMetaData", Description = "Updates Content Metadata")]
public bool UpdateContentMetaData(long ContentID, long MetaDataTypeID, string MetaDataText);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateContentMetaData", Description = "Updates Content Metadata")>
Public Function UpdateContentMetaData(ContentID As long, MetaDataTypeID As long, MetaDataText
As string) As bool
```

## ContentRating Class

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

## ContentRating Methods

The methods of the ContentRating class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddContentRating (see page 1030) | Adds a content rating to the database. Internally a check is done to ensure the user has not already rated the content, and passes down the rating and any comments the user supplied. |
| | GetContentRatingResults (see page 1030) | Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments. |
| | PurgeContentRatings (see page 1030) | Purges all ratings of a particular content block over a defined date/time range. |

## ContentRating.AddContentRating Method

Adds a content rating to the database. Internally a check is done to ensure the user has not already rated the content, and passes down the rating and any comments the user supplied.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContentRating", Description = "")]
public void AddContentRating(long contentid, long userid, int userRating, string userComments);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContentRating", Description = "")>
Public Function AddContentRating(contentid As long, userid As long, userRating As Integer,
userComments As string) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| contentid | The numeric ID associated with the content 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 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentRatingResults", Description = "")]
public System.Data.DataSet GetContentRatingResults(long contentID, DateTime startDate,
DateTime endDate);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentRatingResults", Description = "")>
Public Function GetContentRatingResults(contentID As long, startDate As DateTime, endDate As
DateTime) As System.Data.DataSet
```

**Parameters**

| Parameters | Description |
|---|---|
| contentID | The numeric ID associated with the content 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 block over a defined date/time range.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "PurgeContentRatings", Description = "")]
public void PurgeContentRatings(long contentid, DateTime startDate, DateTime endDate);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "PurgeContentRatings", Description = "")>
Public Function PurgeContentRatings(contentid As long, startDate As DateTime, endDate As
DateTime) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| contentid | The ID of the content that is having its ratings purged. |
| startDate | The beginning of the date range of content to be purged. |
| endDate | The end of the date range of content 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

## CustomFields Methods

The methods of the CustomFields class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddNewEditableCustomFieldAssignments (see page 1032) | Returns the HTML string for the add/edit custom fields. |
| | BoolToYesNo (see page 1033) | Returns labels for yes or no based on locale. |

| | | |
|---|---|---|
| | GetEditableCustomFieldAssignments (see page 1033) | Returns the editable custom fields.<br>Break Inheritance checkbox:<br>• Visible if not at root.<br>• Checked if any fields assigned without inheritance.<br>• If unchecked & form saved, removes all assignments.<br>• If checked & form saved and no assignments chosen, creates NULL assignment (CustomFieldID -1).<br>• When checked, enables all assignment checkboxes (iterate through all of this tables elements).<br>• When unchecked, disables all assignment checkboxes (iterate through all of this tables elements).<br>• Assignment Checkboxes:<br>• When clicked: if 'Assignment' button -> going unchecked, then ensure required us unchecked.<br>• When clicked: if 'Required' button, don't accept checks if assigned is unchecked.<br>• When form submitted, copy checked... more (see page 1033) |
| | GetEnhancedMetadataArea (see page 1034) | Returns the HTML needed to render the EnhancedMetadata selection area. |
| | GetEnhancedMetadataScript (see page 1034) | Returns support Javascript code for enhanced metadata. |
| | GetFieldsByFolder (see page 1035) | Returns all of the custom attributes by folder. |
| | GetSearchProperties (see page 1035) | Returns an HTML string for the search properties display. |
| | ProcessCustomFields (see page 1035) | Process Folders' Custom-Field (and metadata) assignments:<br>When form submitted, copy checked items to hidden field. Form (see page 1058) 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 1036) | Returns the search result based on assets. |
| | WriteFilteredMetadataForEdit (see page 1036) | Returns an HTML string for filtered metadata. |
| | WriteFilteredMetadataForView (see page 1037) | Returns an HTML string for metadata view. |
| | WriteLibrarySearchExtended (see page 1037) | Returns an HTML string for the library search custom metadata properties. |
| | WriteMetadataDefaultForEdit (see page 1037) | Returns an HTML string for the default metadata. |
| | WriteMetadataTypeForView (see page 1038) | Returns a metadata type for view. |
| | WritePublicSearch (see page 1038) | Returns WritepublicsearchEx. |

## CustomFields.AddNewEditableCustomFieldAssignments Method

Returns the HTML string for the add/edit custom fields.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddNewEditableCustomFieldAssignments", Description = "")]
public string AddNewEditableCustomFieldAssignments(long parentfolderid);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddNewEditableCustomFieldAssignments", Description = "")>
Public Function AddNewEditableCustomFieldAssignments(parentfolderid As long) As string
```

### Parameters

| Parameters | Description |
|---|---|
| parentfolderid | The parent folder's ID. |

### Returns

String

## CustomFields.BoolToYesNo Method

Returns labels for yes or no based on locale.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "BoolToYesNo", Description = "")]
public string BoolToYesNo(bool Value);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "BoolToYesNo", Description = "")>
Public Function BoolToYesNo(Value As bool) As string
```

### Parameters

| Parameters | Description |
|---|---|
| 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 (see page 1058) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetEditableCustomFieldAssignments", Description = "")]
public string GetEditableCustomFieldAssignments(object folderID, object editMode);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetEditableCustomFieldAssignments", Description = "")>
Public Function GetEditableCustomFieldAssignments(folderID As object, editMode As object) As
string
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetEnhancedMetadataArea", Description = "")]
public string GetEnhancedMetadataArea();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetEnhancedMetadataArea", Description = "")>
Public Function GetEnhancedMetadataArea() As string
```

### CustomFields.GetEnhancedMetadataScript Method

Returns support Javascript code for enhanced metadata.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetEnhancedMetadataScript", Description = "")]
public string GetEnhancedMetadataScript();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetEnhancedMetadataScript", Description = "")>
Public Function GetEnhancedMetadataScript() As string
```

**Returns**

String

### CustomFields.GetFieldsByFolder Method

Returns all of the custom attributes by folder.

**C#**

```
public Collection GetFieldsByFolder(long FolderID, int LanguageID);
```

**Visual Basic**

```
Public Function GetFieldsByFolder(FolderID As long, LanguageID As Integer) As Collection
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The folder's ID. |
| LanguageID | The language's ID. |

**Returns**

Microsoft.VisualBasic.Collection (see page 88)

### CustomFields.GetSearchProperties Method

Returns an HTML string for the search properties display.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSearchProperties", Description = "")]
public StringBuilder GetSearchProperties(SearchAssetDispayRequest sadReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSearchProperties", Description = "")>
Public Function GetSearchProperties(sadReq As SearchAssetDispayRequest) As StringBuilder
```

**Parameters**

| Parameters | Description |
|---|---|
| sadReq | The search asset objects. (Required) |

**Returns**

System.Text.StringBuilder

### CustomFields.ProcessCustomFields Method

Process Folders' Custom-Field (and metadata) assignments:

When form submitted, copy checked items to hidden field. Form (see page 1058) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ProcessCustomFields", Description = "")]
public object ProcessCustomFields(long folderID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "ProcessCustomFields", Description = "")>
Public Function ProcessCustomFields(folderID As long) As object
```

**Parameters**

| Parameters | Description |
|---|---|
| folderID | The folder's ID. |

**Returns**

Object

### CustomFields.SearchAllAssets Method

Returns the search result based on assets.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SearchAllAssets", Description = "")]
public StringBuilder SearchAllAssets(SearchAssetDispayRequest sadReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SearchAllAssets", Description = "")>
Public Function SearchAllAssets(sadReq As SearchAssetDispayRequest) As StringBuilder
```

**Parameters**

| Parameters | Description |
|---|---|
| sadReq | Search (see page 864) 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 1106) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "WriteFilteredMetadataForView", Description = "")]
public string WriteFilteredMetadataForView(ContentMetaData[] MetadataTypes, long FolderId,
bool SearchableOnly);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "WriteFilteredMetadataForView", Description = "")>
Public Function WriteFilteredMetadataForView(MetadataTypes As ContentMetaData[], FolderId As
long, SearchableOnly As bool) As string
```

### Parameters

| Parameters | Description |
|---|---|
| MetadataTypes | Metadata (see page 1106) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "WriteLibrarySearchExtended", Description = "")]
public StringBuilder WriteLibrarySearchExtended(long folderId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "WriteLibrarySearchExtended", Description = "")>
Public Function WriteLibrarySearchExtended(folderId As long) As StringBuilder
```

### Parameters

| Parameters | Description |
|---|---|
| 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 1106) values. |

**Returns**

System.Text.StringBuilder

**Description**

[WebMethod(MessageName = "WriteMetadataDefaultForEdit", Description = "")]

### CustomFields.WriteMetadataTypeForView Method

Returns a metadata type for view.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "WriteMetadataTypeForView", Description = "")]
public string WriteMetadataTypeForView(string DataType, string Caption, string Value);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "WriteMetadataTypeForView", Description = "")>
Public Function WriteMetadataTypeForView(DataType As string, Caption As string, Value As
string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| DataType | Data types. |
| Caption | Caption value. |
| Value | Value. (Required) |

**Returns**

String

### CustomFields.WritePublicSearch Method

Returns WritepublicsearchEx.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "WritePublicSearch", Description = "")]
public StringBuilder WritePublicSearch(SearchAssetDispayRequest sadReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "WritePublicSearch", Description = "")>
Public Function WritePublicSearch(sadReq As SearchAssetDispayRequest) As StringBuilder
```

### Parameters

| Parameters | Description |
|---|---|
| sadReq | Search (see page 864) asset data. (Required) |

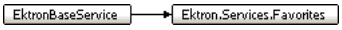
### Returns

System.Text.StringBuilder

# Favorites Class

Summary description for Favorites

### Class Hierarchy




### C#

```
public class Favorites : EktronBaseService;
```

### Visual Basic

```
Public Class Favorites
Inherits EktronBaseService
```

### File

Favorites.cs

## Favorites Methods

The methods of the Favorites class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddContentFavorite (see page 1039) | This is the overview for the AddContentFavorite method overload. |
| | DeleteContentFavorite (see page 1041) | Deletes a piece of content from a user's favorites. |
| | GetContentFavorites (see page 1041) | This is the overview for the GetContentFavorites method overload. |
| | GetFavoritesDirectory (see page 1043) | Returns a user's Favorites (see page 1039) taxonomy directory structure. |
| | IsContentFavorite (see page 1043) | Checks if a content item is in a user's favorites. |
| | MoveContentFavorite (see page 1044) | Moves a content favorite to a new favorites directory. |

### AddContentFavorite Method

This is the overview for the AddContentFavorite method overload.

### Description

This method is overloaded.

- AddContentFavorite (Long, Long) - adds a piece of content to user's favorites.
- AddContentFavorite (Long, Long, Long) - adds a piece of content to user's favorites and places it in a subdirectory.

### Overload List

| | Name | Description |
|---|---|---|
| | Favorites.AddContentFavorite (long, long) (see page 1040) | Adds a piece of content to user's favorites. |
| | Favorites.AddContentFavorite (long, long, long) (see page 1040) | Adds a piece of content to user's favorites and places it in a subdirectory. |

## Favorites.AddContentFavorite Method (long, long)

Adds a piece of content to user's favorites.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContentFavorite", Description = "Adds a piece of content to
user's favorites.")]
public void AddContentFavorite(long contentId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContentFavorite", Description = "Adds a piece of content to
user's favorites.")>
Public Function AddContentFavorite(contentId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content item. |
| userId | ID of the user. |

## Favorites.AddContentFavorite Method (long, long, long)

Adds a piece of content to user's favorites and places it in a subdirectory.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContentFavorite to directory", Description = "Adds a content item
to a user's favorites and places it in a subdirectory.")]
public void AddContentFavorite(long contentId, long userId, long directoryId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContentFavorite to directory", Description = "Adds a content item
to a user's favorites and places it in a subdirectory.")>
Public Function AddContentFavorite(contentId As long, userId As long, directoryId As long) As
void
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content item. |
| userId | ID of the user. |
| directoryId | ID of the directory to add the content in the user's favorites. |

## Favorites.DeleteContentFavorite Method

Deletes a piece of content from a user's favorites.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteContentFavorite", Description = "Deletes a content item from a
user's favorites.")]
public void DeleteContentFavorite(long contentId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteContentFavorite", Description = "Deletes a content item from a
user's favorites.")>
Public Function DeleteContentFavorite(contentId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content item. |
| userId | ID of the user. |

## GetContentFavorites Method

This is the overview for the GetContentFavorites method overload.

**Description**

This method is overloaded.

- GetContentFavorites (Long, PagingInfo) - returns a list of the user's favorites by providing the user's ID. This method allows you to page the favorites.
- GetContentFavorites (Long, Long) - returns a list of the user's favorites in a directory by providing the user's ID and the ID of the directory.
- GetContentFavorites (Long, Long, PagingInfo) - returns a list of the user's favorites in a directory by providing the user's ID and the ID of the directory. This method allows you to page the favorites.

**Overload List**

| | Name | Description |
|---|---|---|
| | Favorites.GetContentFavorites (long, long) (see page 1042) | Returns a list of the user's favorites in a directory. |
| | Favorites.GetContentFavorites (long, long, PagingInfo) (see page 1042) | Returns a list of the user's favorites in a directory. |
| | Favorites.GetContentFavorites (long, PagingInfo) (see page 1041) | Returns a list of the user's favorites. |

## Favorites.GetContentFavorites Method (long, PagingInfo)

Returns a list of the user's favorites.

**C#**

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentFavorites with paging info", Description = "Returns a list
of the user's favorites.")]
public DirectoryContentData[] GetContentFavorites(long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentFavorites with paging info", Description = "Returns a list
of the user's favorites.")>
Public Function GetContentFavorites(userId As long, pagingInfo As PagingInfo) As
DirectoryContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| pagingInfo | Paging information for limiting the resultset. |
| userID | ID of user to retrieve favorites for. |

**Returns**

An array of DirectoryContentData objects.

### Favorites.GetContentFavorites Method (long, long)

Returns a list of the user's favorites in a directory.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentFavorites", Description = "Returns a list of the user's
favorites from a specified directory.")]
public DirectoryContentData[] GetContentFavorites(long userId, long directoryId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentFavorites", Description = "Returns a list of the user's
favorites from a specified directory.")>
Public Function GetContentFavorites(userId As long, directoryId As long) As
DirectoryContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| directoryId | ID of the directory to retrieve favorites from. |
| userID | ID of the user to retrieve favorites for. |

**Returns**

An array of DirectoryContentData objects.

### Favorites.GetContentFavorites Method (long, long, PagingInfo)

Returns a list of the user's favorites in a directory.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentFavorites for directory with paging info", Description =
"Returns a list of the user's favorites in a directory.")]
public DirectoryContentData[] GetContentFavorites(long userId, long directoryId, PagingInfo
pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentFavorites for directory with paging info", Description =
"Returns a list of the user's favorites in a directory.")>
Public Function GetContentFavorites(userId As long, directoryId As long, pagingInfo As
PagingInfo) As DirectoryContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| directoryId | ID of directory to retrieve favorites from. |
| pagingInfo | Paging information for limiting the resultset. |
| userID | ID of user to retrieve favorites for. |

**Returns**

An array of DirectoryContentData objects.

### Favorites.GetFavoritesDirectory Method

Returns a user's Favorites (see page 1039) taxonomy directory structure.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFavoritesDirectory", Description = "Returns user's Favorites
taxonomy directory structure.")]
public TaxonomyData GetFavoritesDirectory(long userId, long directoryId, PagingInfo
pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFavoritesDirectory", Description = "Returns user's Favorites
taxonomy directory structure.")>
Public Function GetFavoritesDirectory(userId As long, directoryId As long, pagingInfo As
PagingInfo) As TaxonomyData
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of the user whose favorites should be retrieved. |
| directoryId | ID of the favorites directory to retrieve. |
| pagingInfo | Paging information for limiting the resultset. |

### Favorites.IsContentFavorite Method

Checks if a content item is in a user's favorites.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsContentFavorite", Description = "Checks if a content item is in a
user's favorites.")]
public bool IsContentFavorite(long contentId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsContentFavorite", Description = "Checks if a content item is in a
user's favorites.")>
```

```
Public Function IsContentFavorite(contentId As long, userId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content item. |
| userId | ID of the user. |

**Returns**

True if the content item is in user's favorites, else false.

### Favorites.MoveContentFavorite Method

Moves a content favorite to a new favorites directory.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MoveContentFavorite", Description = "Moves a content favorite to a
new favorites directory.")]
public void MoveContentFavorite(long contentId, long userId, long directoryId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MoveContentFavorite", Description = "Moves a content favorite to a
new favorites directory.")>
Public Function MoveContentFavorite(contentId As long, userId As long, directoryId As long) As
void
```

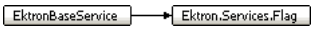
**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content item to move. |
| userId | ID of the user. |
| directoryId | ID of the favorite directory where the content is being moved. |

## Flag Class

The summary description for Taxonomy (see page 1151).

**Class Hierarchy**




**C#**

```
public class Flag : EktronBaseService;
```

**Visual Basic**

```
Public Class Flag
Inherits EktronBaseService
```

**File**

Flag.cs

**Description**

[WebService(Namespace = "http://tempuri.org/")] [WebServiceBinding(ConformsTo = WsiProfiles.None)]

# Flag Methods

The methods of the Flag class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddFlagObject (see page 1045) | Saves a new flag on an object. |
| | AssignFlagToContent (see page 1045) | Assigns a flag definition to a piece of content. |
| | GetAllFlagEntries (see page 1046) | Gets a list of flag entries for a piece of content over a given time period. |
| | GetContentFlagData (see page 1046) | Returns flagData for a piece of content for a given user. |
| | GetFlagDataById (see page 1047) | Retrieves a flag entry on a piece of content. |
| | GetFlaggedRank (see page 1047) | Retrieves a flag ranking for an object. |
| | PurgeFlagEntries (see page 1047) | Purges all flags of a particular content block over a defined date/time range. |
| | RemoveObjectObject (see page 1048) | Removes a flag entry from an object. |
| | UpdateFlagObject (see page 1048) | Updates an existing Flag (see page 1044) on an Object. |

## Flag.AddFlagObject Method

Saves a new flag on an object.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddFlagObject", Description = "adds new flagged entry")]
public long AddFlagObject(ObjectFlagData objectFlag);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddFlagObject", Description = "adds new flagged entry")>
Public Function AddFlagObject(objectFlag As ObjectFlagData) As long
```

### Parameters

| Parameters | Description |
|---|---|
| objectFlag | Object flag data to save. |

## Flag.AssignFlagToContent Method

Assigns a flag definition to a piece of content.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignFlagToContent", Description = "assign flag definition to
content")]
public void AssignFlagToContent(long contentId, long flagId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignFlagToContent", Description = "assign flag definition to
content")>
```

```
Public Function AssignFlagToContent(contentId As long, flagId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of the content to be assigned the flag. |
| flagId | ID of the flag definition. |

## Flag.GetAllFlagEntries Method

Gets a list of flag entries for a piece of content over a given time period.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllFlagEntries", Description = "Gets all flagged entries")]
public ContentFlagData[] GetAllFlagEntries(long contentId, int maxResults, DateTime startDate,
DateTime endDate, ref int totalFlags);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllFlagEntries", Description = "Gets all flagged entries")>
Public Function GetAllFlagEntries(contentId As long, maxResults As Integer, startDate As
DateTime, endDate As DateTime, ByRef totalFlags As Integer) As ContentFlagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | ID of content item for which to retrieve flags. |
| maxResults | Max number of results to retrieve. |
| startDate | Date to begin retrieving flags. |
| endDate | Date to end retrieving flags. |
| totalFlags | Returns the total number of flags found. |

**Returns**

Array of content flag data.

## Flag.GetContentFlagData Method

Returns flagData for a piece of content for a given user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentFlagData", Description = "gets flagged data using user or
visitor id")]
public ContentFlagData GetContentFlagData(long contentId, long userId, String visitorId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentFlagData", Description = "gets flagged data using user or
visitor id")>
Public Function GetContentFlagData(contentId As long, userId As long, visitorId As String) As
ContentFlagData
```

**Parameters**

| Parameters | Description |
|---|---|
| contentid | Id of content to retrieve flag for. |
| userid | Id of cms user who flagged content. |

**Returns**

ContentFlagData object.

## Flag.GetFlagDataById Method

Retrieves a flag entry on a piece of content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFlagDataById", Description = "gets content flag date using flag
id")]
public ContentFlagData GetFlagDataById(long flagId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFlagDataById", Description = "gets content flag date using flag
id")>
Public Function GetFlagDataById(flagId As long) As ContentFlagData
```

**Parameters**

| Parameters | Description |
|---|---|
| flagId | ID of the flag entry. |

**Returns**

ContentFlagData

## Flag.GetFlaggedRank Method

Retrieves a flag ranking for an object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFlaggedRank", Description = "gets content by ranking")]
public FlaggedRankData[] GetFlaggedRank(long objectId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFlaggedRank", Description = "gets content by ranking")>
Public Function GetFlaggedRank(objectId As long) As FlaggedRankData()
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | ID of the object for which to retrieve a flag ranking. |

## Flag.PurgeFlagEntries Method

Purges all flags of a particular content block over a defined date/time range.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "PurgeFlagEntries", Description = "delets matched entries")]
public void PurgeFlagEntries(long objectId, DateTime startDate, DateTime endDate);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "PurgeFlagEntries", Description = "delets matched entries")>
Public Function PurgeFlagEntries(objectId As long, startDate As DateTime, endDate As DateTime)
As void
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The ID of the content that is having its flags purged. |
| startDate | The beginning of the date range of content to be purged. |
| endDate | The end of the date range of content to be purged. |

### Flag.RemoveObjectObject Method

Removes a flag entry from an object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RemoveObjectObject", Description = "remove the flag definition")]
public void RemoveObjectObject(long flagId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveObjectObject", Description = "remove the flag definition")>
Public Function RemoveObjectObject(flagId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| flagId | ID of flag entry to remove. |

### Flag.UpdateFlagObject Method

Updates an existing Flag (see page 1044) on an Object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateFlagObject", Description = "update flag definition")]
public void UpdateFlagObject(ObjectFlagData objectFlag);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateFlagObject", Description = "update flag definition")>
Public Function UpdateFlagObject(objectFlag As ObjectFlagData) As void
```

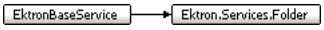
**Parameters**

| Parameters | Description |
|---|---|
| request | Request criteria defining the flag data for the object. |

## Folder Class

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

## Folder Methods

The methods of the Folder class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddFolder (see page 1049) | Adds a folder. |
| | CreateFolder (see page 1050) | Creates a folder. |
| | DeleteFolder (see page 1050) | Deletes folder by folder ID. |
| | GetChildFolders (see page 1050) | This is the overview for the GetChildFolders method overload. |
| | GetFolder (see page 1051) | This is the overview for the GetFolder method overload. |
| | GetFolderID (see page 1053) | This is the overview for the GetFolderID method overload. |
| | GetPath (see page 1054) | Finds a parent ID by using a given folder ID. |
| | RenameFolder (see page 1054) | Renames folder to specified title. |
| | UpdateFolder (see page 1054) | This is the overview for the UpdateFolder method overload. |

## Folder.AddFolder Method

Adds a folder.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddFolder", Description = "Adds a folder")]
public void AddFolder(ref FolderRequest FolderReq);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddFolder", Description = "Adds a folder")>
Public Function AddFolder(ByRef FolderReq As FolderRequest) As void
```

### Parameters

| Parameters | Description |
|---|---|
| FolderReq | Folder (see page 1048) Request Object. |

## Folder.CreateFolder Method

Creates a folder.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CreateFolder", Description = "Creates a folder with the given path")]
public bool CreateFolder(string FolderPath);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CreateFolder", Description = "Creates a folder with the given path")>
Public Function CreateFolder(FolderPath As string) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderPath | The path on the IIS server. (Required) |

## Folder.DeleteFolder Method

Deletes folder by folder ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteFolder", Description = "Deletes folder specified by folder
id")]
public bool DeleteFolder(long FolderID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteFolder", Description = "Deletes folder specified by folder
id")>
Public Function DeleteFolder(FolderID As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The ID of the folder to be deleted. |

## GetChildFolders Method

This is the overview for the GetChildFolders method overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Folder.GetChildFolders (long, bool, Ektron.Cms.Common.EkEnumeration.FolderOrderBy) (see page 1051) | Loads all the subfolders for a given folder ID. |
| | Folder.GetChildFolders (long, bool, string) (see page 1051) | Loads all the subfolders for a given folder ID. |

### Folder.GetChildFolders Method (long, bool, Ektron.Cms.Common.EkEnumeration.FolderOrderBy)

Loads all the subfolders for a given folder ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetChildFolders", Description = "Get all the subfolders for a given
folder ID")]
public FolderData[] GetChildFolders(long FolderID, bool Recursive,
Ektron.Cms.Common.EkEnumeration.FolderOrderBy OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetChildFolders", Description = "Get all the subfolders for a given
folder ID")>
Public Function GetChildFolders(FolderID As long, Recursive As bool, OrderBy As
Ektron.Cms.Common.EkEnumeration.FolderOrderBy) As FolderData()
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The folder ID containing the subfolders. |
| Recursive | True = recursive look up. False = none |
| OrderBy | An order by clause. |

### Folder.GetChildFolders Method (long, bool, string)

Loads all the subfolders for a given folder ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetChildFolders_Obsolete", Description = "This method is obsolete.
Use new GetChildFolders method with FolderOrderBy enum parameter")]
[Obsolete("Call New GetChildFolders with FolderOrderBy enum parameter")]
public FolderData[] GetChildFolders(long FolderID, bool Recursive, string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetChildFolders_Obsolete", Description = "This method is obsolete.
Use new GetChildFolders method with FolderOrderBy enum parameter")>
<Obsolete("Call New GetChildFolders with FolderOrderBy enum parameter")>
Public Function GetChildFolders(FolderID As long, Recursive As bool, OrderBy As string) As
FolderData()
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The folder ID containing the subfolders. |
| Recursive | True = recursive look up. False = none |
| OrderBy | An order by clause. |

### GetFolder Method

This is the overview for the GetFolder method overload.

**Description**

This method is overloaded.

- GetFolder (Long) - retrieves the folder's details by providing the folder ID.
- GetFolder (Long, Bool, Bool) - retrieves the folder's details by providing the folder ID. In addition, this method allows you retrieve taxonomy and flag information for a folder.

### Overload List

| | Name | Description |
|---|---|---|
| | Folder.GetFolder (long) (see page 1052) | Retrieves the folder's details. |
| | Folder.GetFolder (long, bool, bool) (see page 1052) | Retrieves the folder's details. |

## Folder.GetFolder Method (long)

Retrieves the folder's details.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolder", Description = "Retrieves the folder's details")]
public FolderData GetFolder(long FolderID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolder", Description = "Retrieves the folder's details")>
Public Function GetFolder(FolderID As long) As FolderData
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The folder's ID. |

## Folder.GetFolder Method (long, bool, bool)

Retrieves the folder's details.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderByParam", Description = "Retrieves the folder's details")]
public FolderData GetFolder(long FolderID, bool getTaxonomy, bool getFlags);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderByParam", Description = "Retrieves the folder's details")>
Public Function GetFolder(FolderID As long, getTaxonomy As bool, getFlags As bool) As
FolderData
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The folder's ID. |
| getTaxonomy | True if folder taxonomy required. |
| getFlags | True if folder flag required. |

## GetFolderID Method

This is the overview for the GetFolderID method overload.

### Description

This method is overloaded.

- GetFolderID (Long) - finds the folder ID by providing a content item's ID.
- GetFolderID (String, Long) - finds the folder ID by providing the folder's name and in the parent folder's ID.

### Overload List

| | Name | Description |
|---|---|---|
| | Folder.GetFolderID (long) (see page 1053) | Finds the folder ID by using the a content's ID. |
| | Folder.GetFolderID (string, long) (see page 1053) | Finds the folder ID by folder name and in a parent folder specified. |

## Folder.GetFolderID Method (long)

Finds the folder ID by using the a content's ID.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderID", Description = "Finds the folder ID by using the a
content's ID")]
public long GetFolderID(long ContentID);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderID", Description = "Finds the folder ID by using the a
content's ID")>
Public Function GetFolderID(ContentID As long) As long
```

### Parameters

| Parameters | Description |
|---|---|
| ContentID | The content's ID. |

## Folder.GetFolderID Method (string, long)

Finds the folder ID by folder name and in a parent folder specified.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderIDByName", Description = "Finds the folder ID by folder
name and in a parent folder specified.")]
public long GetFolderID(string FolderName, long ParentId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderIDByName", Description = "Finds the folder ID by folder
```

```
name and in a parent folder specified.")>
Public Function GetFolderID(FolderName As string, ParentId As long) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderName | Name of the folder ID. |
| ParentId | Look in a folder. |

**Returns**

Returns the folder ID. Returns 0 if it does not exist.

## Folder.GetPath Method

Finds a parent ID by using a given folder ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPath", Description = "Gets path to folder by folder ID")]
public string GetPath(long FolderID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPath", Description = "Gets path to folder by folder ID")>
Public Function GetPath(FolderID As long) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The folder's ID. |

## Folder.RenameFolder Method

Renames folder to specified title.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenameFolder", Description = "Renames the folder by ID.")]
public bool RenameFolder(long FolderID, string Title);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RenameFolder", Description = "Renames the folder by ID.")>
Public Function RenameFolder(FolderID As long, Title As string) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderID | The folder's ID. |
| Title | The folder's new title |

## UpdateFolder Method

This is the overview for the UpdateFolder method overload.

**Description**

This method is overloaded.

- UpdateFolder (FolderData) (see page 1055)
- UpdateFolder (FolderRequest) (see page 1055)

**Overload List**

| | Name | Description |
|---|---|---|
| | Folder.UpdateFolder (FolderData) (see page 1055) | Updates a folder. |
| | Folder.UpdateFolder (FolderRequest) (see page 1055) | Updates a folder. |

### Folder.UpdateFolder Method (FolderData)

Updates a folder.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateFolder[FolderData]", Description = "Updates a folder")]
public void UpdateFolder(FolderData FolderReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateFolder[FolderData]", Description = "Updates a folder")>
Public Function UpdateFolder(FolderReq As FolderData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderReq | Folder (see page 1048) Request Object. |

### Folder.UpdateFolder Method (FolderRequest)

Updates a folder.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateFolder", Description = "Updates a folder")]
public void UpdateFolder(FolderRequest FolderReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateFolder", Description = "Updates a folder")>
Public Function UpdateFolder(FolderReq As FolderRequest) As void
```

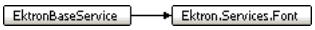
**Parameters**

| Parameters | Description |
|---|---|
| FolderReq | Folder (see page 1048) 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

## Font Methods

The methods of the Font class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddFont (see page 1056) | Adds a new font to the system. |
| | DeleteFont (see page 1056) | Deletes a font from the system. |
| | GetAllFonts (see page 1057) | Loads all of the fonts as a array of FontData. |
| | GetFont (see page 1057) | Loads the required font using the font's ID. |
| | GetFontConfigList (see page 1057) | Returns all of the font details in a drop down control. |
| | UpdateFont (see page 1058) | Updates the font. |

### 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 (see page 1055) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllFonts", Description = "")]
public FontData[] GetAllFonts();
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllFonts", Description = "")>
Public Function GetAllFonts() As FontData()
```

### Returns

Ektron.Cms.FontData()

### Remarks

User (see page 1167) login is required to perform this operation.

## Font.GetFont Method

Loads the required font using the font's ID.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFont", Description = "")]
public FontData GetFont(long Id);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFont", Description = "")>
Public Function GetFont(Id As long) As FontData
```

### Parameters

| Parameters | Description |
|---|---|
| Id | The font's numeric ID. |

### Returns

A numeric value.

## Font.GetFontConfigList Method

Returns all of the font details in a drop down control.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFontConfigList", Description = "")]
public string GetFontConfigList();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFontConfigList", Description = "")>
Public Function GetFontConfigList() As string
```

**Returns**

String

### Font.UpdateFont Method

Updates the font.

**C#**

```
public void UpdateFont(Collection data);
```

**Visual Basic**

```
Public Function UpdateFont(data As Collection) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| data | The font's data. |

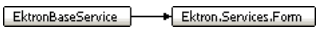
**Remarks**

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

## Form Methods

The methods of the Form class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AssignFormTask (see page 1059) | Assigns a form submission task, as specified in FormInfo (see page 560), of the SubmittedDataXml. |
| | EmailFormFieldData (see page 1059) | Sends an email, as specified in FormInfo (see page 560), of the SubmittedDataXml. |
| | GetAllForms (see page 1060) | This is the overview for the GetAllForms method overload. |
| | GetForm (see page 1061) | Retrieves information about the specified form. |
| | GetFormDataHistogram (see page 1062) | |
| | GetFormFieldData (see page 1062) | This is the overview for the GetFormFieldData method overload. |
| | GetFormFieldList (see page 1063) | Retrieves the list of form fields defined in the specified form. |
| | GetFormFieldQuestions (see page 1064) | 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 1064) | Same as GetFormById, but returns only the form's title. |
| | SubmitFormFieldData (see page 1065) | Adds form data as if it was submitted during a browser post to the web server. |
| | UpdateFormFieldData (see page 1065) | Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form. |

## Form.AssignFormTask Method

Assigns a form submission task, as specified in FormInfo (see page 560), of the SubmittedDataXml.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignFormTask", Description = "")]
public bool AssignFormTask(FormData FormInfo, string SubmittedDataXml);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignFormTask", Description = "")>
Public Function AssignFormTask(FormInfo As FormData, SubmittedDataXml As string) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| FormInfo | The properties of the form associated with the data. This information may be retrieved by calling GetFormById. Value Type: Cms.FormData (see page 1394) (Ektron.Cms.FormData (see page 1394)) |
| 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 1065) 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 1065).

## Form.EmailFormFieldData Method

Sends an email, as specified in FormInfo (see page 560), 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 (see page 1394) (Ektron.Cms.FormData (see page 1394)) |
| SubmittedDataXml | The form data submitted in XML format. Value Type: String (System.String) |

**Remarks**

This routine is called automatically when calling SubmitFormFieldData (see page 1065) 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 1065).

## GetAllForms Method

This is the overview for the GetAllForms method overload.

**Description**

This method is overloaded.

- GetAllForms () - returns all of the forms as an array of FormData.
- GetAllForms (Long, String, Int ContentAPI.FormResultType) - retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type.

**Overload List**

| | Name | Description |
|---|---|---|
| | Form.GetAllForms () (see page 1060) | Returns all of the forms as an array of FormData (see page 1394). |
| | Form.GetAllForms (long, string, int, ContentAPI.FormResultType) (see page 1061) | 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 (see page 1394).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllForms", Description = "")]
public FormData[] GetAllForms();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllForms", Description = "")>
Public Function GetAllForms() As FormData()
```

### Returns

Ektron.Cms.FormData (see page 1394)()

### Remarks

User (see page 1167) login required to perform this operation.

## Form.GetAllForms Method (long, string, int, ContentAPI.FormResultType)

Retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllForms By Id, OrderBy, LanguageID and Type", Description = "")]
public FormData[] GetAllForms(long Id, string OrderBy, int LanguageId,
ContentAPI.FormResultType Type);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllForms By Id, OrderBy, LanguageID and Type", Description = "")>
Public Function GetAllForms(Id As long, OrderBy As string, LanguageId As Integer, Type As
ContentAPI.FormResultType) As FormData()
```

### Parameters

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetForm", Description = "")]
public FormData GetForm(long Id);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetForm", Description = "")>
Public Function GetForm(Id As long) As FormData
```

### Parameters

| Parameters | Description |
|---|---|
| Id | Specifies the form. |

### Returns

Returns information about the form.

### Remarks

Does not require authentication.

## Form.GetFormDataHistogram Method

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormDataHistogram", Description = "")]
public ArrayList GetFormDataHistogram(long FormId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormDataHistogram", Description = "")>
Public Function GetFormDataHistogram(FormId As long) As ArrayList
```

### Parameters

| Parameters | Description |
|---|---|
| FormId | |

## GetFormFieldData Method

This is the overview for the GetFormFieldData method overload.

### Description

This method is overloaded.

- GetFormFieldData (Long, Long) - retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.
- GetFormFieldData (Long, String, String) - retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range.

### Overload List

| | Name | Description |
|---|---|---|
| | Form.GetFormFieldData (long, long) (see page 1062) | Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form. |
| | Form.GetFormFieldData (long, string, string) (see page 1063) | Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range. |

## Form.GetFormFieldData Method (long, long)

Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormFieldData By FormID and FormDataId", Description = "")]
public FormSubmittedData GetFormFieldData(long FormId, long FormDataId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormFieldData By FormID and FormDataId", Description = "")>
Public Function GetFormFieldData(FormId As long, FormDataId As long) As FormSubmittedData
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 1412) (Ektron.Cms.FormSubmittedData (see page 1412))

**Remarks**

Does not require authentication.

### Form.GetFormFieldData Method (long, string, string)

Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormFieldData By FormId, StartDate and EndDate", Description =
"")]
public FormSubmittedData[] GetFormFieldData(long FormId, string StartDate, string EndDate);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormFieldData By FormId, StartDate and EndDate", Description =
"")>
Public Function GetFormFieldData(FormId As long, StartDate As string, EndDate As string) As
FormSubmittedData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 1412) (Ektron.Cms.FormSubmittedData (see page 1412))

**Remarks**

Does not require authentication.

### Form.GetFormFieldList Method

Retrieves the list of form fields defined in the specified form.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "GetFormFieldList", Description = "")]
public FormFieldList GetFormFieldList(long FormId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormFieldList", Description = "")>
Public Function GetFormFieldList(FormId As long) As FormFieldList
```

**Parameters**

| Parameters | Description |
|---|---|
| FormId | Specifies the form. Value Type: Int32 (System.Int32) |

**Returns**

The list for defined fields in the given form. Cms.FormFieldList (see page 1407) (Ektron.Cms.FormFieldList (see page 1407))

## Form.GetFormFieldQuestions Method

Retrieves the table of form fields and their questions in a specified form. In a CMS form, the form question is defined by the user in the descriptive name field in the form element dialog during design time.

**C#**

```
public Hashtable GetFormFieldQuestions(long FormId);
```

**Visual Basic**

```
Public Function GetFormFieldQuestions(FormId As long) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormTitle", Description = "")]
public string GetFormTitle(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormTitle", Description = "")>
Public Function GetFormTitle(Id As long) As string
```

### Parameters

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SubmitFormFieldData", Description = "")]
public void SubmitFormFieldData(long FormId, FormSubmittedData Data, bool SendNotification,
string SubmittedDataXml);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SubmitFormFieldData", Description = "")>
Public Function SubmitFormFieldData(FormId As long, Data As FormSubmittedData,
SendNotification As bool, SubmittedDataXml As string) As void
```

### Parameters

| Parameters | Description |
|---|---|
| FormId | Specifies the form that corresponds to the data. Value Type: Int32 (System.Int32) |
| Data | The data to submit. Value Type: Cms.FormSubmittedData (see page 1412) (Ektron.Cms.FormSubmittedData (see page 1412)) |
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateFormFieldData", Description = "")]
public void UpdateFormFieldData(long FormId, long FormDataId, FormSubmittedData SubmittedData);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateFormFieldData", Description = "")>
Public Function UpdateFormFieldData(FormId As long, FormDataId As long, SubmittedData As
FormSubmittedData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| 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 (see page 1412) (Ektron.Cms.FormSubmittedData (see page 1412)) |

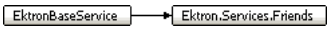
**Remarks**

Does not require authentication.

# Friends Class

Summary description for Friends

**Class Hierarchy**




**C#**

```
public class Friends : EktronBaseService;
```

**Visual Basic**

```
Public Class Friends
Inherits EktronBaseService
```

**File**

Friends.cs

## Friends Methods

The methods of the Friends class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AcceptPendingFriend (see page 1067) | Accepts a friend request from one user to another. |
| | AddFriend (see page 1067) | This is the overview for the AddFriend method overload. |
| | AddPendingFriend (see page 1068) | Adds a pending friend. Sends the other user a friend invitation. |
| | AddSelectedFriend (see page 1069) | Adds a friend to a user's selected friends list. |
| | DeleteFriend (see page 1069) | Deletes a friend from a user's friends list. |
| | DeletePendingFriendRequest (see page 1069) | This is the overview for the DeletePendingFriendRequest method overload. |
| | DeleteSelectedFriend (see page 1071) | Delete's a friend from a user's selected friends list. |
| | GetFriends (see page 1071) | This is the overview for the GetFriends method overload. |
| | GetFriendsDirectory (see page 1073) | Retrieves friends directory. |
| | GetFriendStatus (see page 1074) | Get another user's friend status in relation to you. For instance, user "A" may be a friend to User (see page 1167) "B". User (see page 1167) "C" may be a pending friend to User (see page 1167) "D". |
| | GetInvitedUsers (see page 1074) | Gets a list of sent friend invitations for a given user. |
| | GetPendingFriends (see page 1075) | Gets a list of users that have sent another user friend invitations. This only returns users who's invitations have not yet been accepted and are still pending. |
| | GetSelectedFriends (see page 1075) | Get a user's selected friends list. |
| | GetSentFriendInvitations (see page 1076) | Gets a list of pending Friend invitations sent out by a user. |
| | InviteUsers (see page 1076) | Send a friend invitation a list of users and\or email addresses. |
| | IsFriend (see page 1077) | Returns true if FriendUserId and UserID are friends. |

| | | |
|---|---|---|
| | IsSelectedFriend (see page 1077) | Check if a user is a selected friend of another user. |
| | MoveFriend (see page 1077) | Moves a friend into another friend directory. |

## Friends.AcceptPendingFriend Method

Accepts a friend request from one user to another.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AcceptPendingFriend", Description = "Accepts a friend request from
one user to another.")]
public void AcceptPendingFriend(long fromUserId, long toUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AcceptPendingFriend", Description = "Accepts a friend request from
one user to another.")>
Public Function AcceptPendingFriend(fromUserId As long, toUserId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| fromUserId | ID of user who sent friend request. |
| toUserId | ID of user who received friend request and is accepting it. |

## AddFriend Method

This is the overview for the AddFriend method overload.

**Description**

This method is overloaded.

- AddFriend (Long, Long) - adds a user to another user's friends list based on the ID of the user who is being added as a friend and the ID of the user adding the friend.
- AddFriend (Long, Long, Long) - adds a user to a specified directory of another user's friends list based on the ID of the user who is being added as a friend, the ID of the user adding the friend and the ID of the directory where the user is being added.

**Overload List**

| | Name | Description |
|---|---|---|
| | Friends.AddFriend (long, long) (see page 1067) | Adds a user to another user's friends list. |
| | Friends.AddFriend (long, long, long) (see page 1068) | Adds a user to another user's friends list in a specific sub directory. |

## Friends.AddFriend Method (long, long)

Adds a user to another user's friends list.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddFriend", Description = "Adds a user to another user's friends
```

```
list.")]
public void AddFriend(long friendUserId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddFriend", Description = "Adds a user to another user's friends
list.")>
Public Function AddFriend(friendUserId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | The user ID to add to users friends list. |
| userId | The ID of the user adding friend. |

### Friends.AddFriend Method (long, long, long)

Adds a user to another user's friends list in a specific sub directory.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddFriend for directory", Description = "Adds a user to aother
user's friends list in a specific sub directory.")]
public void AddFriend(long friendUserId, long userId, long directoryId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddFriend for directory", Description = "Adds a user to aother
user's friends list in a specific sub directory.")>
Public Function AddFriend(friendUserId As long, userId As long, directoryId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | The user ID of friend to add to users friends list. |
| userId | The ID of user adding friend. |
| directoryId | The ID of directory to place friend into. |

### Friends.AddPendingFriend Method

Adds a pending friend. Sends the other user a friend invitation.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddPendingFriend", Description = "Adds a pending friend. Sends the
other user a friend invitation.")]
public void AddPendingFriend(long friendUserId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddPendingFriend", Description = "Adds a pending friend. Sends the
other user a friend invitation.")>
Public Function AddPendingFriend(friendUserId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | The user receiving a friend request. |

| | |
|---|---|
| userId | The user sending a friend request. |

### Friends.AddSelectedFriend Method

Adds a friend to a user's selected friends list.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddSelectedFriend", Description = "Adds a friend to a user's
selected friends list.")]
public void AddSelectedFriend(long selectedFriendUserId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddSelectedFriend", Description = "Adds a friend to a user's
selected friends list.")>
Public Function AddSelectedFriend(selectedFriendUserId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| selectedFriendUserId | The ID of the friend to be added to the selected list. |
| userId | The ID of the user adding the selected friend. |

### Friends.DeleteFriend Method

Deletes a friend from a user's friends list.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteFriend", Description = "Deletes a friend from a user's friends
list.")]
public void DeleteFriend(long friendUserId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteFriend", Description = "Deletes a friend from a user's friends
list.")>
Public Function DeleteFriend(friendUserId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | User (see page 1167) ID of the friend to delete from a user's friends list. |
| userId | The ID of a user deleting the friend. |

### DeletePendingFriendRequest Method

This is the overview for the DeletePendingFriendRequest method overload.

**Description**

This method is overloaded.

- DeletePendingFriendRequest (Long, Long) - deletes a pending friend request by providing the ID of the user who sent the original friend request and the ID of the user who received the request.
- DeletePendingFriendRequest (Long, String) - Ektron.Services.Friends.DeletePendingFriendRequest@int@stringdeletes a pending friend request by providing the ID of the user who sent the original friend request and the email address of the user who received the request.

### Overload List

| | Name | Description |
|---|---|---|
| | Friends.DeletePendingFriendRequest (long, long) (see page 1070) | Deletes a pending friend request. |
| | Friends.DeletePendingFriendRequest (long, string) (see page 1070) | Deletes a pending friend request. |

## Friends.DeletePendingFriendRequest Method (long, long)

Deletes a pending friend request.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeletePendingFriendRequest", Description = "Deletes a pending friend
request.")]
public void DeletePendingFriendRequest(long fromUserId, long toUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeletePendingFriendRequest", Description = "Deletes a pending friend
request.")>
Public Function DeletePendingFriendRequest(fromUserId As long, toUserId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| fromUserId | The ID of the user who sent original friend request. |
| toUserId | The ID of the user who received friend request. |

## Friends.DeletePendingFriendRequest Method (long, string)

Deletes a pending friend request.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeletePendingFriendRequest by email address", Description = "Deletes
a pending friend request.")]
public void DeletePendingFriendRequest(long fromUserId, string toEmailAddress);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeletePendingFriendRequest by email address", Description = "Deletes
a pending friend request.")>
Public Function DeletePendingFriendRequest(fromUserId As long, toEmailAddress As string) As
void
```

**Parameters**

| Parameters | Description |
|---|---|
| fromUserId | The ID of the user who sent original friend request. |
| toEmailAddress | The Email address of the user who received friend request. |

## Friends.DeleteSelectedFriend Method

Delete's a friend from a user's selected friends list.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteSelectedFriend", Description = "Delete's a friend from a
user's selected friends list.")]
public void DeleteSelectedFriend(long selectedFriendUserId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteSelectedFriend", Description = "Delete's a friend from a
user's selected friends list.")>
Public Function DeleteSelectedFriend(selectedFriendUserId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| selectedFriendUserId | User (see page 1167) ID of the friend to delete from user's selected friends list. |
| userId | ID of the user who is deleting the friend. |

## GetFriends Method

This is the overview for the GetFriends method overload.

**Description**

This method is overloaded.

- GetFriends (Long) - returns a list of friends for a user by providing the user's ID.
- GetFriends (Long, PagingInfo) - returns a list of friends for a user by providing the user's ID. This method allows for paging of the list.
- GetFriends (Long, PagingInfo, Long) - returns a user's list of friends in a specific subdirectory by providing the user's ID and the ID of the sub directory. This method allows for paging of the list.
- GetFriends (Long, Long) - returns a user's list of friends in a specific subdirectory by providing the user's ID and the ID of the sub directory.

**Overload List**

| | Name | Description |
|---|---|---|
| | Friends.GetFriends (long) (see page 1072) | Returns a list of friends for a user. |
| | Friends.GetFriends (long, long) (see page 1073) | Returns a user's list of friends in a specific subdirectory. |
| | Friends.GetFriends (long, PagingInfo) (see page 1072) | Returns a list of friends for a user. |
| | Friends.GetFriends (long, PagingInfo, long) (see page 1072) | Returns a user's list of friends in a specific subdirectory. |

### Friends.GetFriends Method (long)

Returns a list of friends for a user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriends", Description = "Returns a list of friends for a user.")]
public DirectoryUserData[] GetFriends(long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriends", Description = "Returns a list of friends for a user.")>
Public Function GetFriends(userId As long) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of user to retrieve friends for. |

**Returns**

An array of DirectoryUserData.

### Friends.GetFriends Method (long, PagingInfo)

Returns a list of friends for a user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriends with paging info", Description = "Returns a list of
friends for a user.")]
public DirectoryUserData[] GetFriends(long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriends with paging info", Description = "Returns a list of
friends for a user.")>
Public Function GetFriends(userId As long, pagingInfo As PagingInfo) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of user to retrieve friends for. |
| pagingInfo | Supply paging information to control retrieval. |

**Returns**

An array of DirectoryUserData.

### Friends.GetFriends Method (long, PagingInfo, long)

Returns a user's list of friends in a specific subdirectory.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriends with paging info for directory", Description = "Returns a
user's list of friends in a specific subdirectory.")]
public DirectoryUserData[] GetFriends(long userId, PagingInfo pagingInfo, long directoryID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriends with paging info for directory", Description = "Returns a
user's list of friends in a specific subdirectory.")>
Public Function GetFriends(userId As long, pagingInfo As PagingInfo, directoryID As long) As
DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of a user to retrieve friends for. |
| pagingInfo | Supply paging information to control retrieval. |
| directoryID | The ID of directory to retrieve friends from. |

**Returns**

An array of DirectoryUserData.

## Friends.GetFriends Method (long, long)

Returns a user's list of friends in a specific subdirectory.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriends for directory", Description = "Returns a user's list of
friends in a specific subdirectory.")]
public DirectoryUserData[] GetFriends(long userId, long directoryID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriends for directory", Description = "Returns a user's list of
friends in a specific subdirectory.")>
Public Function GetFriends(userId As long, directoryID As long) As DirectoryUserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | ID of a user to retrieve friends for. |
| directoryID | ID of directory to retrieve friends from. |

**Returns**

An array of DirectoryUserData.

## Friends.GetFriendsDirectory Method

Retrieves friends directory.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriendsDirectory", Description = "Retrieves friends directory.")]
public DirectoryAdvancedUserData GetFriendsDirectory(long userId, long directoryId, PagingInfo
pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriendsDirectory", Description = "Retrieves friends directory.")>
Public Function GetFriendsDirectory(userId As long, directoryId As long, pagingInfo As
```

```
PagingInfo) As DirectoryAdvancedUserData
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user who is retrieving a friends directory. |
| directoryId | The ID of friends directory to retrieve. |
| pagingInfo | Supply paging information to control retrieval. |

**Returns**

DirectoryAdvancedUserData

### Friends.GetFriendStatus Method

Get another user's friend status in relation to you. For instance, user "A" may be a friend to User (see page 1167) "B". User (see page 1167) "C" may be a pending friend to User (see page 1167) "D".

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFriendStatus", Description = "Get another user's friend status in
relation to you.  For instance, user A may be  a friend to User B.  User C may be a pending
friend to User D.")]
public EkEnumeration.FriendStatus GetFriendStatus(long friendUserId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFriendStatus", Description = "Get another user's friend status in
relation to you.  For instance, user A may be  a friend to User B.  User C may be a pending
friend to User D.")>
Public Function GetFriendStatus(friendUserId As long, userId As long) As
EkEnumeration.FriendStatus
```

**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | User (see page 1167) to get status. |
| userId | User (see page 1167) seeking status. |

**Returns**

FriendStatus enumeration.

### Friends.GetInvitedUsers Method

Gets a list of sent friend invitations for a given user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetInvitedUsers", Description = "Gets a list of sent friend
invitations for a given user.")]
public DirectoryUserData[] GetInvitedUsers(long userId, PagingInfo pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetInvitedUsers", Description = "Gets a list of sent friend
invitations for a given user.")>
Public Function GetInvitedUsers(userId As long, pagingInfo As PagingInfo) As
DirectoryUserData()
```

### Parameters

| Parameters | Description |
|---|---|
| userId | ID of user who sent friend invitations. |
| pagingInfo | supply paging information to control retrieval. |

### Returns

An array of DirectoryUserData

## Friends.GetPendingFriends Method

Gets a list of users that have sent another user friend invitations. This only returns users who's invitations have not yet been accepted and are still pending.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPendingFriends", Description = "Gets a list of users that have
sent another user friend invitations. This only returns users who's invitations have not yet
been accepted and are still pending.")]
public DirectoryUserData[] GetPendingFriends(long userId, PagingInfo pagingInfo);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPendingFriends", Description = "Gets a list of users that have
sent another user friend invitations. This only returns users who's invitations have not yet
been accepted and are still pending.")>
Public Function GetPendingFriends(userId As long, pagingInfo As PagingInfo) As
DirectoryUserData()
```

### Parameters

| Parameters | Description |
|---|---|
| userId | ID of user who has received invitations. |
| pagingInfo | Supply paging information to control retrieval. |

### Returns

An array of DirectoryUserData.

## Friends.GetSelectedFriends Method

Get a user's selected friends list.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSelectedFriends", Description = "Get a list of user's selected
friends.")]
public DirectoryUserData[] GetSelectedFriends(long userId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSelectedFriends", Description = "Get a list of user's selected
friends.")>
Public Function GetSelectedFriends(userId As long) As DirectoryUserData()
```

### Parameters

| Parameters | Description |
|---|---|
| userId | ID of user to retrieve selected friends for. |

### Returns

An array of DirectoryUserData

## Friends.GetSentFriendInvitations Method

Gets a list of pending Friend invitations sent out by a user.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSentFriendInvitations", Description = "Gets a list of pending
Friend invitations sent out by a user.")]
public FriendInvitation[] GetSentFriendInvitations(long userId, PagingInfo pagingInfo);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSentFriendInvitations", Description = "Gets a list of pending
Friend invitations sent out by a user.")>
Public Function GetSentFriendInvitations(userId As long, pagingInfo As PagingInfo) As
FriendInvitation()
```

### Parameters

| Parameters | Description |
|---|---|
| userId | The ID of user who sent the invitations. |
| pagingInfo | Supply paging information to control retrieval. |

### Returns

An array of FriendInvitations.

## Friends.InviteUsers Method

Send a friend invitation a list of users and\or email addresses.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "InviteUsers", Description = "Send a friend invitation a list of
users and\\or email addresses.")]
public Collection<Invitation> InviteUsers(InvitationSendRequestData invitationData);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "InviteUsers", Description = "Send a friend invitation a list of
users and\\or email addresses.")>
Public Function InviteUsers(invitationData As InvitationSendRequestData) As
Collection<Invitation>
```

### Parameters

| Parameters | Description |
|---|---|
| invitationData | The invitation request data detailing invitation information. |

### Returns

A collection of invitation objects that have been sent out.

## Friends.IsFriend Method

Returns true if FriendUserId and UserID are friends.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsFriend", Description = "Returns true if the FriendUserId and
UserID are friends.")]
public bool IsFriend(long friendUserId, long userId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsFriend", Description = "Returns true if the FriendUserId and
UserID are friends.")>
Public Function IsFriend(friendUserId As long, userId As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| friendUserId | First user ID. |
| userId | Second user ID. |

### Returns

Returns true if FriendUserId and UserID are friends, else false.

## Friends.IsSelectedFriend Method

Check if a user is a selected friend of another user.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsSelectedFriend", Description = "Check if a user is a selected
friend of another user.")]
public bool IsSelectedFriend(long friendUserId, long userId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsSelectedFriend", Description = "Check if a user is a selected
friend of another user.")>
Public Function IsSelectedFriend(friendUserId As long, userId As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| friendUserId | User (see page 1167) ID to check for in selected friends list. |
| userId | ID of user who is checking for selected friend. |

### Returns

True if friendUserId is in the selected friends list of UserId, else false.

## Friends.MoveFriend Method

Moves a friend into another friend directory.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MoveFriend", Description = "Moves a friend into another friend
directory.")]
public void MoveFriend(long friendUserId, long userId, long directoryId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MoveFriend", Description = "Moves a friend into another friend
directory.")>
Public Function MoveFriend(friendUserId As long, userId As long, directoryId As long) As void
```

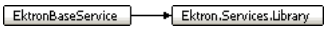
**Parameters**

| Parameters | Description |
|---|---|
| friendUserId | ID of friend to move. |
| userId | ID of user moving the friend. |
| directoryId | ID of the directory to move the friend into. |

# 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

## Library Methods

The methods of the Library class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddLoadBalanceItem (see page 1079) | Adds a load balance path for files and images. |
| | DeleteLibraryItem (see page 1079) | Deletes the library item by extension. |
| | DeleteLibraryItemById (see page 1079) | Deletes the library items from the system. |
| | DeleteLoadBalanceItem (see page 1080) | Deletes the load balance item from the system. |
| | GetAllChildLibItems (see page 1080) | Loads all of the library items by type. |

| | | |
|---|---|---|
| | GetAllLoadBalancePaths (see page 1081) | This is the overview for the GetAllLoadBalancePaths method overload. |
| | GetLibraryItem (see page 1082) | This is the overview for the GetLibraryItem method overload. |
| | GetLibrarySettings (see page 1083) | Loads the library settings. |
| | GetLibraryTypes (see page 1084) | Loads the library types. |
| | GetLoadBalancePath (see page 1084) | Gets the load balance path by ID. |
| | InternalLibrarySearch (see page 1085) | Searches library items. |
| | UpdateLibrary (see page 1085) | Updates a library item. |
| | UpdateLibrarySettings (see page 1085) | Update the library settings. |
| | UpdateLoadBalanceSettings (see page 1086) | Updates the load balance settings. |
| | UpdateQlinkTemplates (see page 1086) | Updates the Quicklink template. |

## 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 1078) 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 1078) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteLoadBalanceItem", Description = "")]
public void DeleteLoadBalanceItem(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteLoadBalanceItem", Description = "")>
Public Function DeleteLoadBalanceItem(Id As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllChildLibItems", Description = "Loads all of the library items
by type")]
public LibraryData[] GetAllChildLibItems(string Type, long ParentId, string OrderBy, int
currentPageNum, int pageSize, ref int totalPages);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllChildLibItems", Description = "Loads all of the library items
by type")>
Public Function GetAllChildLibItems(Type As string, ParentId As long, OrderBy As string,
currentPageNum As Integer, pageSize As Integer, ByRef totalPages As Integer) As LibraryData()
```

### Parameters

| Parameters | Description |
|---|---|
| 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 (see page 1167) login required to perform this operation.

## GetAllLoadBalancePaths Method

This is the overview for the GetAllLoadBalancePaths method overload.

### Description

This method is overloaded.

- GetAllLoadBalancePaths (Long, String) - returns an array of LoadBalanceData for either images or files.
- GetAllLoadBalancePaths (String) - returns all the load balance paths as a array of LoadBalanceData.

### Overload List

| | Name | Description |
|---|---|---|
| | Library.GetAllLoadBalancePaths (long, string) (see page 1081) | Returns an array of LoadBalanceData. |
| | Library.GetAllLoadBalancePaths (string) (see page 1082) | Returns all the load balance paths as a array of LoadBalanceData. |

## Library.GetAllLoadBalancePaths Method (long, string)

Returns an array of LoadBalanceData.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllLoadBalancePaths By Id and Type", Description = "")]
public LoadBalanceData[] GetAllLoadBalancePaths(long Id, string Type);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllLoadBalancePaths By Id and Type", Description = "")>
Public Function GetAllLoadBalancePaths(Id As long, Type As string) As LoadBalanceData()
```

### Parameters

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllLoadBalancePaths By Type", Description = "")]
public LoadBalanceData[] GetAllLoadBalancePaths(string Type);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllLoadBalancePaths By Type", Description = "")>
Public Function GetAllLoadBalancePaths(Type As string) As LoadBalanceData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Type | Type (images or files). |

**Returns**

Ektron.Cms.LoadBalanceData()

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

### GetLibraryItem Method

This is the overview for the GetLibraryItem method overload.

**Description**

This method is overloaded.

- GetLibraryItem (Long) - loads the library item data based on the given library ID.
- GetLibraryItem (Long, Long) - loads the library item data based on the given library ID and parent folder ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Library.GetLibraryItem (long) (see page 1083) | Loads the library item data. |
| | Library.GetLibraryItem (long, long) (see page 1083) | Loads the library item's data. |

### Library.GetLibraryItem Method (long)

Loads the library item data.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibraryItem By Id", Description = "")]
public LibraryData GetLibraryItem(long LibID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibraryItem By Id", Description = "")>
Public Function GetLibraryItem(LibID As long) As LibraryData
```

**Parameters**

| Parameters | Description |
|---|---|
| LibID | The library ID. |

**Returns**

Ektron.Cms.LibraryData

### Library.GetLibraryItem Method (long, long)

Loads the library item's data.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibraryItem By Id And ParentID", Description = "")]
public LibraryData GetLibraryItem(long Id, long ParentId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibraryItem By Id And ParentID", Description = "")>
Public Function GetLibraryItem(Id As long, ParentId As long) As LibraryData
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibrarySettings", Description = "")]
```

```
public LibraryConfigData GetLibrarySettings(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibrarySettings", Description = "")>
Public Function GetLibrarySettings(Id As long) As LibraryConfigData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The folder's ID. |

**Returns**

Ektron.Cms.LibraryConfigData

**Remarks**

Login (see page 337) required to perform this operation.

### Library.GetLibraryTypes Method

Loads the library types.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibraryTypes", Description = "")]
public LibraryTypeData[] GetLibraryTypes(long Id, string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibraryTypes", Description = "")>
Public Function GetLibraryTypes(Id As long, OrderBy As string) As LibraryTypeData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The folder's ID. |
| OrderBy | Order by clause. |

**Returns**

Object

**Remarks**

Library (see page 1078) read permission is required to perform this operation.

### Library.GetLoadBalancePath Method

Gets the load balance path by ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "", Description = "")]
public LoadBalanceData GetLoadBalancePath(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "", Description = "")>
```

```
Public Function GetLoadBalancePath(Id As long) As LoadBalanceData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The 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 (see page 864) criteria. |
| sar | Search (see page 864) asset criteria. |

**Returns**

Ektron.Cms.SearchResultData()

**Remarks**

User (see page 1167) 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 1078) 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 1078) 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 (see page 1167) 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.

## MessageBoard Class

Summary description for MessageBoard

**Class Hierarchy**

**C#**

```
public class MessageBoard : EktronBaseService;
```

**Visual Basic**

```
Public Class MessageBoard
Inherits EktronBaseService
```

**File**

MessageBoard.cs

## MessageBoard Methods

The methods of the MessageBoard class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddMessageBoardEntry (see page 1087) | Adds a Message Board Entry |
| | ApproveMessage (see page 1088) | This is the overview for the ApproveMessage method overload. |
| | DeleteMessageBoardEntries (see page 1090) | This is the overview for the DeleteMessageBoardEntries method overload. |
| | DeleteMessageBoardEntriesByUser (see page 1092) | This is the overview for the DeleteMessageBoardEntriesByUser method overload. |
| | DeleteMessageBoardEntry (see page 1094) | This is the overview for the DeleteMessageBoardEntry method overload. |
| | GetAllMessageBoardEntries (see page 1096) | This is the overview for the GetAllMessageBoardEntries method overload. |
| | GetAllMessageBoardEntriesByUser (see page 1098) | This is the overview for the GetAllMessageBoardEntriesByUser method overload. |
| | GetMessageBoardEntry (see page 1101) | This is the overview for the GetMessageBoardEntry method overload. |
| | IsUserMessageBoardAdmin (see page 1103) | This function determines if the user has administrative privileges for the message board. |
| | UpdateMessageText (see page 1103) | This is the overview for the UpdateMessageText method overload. |

### MessageBoard.AddMessageBoardEntry Method

Adds a Message Board Entry

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddMessageBoardEntry", Description = "Adds a Message Board Entry.")]
public MessageBoardData AddMessageBoardEntry(long objectId,
EkEnumeration.MessageBoardObjectType objectType, string messageText, bool isModerated, long
callerUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddMessageBoardEntry", Description = "Adds a Message Board Entry.")>
Public Function AddMessageBoardEntry(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, messageText As string, isModerated As bool, callerUserId
As long) As MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |
| messageText | |
| isModerated | The moderation switch. |

| | |
|---|---|
| callerUserId | The userid of the member requesting data. |

## ApproveMessage Method

This is the overview for the ApproveMessage method overload.

### Description

This method is overloaded.

- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType) - change the flag of the message from pending to approved by providing the message ID, object ID and type.
- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - change the flag of the message from pending to approved by providing the message ID, object ID and type, and the ID of the user who changing the flag.
- ApproveMessage (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Bool) - change the flag of the message from pending to approved by providing the message ID, object ID and type, the ID of the user who changing the flag and whether the message board is moderated.

### Overload List

| | Name | Description |
|---|---|---|
| | MessageBoard.ApproveMessage (long, long, EkEnumeration.MessageBoardObjectType) (see page 1088) | This function changes the flag of the message from pending to approve. |
| | MessageBoard.ApproveMessage (long, long, EkEnumeration.MessageBoardObjectType, long) (see page 1089) | This function changes the flag of the message from pending to approve. |
| | MessageBoard.ApproveMessage (long, long, EkEnumeration.MessageBoardObjectType, long, bool) (see page 1089) | This function changes the flag of the message from pending to approve. |

## MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType)

This function changes the flag of the message from pending to approve.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ApproveMessage", Description = "Updates the Text of a message.")]
public MessageBoardData ApproveMessage(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ApproveMessage", Description = "Updates the Text of a message.")>
Public Function ApproveMessage(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType) As MessageBoardData
```

### Parameters

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object. |

| | |
|---|---|
| objectType | The type of the object. |

**Returns**

Returns the updated row.

### MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType, long)

This function changes the flag of the message from pending to approve.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ApproveMessage with caller user id", Description = "Updates the Text
of a message.")]
public MessageBoardData ApproveMessage(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ApproveMessage with caller user id", Description = "Updates the Text
of a message.")>
Public Function ApproveMessage(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long) As MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object. |
| objectType | The type of the object. |
| callerUserId | The userid of the member doing the update. |

**Returns**

Returns the updated row.

### MessageBoard.ApproveMessage Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool)

This function changes the flag of the message from pending to approve.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ApproveMessage with caller user id and moderated flag", Description
= "Updates the Text of a message.")]
public MessageBoardData ApproveMessage(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId, bool isModerated);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ApproveMessage with caller user id and moderated flag", Description
= "Updates the Text of a message.")>
Public Function ApproveMessage(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long, isModerated As bool) As
MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |

| | |
|---|---|
| objectId | The id of the object. |
| objectType | The type of the object. |
| callerUserId | The userid of the member doing the update. |
| isModerated | The moderation switch. |

**Returns**

Returns the updated row.

## DeleteMessageBoardEntries Method

This is the overview for the DeleteMessageBoardEntries method overload.

**Description**

This method is overloaded.

- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType) - deletes all the message board records for a particular object based on the object ID and type.
- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long)Ektron.Services.MessageBoard.DeleteMessageBoardEntries@int@EkEnumeration.MessageBoardObjectType@int - deletes all the message board records for a particular object based on the object ID and type, and the user ID of the member deleting the messages.
- DeleteMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, Long, Bool) - deletes all the message board records for a particular object based on the object ID and type, the user ID of the member deleting the messages, and whether the message board is moderated.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.DeleteMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType) (see page 1090) | Deletes all the message board records for a particular object. |
| | MessageBoard.DeleteMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, long) (see page 1091) | Deletes all the message board records for a particular object. |
| | MessageBoard.DeleteMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, long, bool) (see page 1091) | Deletes all the message board records for a particular object. |

## MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType)

Deletes all the message board records for a particular object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntries", Description = "Deletes all the message
board records for a particular object.")]
public bool DeleteMessageBoardEntries(long objectId, EkEnumeration.MessageBoardObjectType
objectType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntries", Description = "Deletes all the message
board records for a particular object.")>
Public Function DeleteMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |

**Returns**

Returns true when all the messages are deleted successfully

### MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, long)

Deletes all the message board records for a particular object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntries with caller user id", Description =
"Deletes all the message board records for a particular object.")]
public bool DeleteMessageBoardEntries(long objectId, EkEnumeration.MessageBoardObjectType
objectType, long callerUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntries with caller user id", Description =
"Deletes all the message board records for a particular object.")>
Public Function DeleteMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The userid of the member deleting the messages. |

**Returns**

Returns true when all the messages are deleted successfully

### MessageBoard.DeleteMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, long, bool)

Deletes all the message board records for a particular object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntries with caller user id and moderated flag",
Description = "Deletes all the message board records for a particular object.")]
public bool DeleteMessageBoardEntries(long objectId, EkEnumeration.MessageBoardObjectType
objectType, long callerUserId, bool isModerated);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntries with caller user id and moderated flag",
```

```
Description = "Deletes all the message board records for a particular object.")>
Public Function DeleteMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long, isModerated As bool) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The userid of the member deleting the messages. |
| isModerated | The moderation switch. |

**Returns**

Returns true when all the messages are deleted successfully

## DeleteMessageBoardEntriesByUser Method

This is the overview for the DeleteMessageBoardEntriesByUser method overload.

**Description**

This method is overloaded.

- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long) - deletes all of a user's messages on a given message board based on the object ID and type, and the user's ID.
- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Bool) - deletes all of a user's messages on a given message board based on the object ID and type, the ID of the user whose messages are being removed and the ID of the user removing the messages.
- DeleteMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, Long, Bool) - deletes all of a user's messages on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and whether the message board is moderated.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.DeleteMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long) (see page 1092) | Deletes all the message board records for a particular User (see page 1167). |
| | MessageBoard.DeleteMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, long) (see page 1093) | Deletes all the message board records for a particular User (see page 1167). |
| | MessageBoard.DeleteMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, long, bool) (see page 1093) | Deletes all the message board records for a particular User (see page 1167). |

## MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long)

Deletes all the message board records for a particular User (see page 1167).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntriesByUser", Description = "Deletes all the
message board records for a particular User.")]
public bool DeleteMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntriesByUser", Description = "Deletes all the
message board records for a particular User.")>
Public Function DeleteMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |
| userId | The user ID whose messages to you want to delete. |

## MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, long)

Deletes all the message board records for a particular User (see page 1167).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntriesByUser with caller user id", Description =
"Deletes all the message board records for a particular User.")]
public bool DeleteMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, long callerUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntriesByUser with caller user id", Description =
"Deletes all the message board records for a particular User.")>
Public Function DeleteMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, callerUserId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |
| userId | The user ID whose messages to you want to delete. |
| callerUserId | The userid of the member deleting the messages. |

## MessageBoard.DeleteMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, long, bool)

Deletes all the message board records for a particular User (see page 1167).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntriesByUser with caller user id and moderated
flag", Description = "Deletes all the message board records for a particular User.")]
public bool DeleteMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, long callerUserId, bool
isModerated);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntriesByUser with caller user id and moderated
```

```
flag", Description = "Deletes all the message board records for a particular User.")>
Public Function DeleteMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, callerUserId As long, isModerated As
bool) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |
| userId | The user ID whose messages to you want to delete. |
| callerUserId | The userid of the member deleting the messages. |
| isModerated | The moderation switch. |

## DeleteMessageBoardEntry Method

This is the overview for the DeleteMessageBoardEntry method overload.

**Description**

This method is overloaded.

- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) - deletes single message board record based on the message ID, object ID and object type.
- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - deletes single message board record based on the message ID, object ID, object type and user ID requesting the data.
- DeleteMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Bool) - deletes single message board record based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.DeleteMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType) (see page 1094) | Deletes single message board record. |
| | MessageBoard.DeleteMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType, long) (see page 1095) | Deletes single message board record. |
| | MessageBoard.DeleteMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType, long, bool) (see page 1095) | Deletes single message board record. |

## MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType)

Deletes single message board record.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntry", Description = "Deletes single message
board record.")]
public bool DeleteMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntry", Description = "Deletes single message
board record.")>
Public Function DeleteMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |

**Returns**

Returns true when the message is deleted successfully.

### MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long)

Deletes single message board record.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntry with caller user id", Description = "Deletes
single message board record.")]
public bool DeleteMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntry with caller user id", Description = "Deletes
single message board record.")>
Public Function DeleteMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The userid of the member deleting the message. |

**Returns**

Returns true when the message is deleted successfully.

### MessageBoard.DeleteMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool)

Deletes single message board record.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMessageBoardEntry with caller user id and moderated flag",
Description = "Deletes single message board record.")]
```

```
public bool DeleteMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId, bool isModerated);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMessageBoardEntry with caller user id and moderated flag",
Description = "Deletes single message board record.")>
Public Function DeleteMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long, isModerated As bool) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The userid of the member deleting the message. |
| isModerated | The moderation switch. |

**Returns**

Returns true when the message is deleted successfully.

## GetAllMessageBoardEntries Method

This is the overview for the GetAllMessageBoardEntries method overload.

**Description**

This method is overloaded.

- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo) - gets all messages for a particular object based on the object ID, object type and paging information.
- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long) - gets all messages for a particular object based on the object ID, object type, ID of the user requesting the list and paging information.
- GetAllMessageBoardEntries (Long, EkEnumeration.MessageBoardObjectType, PagingInfo, Long, Bool) - gets all messages for a particular object based on the object ID, object type, ID of the user requesting the list, paging information and the moderation switch.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.GetAllMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, PagingInfo) (see page 1097) | Gets all messages for a particular object. |
| | MessageBoard.GetAllMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long) (see page 1097) | Gets all messages for a particular object. |
| | MessageBoard.GetAllMessageBoardEntries (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long, bool) (see page 1098) | Gets all messages for a particular object. |

### MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo)

Gets all messages for a particular object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntries", Description = "Gets all messages for a
particular object. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntries(long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntries", Description = "Gets all messages for a
particular object. ")>
Public Function GetAllMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, paging As PagingInfo) As Collection<MessageBoardData>
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object. |
| objectType | The type of the object. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

### MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long)

Gets all messages for a particular object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntries with caller user id", Description = "Gets
all messages for a particular object. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntries(long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging, long callerUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntries with caller user id", Description = "Gets
all messages for a particular object. ")>
Public Function GetAllMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, paging As PagingInfo, callerUserId As long) As
Collection<MessageBoardData>
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object. |
| objectType | The type of the object. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |
| callerUserId | The userid of the member requesting data. |

### Returns

Returns a collection of message board data objects for each record retrieved from the database.

## MessageBoard.GetAllMessageBoardEntries Method (long, EkEnumeration.MessageBoardObjectType, PagingInfo, long, bool)

Gets all messages for a particular object.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntries with caller user id and moderated flag",
Description = "Gets all messages for a particular object. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntries(long objectId,
EkEnumeration.MessageBoardObjectType objectType, PagingInfo paging, long callerUserId, bool
isModerated);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntries with caller user id and moderated flag",
Description = "Gets all messages for a particular object. ")>
Public Function GetAllMessageBoardEntries(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, paging As PagingInfo, callerUserId As long, isModerated
As bool) As Collection<MessageBoardData>
```

### Parameters

| Parameters | Description |
|---|---|
| objectId | The id of the object. |
| objectType | The type of the object. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |
| callerUserId | The userid of the member requesting data. |
| isModerated | The moderation switch. |

### Returns

Returns a collection of message board data objects for each record retrieved from the database.

## GetAllMessageBoardEntriesByUser Method

This is the overview for the GetAllMessageBoardEntriesByUser method overload.

### Description

This method is overloaded.

- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo) - gets all messages entered by a user on a given message board based on the object ID and type, the user's ID and paging information.
- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long) - gets all messages entered by a user on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and paging information.
- GetAllMessageBoardEntriesByUser (Long, EkEnumeration.MessageBoardObjectType, Long, PagingInfo, Long, Bool) - gets all messages entered by a user on a given message board based on the object ID and type, the ID of the user whose messages are being removed, the ID of the user removing the messages and whether the message board is moderated.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.GetAllMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo) (see page 1099) | Gets all messages entered by a user. |
| | MessageBoard.GetAllMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long) (see page 1099) | Gets all messages entered by a user. |
| | MessageBoard.GetAllMessageBoardEntriesByUser (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long, bool) (see page 1100) | Gets all messages entered by a user. |

## MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo)

Gets all messages entered by a user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntriesByUser", Description = "Gets all messages
for a particular user. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, PagingInfo paging);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntriesByUser", Description = "Gets all messages
for a particular user. ")>
Public Function GetAllMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, paging As PagingInfo) As
Collection<MessageBoardData>
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object. |
| objectType | The type of the object. |
| userId | The userid of the for which to retrieve data for. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

## MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long)

Gets all messages entered by a user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntriesByUser with caller user id", Description =
"Gets all messages for a particular user. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, PagingInfo paging, long
```

```
callerUserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntriesByUser with caller user id", Description =
"Gets all messages for a particular user. ")>
Public Function GetAllMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, paging As PagingInfo, callerUserId As
long) As Collection<MessageBoardData>
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object. |
| objectType | The type of the object. |
| userId | The userid of the for which to retrieve data for. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |
| callerUserId | The userid of the member requesting data. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

## MessageBoard.GetAllMessageBoardEntriesByUser Method (long, EkEnumeration.MessageBoardObjectType, long, PagingInfo, long, bool)

Gets all messages entered by a user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllMessageBoardEntriesByUser with caller user id and moderated
flag", Description = "Gets all messages for a particular user. ")]
public Collection<MessageBoardData> GetAllMessageBoardEntriesByUser(long objectId,
EkEnumeration.MessageBoardObjectType objectType, long userId, PagingInfo paging, long
callerUserId, bool isModerated);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllMessageBoardEntriesByUser with caller user id and moderated
flag", Description = "Gets all messages for a particular user. ")>
Public Function GetAllMessageBoardEntriesByUser(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long, paging As PagingInfo, callerUserId As
long, isModerated As bool) As Collection<MessageBoardData>
```

**Parameters**

| Parameters | Description |
|---|---|
| objectId | The id of the object. |
| objectType | The type of the object. |
| userId | The userid of the for which to retrieve data for. |
| paging | The paging object. Setting "RecordsPerPage" to zero retrieves all rows. |
| callerUserId | The userid of the member requesting data. |
| isModerated | The moderation switch. |

**Returns**

Returns a collection of message board data objects for each record retrieved from the database.

## GetMessageBoardEntry Method

This is the overview for the GetMessageBoardEntry method overload.

### Description

This method is overloaded.

- GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType) - gets a specific message from a message board based on the message ID, object ID and object type.
- GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long) - gets a specific message from a message board based on the message ID, object ID, object type and user ID requesting the data.
- GetMessageBoardEntry (Long, Long, EkEnumeration.MessageBoardObjectType, Long, Bool) - gets a specific message from a message board based on the message ID, object ID, object type, the moderation switch and user ID requesting the data.

### Overload List

| | Name | Description |
|---|---|---|
| | MessageBoard.GetMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType) (see page 1101) | Returns a single row for the requested parameters. |
| | MessageBoard.GetMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType, long) (see page 1102) | Returns a single row for the requested parameters. |
| | MessageBoard.GetMessageBoardEntry (long, long, EkEnumeration.MessageBoardObjectType, long, bool) (see page 1102) | Returns a single row for the requested parameters. |

## MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType)

Returns a single row for the requested parameters.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMessageBoardEntry", Description = "Returns a single row for the
requested parameters.")]
public MessageBoardData GetMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMessageBoardEntry", Description = "Returns a single row for the
requested parameters.")>
Public Function GetMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType) As MessageBoardData
```

### Parameters

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |

### Returns

The message board data object with the data from the database.

## MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long)

Returns a single row for the requested parameters.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMessageBoardEntry with caller user id", Description = "Returns a
single row for the requested parameters.")]
public MessageBoardData GetMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMessageBoardEntry with caller user id", Description = "Returns a
single row for the requested parameters.")>
Public Function GetMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long) As MessageBoardData
```

### Parameters

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object that has the message board. |
| objectType | The type of object. |
| callerUserId | The userid of the member requesting data. |

### Returns

The message board data object with the data from the database.

## MessageBoard.GetMessageBoardEntry Method (long, long, EkEnumeration.MessageBoardObjectType, long, bool)

Returns a single row for the requested parameters.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMessageBoardEntry with caller user id and moderated flag",
Description = "Returns a single row for the requested parameters.")]
public MessageBoardData GetMessageBoardEntry(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, long callerUserId, bool isModerated);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMessageBoardEntry with caller user id and moderated flag",
Description = "Returns a single row for the requested parameters.")>
Public Function GetMessageBoardEntry(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, callerUserId As long, isModerated As bool) As
MessageBoardData
```

### Parameters

| Parameters | Description |
|---|---|
| objectId | The id of the object that has the message board. |

| | |
|---|---|
| objectType | The type of object. |
| callerUserId | The userid of the member requesting data. |
| isModerated | The moderation switch. |
| MessageId | The unique id of the message. |

**Returns**

The message board data object with the data from the database.

## MessageBoard.IsUserMessageBoardAdmin Method

This function determines if the user has administrative privileges for the message board.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsUserMessageBoardAdmin", Description = "This function determines if
the user has administrative privileges for the message board.")]
public bool IsUserMessageBoardAdmin(long objectId, EkEnumeration.MessageBoardObjectType
objectType, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsUserMessageBoardAdmin", Description = "This function determines if
the user has administrative privileges for the message board.")>
Public Function IsUserMessageBoardAdmin(objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, userId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The UserId of the user |

**Returns**

Returns true if the user has administrative privileges.

## UpdateMessageText Method

This is the overview for the UpdateMessageText method overload.

**Description**

This method is overloaded.

- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String) - updates the text of a message based on the message ID, object ID and type, and the new text.
- UpdatedMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long) - updates the text of a message based on the message ID, object ID and type, the new text and the ID of the user changing the text.
- UpdateMessageText (Long, Long, EkEnumeration.MessageBoardObjectType, String, Long, Bool) - updates the text of a message based on the message ID, object ID and type, the new text, the ID of the user changing the text and whether the board is moderated.

**Overload List**

| | Name | Description |
|---|---|---|
| | MessageBoard.UpdateMessageText (long, long, EkEnumeration.MessageBoardObjectType, string) (see page 1104) | Updates the Text of a message. |
| | MessageBoard.UpdateMessageText (long, long, EkEnumeration.MessageBoardObjectType, string, long) (see page 1104) | Updates the Text of a message. |
| | MessageBoard.UpdateMessageText (long, long, EkEnumeration.MessageBoardObjectType, string, long, bool) (see page 1105) | Updates the Text of a message. |

### MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string)

Updates the Text of a message.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateMessageText", Description = "Updates the Text of a message.")]
public MessageBoardData UpdateMessageText(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, string messageText);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateMessageText", Description = "Updates the Text of a message.")>
Public Function UpdateMessageText(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, messageText As string) As MessageBoardData
```

**Parameters**

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object. |
| objectType | The type of the object. |
| messageText | The new message text. |

**Returns**

Returns the updated row.

### MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string, long)

Updates the Text of a message.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateMessageText with caller user id", Description = "Updates the
Text of a message.")]
public MessageBoardData UpdateMessageText(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, string messageText, long callerUserId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateMessageText with caller user id", Description = "Updates the
Text of a message.")>
Public Function UpdateMessageText(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, messageText As string, callerUserId As long) As
MessageBoardData
```

### Parameters

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object. |
| objectType | The type of the object. |
| messageText | The new message text. |
| callerUserId | The userid of the member doing the update. |

### Returns

Returns the updated row.

## MessageBoard.UpdateMessageText Method (long, long, EkEnumeration.MessageBoardObjectType, string, long, bool)

Updates the Text of a message.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateMessageText with caller user id and moderated flag",
Description = "Updates the Text of a message.")]
public MessageBoardData UpdateMessageText(long messageId, long objectId,
EkEnumeration.MessageBoardObjectType objectType, string messageText, long callerUserId, bool
isModerated);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateMessageText with caller user id and moderated flag",
Description = "Updates the Text of a message.")>
Public Function UpdateMessageText(messageId As long, objectId As long, objectType As
EkEnumeration.MessageBoardObjectType, messageText As string, callerUserId As long, isModerated
As bool) As MessageBoardData
```

### Parameters

| Parameters | Description |
|---|---|
| messageId | The unique id of the message. |
| objectId | The id of the object. |
| objectType | The type of the object. |
| messageText | The new message text. |
| callerUserId | The userid of the member doing the update. |
| isModerated | The moderation switch. |

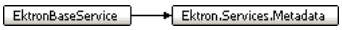
### Returns

Returns the updated row.

# 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

## Metadata Methods

The methods of the Metadata class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddMetaDataType (see page 1106) | Adds a new metadata type to the system |
| | DeleteMetadataType (see page 1107) | Deletes a metadata type from the system. |
| | GetContentMetadataList (see page 1107) | This is the overview for the GetContentMetadataList method overload. |
| | GetMetadataType (see page 1108) | Loads all content metadata. |
| | GetMetaDataTypes (see page 1109) | Loads all the metadata types. |
| | UpdateMetadataType (see page 1109) | Updates the metadata type. |

## Metadata.AddMetaDataType Method

Adds a new metadata type to the system

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddMetaDataType", Description = "Adds a new metadata type
definition")]
public void AddMetaDataType(ref ContentMetaData metaDataItem);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddMetaDataType", Description = "Adds a new metadata type
definition")>
Public Function AddMetaDataType(ByRef metaDataItem As ContentMetaData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteMetadataType", Description = "")]
public void DeleteMetadataType(long MetadataId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMetadataType", Description = "")>
Public Function DeleteMetadataType(MetadataId As long) As void
```

### Parameters

| Parameters | Description |
|---|---|
| MetadataId | Metatype ID |

### Remarks

A user with administrator privileges must be logged in to perform this operation.

## GetContentMetadataList Method

This is the overview for the GetContentMetadataList method overload.

### Description

This method is overloaded.

- GetContentMetadataList (Long) - loads all of the metadata for the given content based on the content's ID.
- GetContentMetadataList (Long, Bool) - loads all of the metadata for the given content base on the content ID. This method allows you to load metadata for a staged content item.

### Overload List

| | Name | Description |
|---|---|---|
| | Metadata.GetContentMetadataList (long) (see page 1107) | Loads all of the metadata for the given content. |
| | Metadata.GetContentMetadataList (long, bool) (see page 1108) | Loads all of the metadata for the given content. |

## Metadata.GetContentMetadataList Method (long)

Loads all of the metadata for the given content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentMetadataList", Description = "")]
public CustomAttributeList GetContentMetadataList(long ContentId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentMetadataList", Description = "")>
Public Function GetContentMetadataList(ContentId As long) As CustomAttributeList
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | The contents ID. |

**Returns**

Ektron.Cms.CustomAttributeList

**Remarks**

Read permissions required.

### Metadata.GetContentMetadataList Method (long, bool)

Loads all of the metadata for the given content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentMetadataList with preview", Description = "")]
public CustomAttributeList GetContentMetadataList(long ContentId, bool preview);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentMetadataList with preview", Description = "")>
Public Function GetContentMetadataList(ContentId As long, preview As bool) As
CustomAttributeList
```

**Parameters**

| Parameters | Description |
|---|---|
| ContentId | The contents ID. |
| preview | True implies get metadata from staging content, else use published content |

**Returns**

Ektron.Cms.CustomAttributeList

**Remarks**

Read permissions required.

### Metadata.GetMetadataType Method

Loads all content metadata.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMetadataType", Description = "")]
```

```
public ContentMetaData GetMetadataType(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMetadataType", Description = "")>
Public Function GetMetadataType(Id As long) As ContentMetaData
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMetaDataTypes", Description = "")]
public ContentMetaData[] GetMetaDataTypes(string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMetaDataTypes", Description = "")>
Public Function GetMetaDataTypes(OrderBy As string) As ContentMetaData()
```

**Parameters**

| Parameters | Description |
|---|---|
| OrderBy | Order by clause |

**Returns**

Ektron.Cms.ContentMetaData()

**Remarks**

User (see page 1167) 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 1106) 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

## Permissions Methods

The methods of the Permissions class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddItemPermission (see page 1110) | Adds the item permissions for a user or a usergroup. |
| | DeleteItemPermission (see page 1111) | Deletes a user or a group from the item's permissions |
| | DisableContentInheritance (see page 1111) | Disable permissions inheritance on the content and copies current permissions to the content. |
| | DisableFolderInheritance (see page 1112) | Disable permissions inheritance on the folder and copies current permissions to the folder. |
| | EnableContentInheritance (see page 1112) | Deletes current permissions from the content and enable permissions inheritance on the content. |
| | EnableFolderInheritance (see page 1113) | Deletes current permissions from the folder and enable permissions inheritance on the folder. |
| | GetUserPermissions (see page 1113) | This is the overview for the GetUserPermissions method overload. |
| | LoadPermissions (see page 1114) | Loads the permission for given object, such as content, folder, task, etc. |
| | UpdateItemPermission (see page 1115) | Updates the item permissions for a user or a usergroup. |

### Permissions.AddItemPermission Method

Adds the item permissions for a user or a usergroup.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddItemPermission", Description = "Adds the item permissions for a
user or a usergroup.")]
```

```
public bool AddItemPermission(UserPermissionData userPermData);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddItemPermission", Description = "Adds the item permissions for a
user or a usergroup.")>
Public Function AddItemPermission(userPermData As UserPermissionData) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| userPermData | User (see page 1167)'s Permissions (see page 1110) Data. |

**Returns**

False if it's successful.

### Permissions.DeleteItemPermission Method

Deletes a user or a group from the item's permissions

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteItemPermission", Description = "Deletes a user or a group from
the item's permissions")]
public bool DeleteItemPermission(long id, bool IsGroupId, long itemId,
EkEnumeration.CMSObjectTypes itemType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteItemPermission", Description = "Deletes a user or a group from
the item's permissions")>
Public Function DeleteItemPermission(id As long, IsGroupId As bool, itemId As long, itemType
As EkEnumeration.CMSObjectTypes) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| id | The user or group ID. |
| IsGroupId | Enter true if the ID is a group ID. |
| itemId | Enter a content ID or folder ID. |
| itemType | Is the given item ID EkEnumeration.CMSObjectTypes.Content (see page 1016) or EkEnumeration.CMSObjectTypes.Folder (see page 1048)? |

**Returns**

False if it's successful.

### Permissions.DisableContentInheritance Method

Disable permissions inheritance on the content and copies current permissions to the content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DisableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool DisableContentInheritance(long id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DisableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function DisableContentInheritance(id As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| id | The content's ID. |

**Returns**

False if it's successful.

### Permissions.DisableFolderInheritance Method

Disable permissions inheritance on the folder and copies current permissions to the folder.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DisableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool DisableFolderInheritance(long id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DisableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function DisableFolderInheritance(id As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| id | The Folder (see page 1048) ID for which to disable permission inheritance. |

**Returns**

False if it's successful.

### Permissions.EnableContentInheritance Method

Deletes current permissions from the content and enable permissions inheritance on the content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EnableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool EnableContentInheritance(long id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "EnableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function EnableContentInheritance(id As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| id | The content's ID. |

### Returns

False if it's successful.

## Permissions.EnableFolderInheritance Method

Deletes current permissions from the folder and enable permissions inheritance on the folder.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EnableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool EnableFolderInheritance(long id);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "EnableFolderInheritance", Description = "Deletes a user or a group
from the item's permissions")>
Public Function EnableFolderInheritance(id As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| id | The Folder (see page 1048) ID. |

### Returns

False if it's successful.

## GetUserPermissions Method

This is the overview for the GetUserPermissions method overload.

### Description

This method is overloaded.

### Overload List

| | Name | Description |
|---|---|---|
| | Permissions.GetUserPermissions (long, string, long, string) (see page 1113) | Loads the user permissions as a array of UserPermissionData. |
| | Permissions.GetUserPermissions (long, string, long, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) (see page 1114) | Loads the user permissions as a array of UserPermissionData. |

## Permissions.GetUserPermissions Method (long, string, long, string)

Loads the user permissions as a array of UserPermissionData.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPermissions", Description = "Loads the user permissions as a
array of UserPermissionData.")]
public UserPermissionData[] GetUserPermissions(long Id, string ItemType, long UserId, string
```

```
UserGroupList);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPermissions", Description = "Loads the user permissions as a
array of UserPermissionData.")>
Public Function GetUserPermissions(Id As long, ItemType As string, UserId As long,
UserGroupList As string) As UserPermissionData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (long, string, long, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType)

Loads the user permissions as a array of UserPermissionData.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPermissions By PermissionType", Description = "Loads the user
permissions as a array of UserPermissionData.")]
public UserPermissionData[] GetUserPermissions(long Id, string ItemType, long UserId, string
UserGroupList, ContentAPI.PermissionUserType PermissionType, ContentAPI.PermissionRequestType
PermissionRequest);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPermissions By PermissionType", Description = "Loads the user
permissions as a array of UserPermissionData.")>
Public Function GetUserPermissions(Id As long, ItemType As string, UserId As long,
UserGroupList As string, PermissionType As ContentAPI.PermissionUserType, PermissionRequest As
ContentAPI.PermissionRequestType) As UserPermissionData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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, folder, task, etc.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "LoadPermissions", Description = "Loads the permission for given
object, such as content, folder, task, etc.")]
public PermissionData LoadPermissions(long Id, string RequestType,
ContentAPI.PermissionResultType Type);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "LoadPermissions", Description = "Loads the permission for given
object, such as content, folder, task, etc.")>
Public Function LoadPermissions(Id As long, RequestType As string, Type As
ContentAPI.PermissionResultType) As PermissionData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The object's ID. |
| RequestType | The object type:= content \| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateItemPermission", Description = "Updates the item permissions
for a user or a usergroup.")]
public bool UpdateItemPermission(UserPermissionData userPermData);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateItemPermission", Description = "Updates the item permissions
for a user or a usergroup.")>
Public Function UpdateItemPermission(userPermData As UserPermissionData) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| userPermData | User (see page 1167)'s Permissions (see page 1110) 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

## SearchManager Methods

The methods of the SearchManager class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | ExecuteSearch (see page 1116) | This is the overview for the ExecuteSearch method overload. |
| | Search (see page 1117) | This is the overview for the Search method overload. |
| | SearchForCategories (see page 1119) | Search (see page 1118) based on taxonomy category. |
| | SearchTable (see page 1119) | Search (see page 1118) html, content, images, multimedia, documents based on content or metadata. |

### ExecuteSearch Method

This is the overview for the ExecuteSearch method overload.

**Description**

Returns the array of ContentData. This method is overloaded.

ExecuteSearch (ContentSearchCondition) (see page 1116)

ExecuteSearch (UserSearchCondition) (see page 1117)

**Overload List**

| | Name | Description |
|---|---|---|
| | SearchManager.ExecuteSearch (ContentSearchCondition) (see page 1116) | Returns the array of ContentData (see page 1378) based on the criteria ContentSearchCondition. |
| | SearchManager.ExecuteSearch (UserSearchCondition) (see page 1117) | Returns the array of UserData (see page 1419) based on the criteria UserSearchCondition. |

### SearchManager.ExecuteSearch Method (ContentSearchCondition)

Returns the array of ContentData (see page 1378) based on the criteria ContentSearchCondition.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ExecuteSearch For Content", Description = "")]
```

```
public ContentData[] ExecuteSearch(ContentSearchCondition condition);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch For Content", Description = "")>
Public Function ExecuteSearch(condition As ContentSearchCondition) As ContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| condition | ContentSearchCondition |

**Returns**

Array of ContentData (see page 1378)

### SearchManager.ExecuteSearch Method (UserSearchCondition)

Returns the array of UserData (see page 1419) based on the criteria UserSearchCondition.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ExecuteSearch For User", Description = "")]
public UserData[] ExecuteSearch(UserSearchCondition condition);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch For User", Description = "")>
Public Function ExecuteSearch(condition As UserSearchCondition) As UserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| condition | UserSearchCondition |

**Returns**

Array of UserData (see page 1419)

### Search Method

This is the overview for the Search method overload.

**Description**

This method is overloaded.

**Overload List**

| | Name | Description |
|---|---|---|
| | SearchManager.Search (MapRequestData, int, int, int) (see page 1118) | Map (see page 340) search for retrieving geographical content from CMS400.Net site. |
| | SearchManager.Search (SearchRequestData, int) (see page 1118) | Search html, content, images, multimedia, documents based on content or metadata. |

| | | |
|---|---|---|
| | SearchManager.Search (string, int) (see page 1119) | Search the site for occurrence(s) of search text. |

### SearchManager.Search Method (MapRequestData, int, int, int)

Map (see page 340) search for retrieving geographical content from CMS400.Net site.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ExecuteSearch search complete", Description = "")]
public MapResponseData[] Search(MapRequestData mapReqData, int index, int count, ref int
resultCount);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch search complete", Description = "")>
Public Function Search(mapReqData As MapRequestData, index As Integer, count As Integer, ByRef
resultCount As Integer) As MapResponseData()
```

**Parameters**

| Parameters | Description |
|---|---|
| mapReqData | Search criteria bundled in a MapRequestData object. |
| index | Page number. |
| count | Page size. |
| resultCount | Number of records returned. |
| context | The current Http request. |

**Returns**

Array of MapResponseData - content satisfying location criteria

### SearchManager.Search Method (SearchRequestData, int)

Search html, content, images, multimedia, documents based on content or metadata.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ExecuteSearch search complete Search Request Data", Description =
"")]
public SearchResponseData[] Search(SearchRequestData searchReqData, ref int resultCount);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch search complete Search Request Data", Description =
"")>
Public Function Search(searchReqData As SearchRequestData, ByRef resultCount As Integer) As
SearchResponseData()
```

**Parameters**

| Parameters | Description |
|---|---|
| searchReqData | Search criteria bundled in a SearchRequestData object. |
| resultCount | Number of records returned. |
| context | The current Http request. |

### Returns

Array of SearchResponseData - content satisfying search criteria

## SearchManager.Search Method (string, int)

Search the site for occurrence(s) of search text.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ExecuteSearch basic", Description = "")]
public SearchResponseData[] Search(string searchText, ref int resultCount);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ExecuteSearch basic", Description = "")>
Public Function Search(searchText As string, ByRef resultCount As Integer) As
SearchResponseData()
```

### Parameters

| Parameters | Description |
|---|---|
| resultCount | Number of records returned. |
| searchReqData | Search criteria bundled in a SearchRequestData object. |

## SearchManager.SearchForCategories Method

Search (see page 1118) based on taxonomy category.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "Search for categories", Description = "")]
public String[] SearchForCategories(SearchRequestData searchReqData, int index, int count, ref
int resultCount);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "Search for categories", Description = "")>
Public Function SearchForCategories(searchReqData As SearchRequestData, index As Integer,
count As Integer, ByRef resultCount As Integer) As String()
```

### Parameters

| Parameters | Description |
|---|---|
| searchReqData | Search (see page 1118) criteria bundled in a SearchRequestData object. |
| index | Page number. |
| count | Page size. |
| resultCount | Number of records returned. |
| context | The current Http request. |

### Returns

Array of String - List of categories

## SearchManager.SearchTable Method

Search (see page 1118) html, content, images, multimedia, documents based on content or metadata.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "Return SearchData Table", Description = "")]
public DataSet SearchTable(SearchRequestData searchReqData, string queryParams, ref int
resultCount);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "Return SearchData Table", Description = "")>
Public Function SearchTable(searchReqData As SearchRequestData, queryParams As string, ByRef
resultCount As Integer) As DataSet
```

### Parameters

| Parameters | Description |
|---|---|
| searchReqData | Search (see page 1118) criteria bundled in a SearchRequestData object. |
| resultCount | Number of records returned. |
| context | The current Http request. |

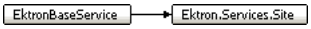
### Returns

Array of SearchResponseData - content satisfying search criteria

## Site Class

Summary description for Site

### Class Hierarchy




### C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Site : EktronBaseService;
```

### Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Site
Inherits EktronBaseService
```

### File

Site.cs

## Site Methods

The methods of the Site class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | BreakTaskPermissionInherit (see page 1121) | Removes the task permissions for a given user. |
| | GetAllActiveLanguages (see page 1121) | Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language Settings area. |
| | GetAllLanguages (see page 1122) | This is the overview for the GetAllLanguages method overload. |

| | | |
|---|---|---|
| | GetClientDayNames (see page 1123) | Returns the day name of a given content language (or site language). |
| | GetClientMonthNames (see page 1123) | Returns the month name of a given content language (or site language) |
| | GetCookieValue (see page 1123) | Reads the cookie value. |
| | GetLanguage (see page 1124) | Returns the details about the given language ID. |
| | GetLanguages (see page 1124) | Gets a list of all languages that meet a given state, for example, active. |
| | GetSiteVariables (see page 1125) | Returns all of the settings data. |
| | GetTaskPermission (see page 1125) | Use this method to find out whether a user has task permissions. |
| | IsTaskPermissionInherited (see page 1126) | Returns true if a user has already granted one of the task permissions. |
| | SetCookieValue (see page 1126) | Resets the cookie value. |
| | SetTaskPermission (see page 1126) | Updates the task permission(s) for the given user or group. |
| | UpdateSiteEnabledLanguages (see page 1127) | Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled. |

## Site.BreakTaskPermissionInherit Method

Removes the task permissions for a given user.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "BreakTaskPermissionInherit", Description = "")]
public void BreakTaskPermissionInherit(long UserId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "BreakTaskPermissionInherit", Description = "")>
Public Function BreakTaskPermissionInherit(UserId As long) As void
```

### Parameters

| Parameters | Description |
|---|---|
| UserId | User (see page 1167) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllActiveLanguages", Description = "")]
public LanguageData[] GetAllActiveLanguages();
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllActiveLanguages", Description = "")>
Public Function GetAllActiveLanguages() As LanguageData()
```

**Returns**

Array of Ektron.Cms.LanguageData

## GetAllLanguages Method

This is the overview for the GetAllLanguages method overload.

**Description**

This method is overloaded.

- GetAllLanguages () (see page 1122) - gets a list of all the available languages sorted by language name in ascending order.
- GetAllLanguages (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) (see page 1122) - gets a list of all the available languages and allows you to choose the order direction of the list.

**Overload List**

| | Name | Description |
|---|---|---|
| | Site.GetAllLanguages () (see page 1122) | Gets a list of all the available languages sorted by language name in ascending order. |
| | Site.GetAllLanguages (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) (see page 1122) | Gets a list of all the available languages. |

## Site.GetAllLanguages Method ()

Gets a list of all the available languages sorted by language name in ascending order.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllLanguagesSortByName", Description = "")]
public LanguageData[] GetAllLanguages();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllLanguagesSortByName", Description = "")>
Public Function GetAllLanguages() As LanguageData()
```

**Returns**

Array of Ektron.Cms.LanguageData sorted by

## Site.GetAllLanguages Method (LanguageDataSet.SortBy, LanguageDataSet.SortDirection)

Gets a list of all the available languages.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllLanguages", Description = "")]
public LanguageData[] GetAllLanguages(LanguageDataSet.SortBy orderBy,
LanguageDataSet.SortDirection direction);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "GetAllLanguages", Description = "")>
Public Function GetAllLanguages(orderBy As LanguageDataSet.SortBy, direction As
LanguageDataSet.SortDirection) As LanguageData()
```

**Parameters**

| Parameters | Description |
|---|---|
| orderBy | Determines the sort order. |
| direction | Determines the sort direction. |

**Returns**

Array of Ektron.Cms.LanguageData

### Site.GetClientDayNames Method

Returns the day name of a given content language (or site language).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetClientDayNames", Description = "")]
public string GetClientDayNames();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetClientDayNames", Description = "")>
Public Function GetClientDayNames() As string
```

**Returns**

String

### Site.GetClientMonthNames Method

Returns the month name of a given content language (or site language)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetClientMonthNames", Description = "")]
public string GetClientMonthNames();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetClientMonthNames", Description = "")>
Public Function GetClientMonthNames() As string
```

**Returns**

String

### Site.GetCookieValue Method

Reads the cookie value.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCookieValue", Description = "")]
public string GetCookieValue(string Keyname);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCookieValue", Description = "")>
Public Function GetCookieValue(Keyname As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Keyname | The Keyname. (Required) |

**Returns**

String

## Site.GetLanguage Method

Returns the details about the given language ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLanguage", Description = "")]
public LanguageData GetLanguage(int Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLanguage", Description = "")>
Public Function GetLanguage(Id As Integer) As LanguageData
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLanguages", Description = "")]
public LanguageDataSet.LanguageDataTable GetLanguages(LanguageDataSet.LanguageState State,
LanguageDataSet.SortBy SortBy, LanguageDataSet.SortDirection SortDirection);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLanguages", Description = "")>
Public Function GetLanguages(State As LanguageDataSet.LanguageState, SortBy As
LanguageDataSet.SortBy, SortDirection As LanguageDataSet.SortDirection) As
LanguageDataSet.LanguageDataTable
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetSiteVariables", Description = "")]
public SettingsData GetSiteVariables(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSiteVariables", Description = "")>
Public Function GetSiteVariables(Id As long) As SettingsData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | User (see page 1167) ID |

**Returns**

Ektron.Cms.SettingsData

**Remarks**

If user ID is passed it returns the additional value of MaxUsersAllowed, AdInstalled and AdAllowed properties.

## Site.GetTaskPermission Method

Use this method to find out whether a user has task permissions.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskPermission", Description = "")]
public long GetTaskPermission(long Id, int Type);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskPermission", Description = "")>
Public Function GetTaskPermission(Id As long, Type As Integer) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | User (see page 1167) or group ID. |
| Type | User (see page 1167) or group (1 = user, 2 = group). |

**Returns**

Returns 1 if the user has one of the task permissions, otherwise 0.

**Remarks**

The user should be logged in to the system. Otherwise, the user will get the permission denied error.

## Site.IsTaskPermissionInherited Method

Returns true if a user has already granted one of the task permissions.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsTaskPermissionInherited", Description = "")]
public bool IsTaskPermissionInherited(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsTaskPermissionInherited", Description = "")>
Public Function IsTaskPermissionInherited(Id As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | User (see page 1167) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetCookieValue", Description = "")]
public void SetCookieValue(string Keyname, string CookieValue);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SetCookieValue", Description = "")>
Public Function SetCookieValue(Keyname As string, CookieValue As string) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| Keyname | The Keyname. (Required) |
| CookieValue | Value |

## Site.SetTaskPermission Method

Updates the task permission(s) for the given user or group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetTaskPermission", Description = "")]
public void SetTaskPermission(long Id, int UserType, int PermissionType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SetTaskPermission", Description = "")>
Public Function SetTaskPermission(Id As long, UserType As Integer, PermissionType As Integer)
As void
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | User (see page 1167) ID or Group ID |
| UserType | User (see page 1167) (1) or Group (2) |
| PermissionType | Ektron.Cms.Common.TaskPermission (1 = Create Task (see page 956), 2 = Redirect Task (see page 956), 3 = Delete Task (see page 956), 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateSiteEnabledLanguages", Description = "")]
public void UpdateSiteEnabledLanguages(string LanguageIDs);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateSiteEnabledLanguages", Description = "")>
Public Function UpdateSiteEnabledLanguages(LanguageIDs As string) As void
```

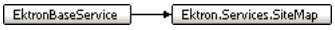
**Parameters**

| Parameters | Description |
|---|---|
| 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

## SiteMap Methods

The methods of the SiteMap class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | GetFolderBreadcrumbPath (see page 1128) | Get the breadcrumb path of the folder in which content block exists. |
| | GetWebsiteMap (see page 1128) | Creates the site map from folder breadcrumb. |
| | RenderFolderBreadcrumbHtml (see page 1129) | Render folder breadcrumb HTML. |
| | RenderWebsiteMap (see page 1129) | Create HTML to render on the page. |

### SiteMap.GetFolderBreadcrumbPath Method

Get the breadcrumb path of the folder in which content block exists.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderBreadcrumbPath", Description = "")]
public SitemapPath[] GetFolderBreadcrumbPath(long id, bool isFolder);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderBreadcrumbPath", Description = "")>
Public Function GetFolderBreadcrumbPath(id As long, isFolder As bool) As SitemapPath()
```

**Parameters**

| Parameters | Description |
|---|---|
| id | A Content (see page 1016) or Folder (see page 1048) ID. |
| isFolder | Set to True if the ID is folder ID. |

**Returns**

Array of nodes in the path.

### SiteMap.GetWebsiteMap Method

Creates the site map from folder breadcrumb.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetWebsiteMap", Description = "")]
public WebsiteMap GetWebsiteMap(long folderId, bool getSameLevelNodes);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetWebsiteMap", Description = "")>
Public Function GetWebsiteMap(folderId As long, getSameLevelNodes As bool) As WebsiteMap
```

**Parameters**

| Parameters | Description |
|---|---|
| folderId | Build the Web sitemap from the specified folder. |
| getSameLevelNodes | Set to True to include all sites at same level as the folder ID. |

### SiteMap.RenderFolderBreadcrumbHtml Method

Render folder breadcrumb HTML.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenderFolderBreadcrumbHtml", Description = "")]
public string RenderFolderBreadcrumbHtml(SitemapPath[] nodes, string pathSeparator, bool
linkNodes, string linkTarget, bool displayVertical, string pageUrl);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RenderFolderBreadcrumbHtml", Description = "")>
Public Function RenderFolderBreadcrumbHtml(nodes As SitemapPath[], pathSeparator As string,
linkNodes As bool, linkTarget As string, displayVertical As bool, pageUrl As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| nodes | The items contained in the breadcrumb path. Ektron.Cms.Common.SitemapPath |
| pathSeparator | A special character that separates the items in a breadcrumb trail. |
| linkNodes | Should the items in the breadcrumb path be Hyperlinked?<br>• **True** = hyperlink breadcrumb items.<br>• **False** = breadcrumb items appear as plain text. |
| linkTarget | Determines the type of window that appears on this Web form when a user clicks an item in the breadcrumb trail. The default is _self.<br>• **_Self** - opens link in the same window.<br>• **_Top** - opens link in a parent window.<br>• **_Blank** - opens link in a new window.<br>• **_Parent** - opens link in the parent frame. |
| displayVerticle | Set to True to display the breadcrumb vertically. Set to false to display horizontally.<br>• **True** = display the breadcrumb trail vertically.<br>• **False** = display the breadcrumb trail horizontally. |

**Returns**

HTML

**Description**

### SiteMap.RenderWebsiteMap Method

Create HTML to render on the page.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenderWebsiteMap", Description = "")]
public string RenderWebsiteMap(WebsiteMap webSitemap, string displayType, string className,
int maxLevel, int startLevel, string pageUrl);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "RenderWebsiteMap", Description = "")>
Public Function RenderWebsiteMap(webSitemap As WebsiteMap, displayType As string, className As
string, maxLevel As Integer, startLevel As Integer, pageUrl As string) As string
```

### Parameters

| Parameters | Description |
|---|---|
| webSitemap | 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

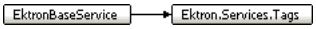
### Remarks

Supported displayType: ListWithDescription | List | RawXml

## Tags Class

Summary description for Tags

### Class Hierarchy




### C#

```
public class Tags : EktronBaseService;
```

### Visual Basic

```
Public Class Tags
Inherits EktronBaseService
```

### File

Tags.cs

### Description

[WebService(Namespace = "http://microsoft.com/webservices/")]

## Tags Methods

The methods of the Tags class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddTag (see page 1131) | Saves a Tag and returns new ID. If the tag already exists, the existing Tag ID is returned. |
| | AddTagToCommunityGroup (see page 1132) | This is the overview for the AddTagToCommunityGroup method overload. |
| | AddTagToContent (see page 1134) | This is the overview for the AddTagToContent method overload. |
| | AddTagToObject (see page 1135) | This is the overview for the AddTagToObject method overload. |
| | AddTagToUser (see page 1137) | This is the overview for the AddTagToUser method overload. |
| | DeleteTagById (see page 1139) | Permanently deletes tag and all associations. Even deletes system tags so this should only be called if user has permission to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser (see page 1140). |

| | | |
|---|---|---|
| | DeleteTagOnObject (see page 1139) | Deletes a user's tag on an object. For example, A user had tagged content ID #101 with "CMS" and now wishes to delete that object tag. |
| | DeleteTagsByUser (see page 1140) | Deletes all tags applied by a user. To delete tags ON a user, use DeleteTagsOnUser (see page 1142). |
| | DeleteTagsOnCommunityGroup (see page 1140) | Deletes all tags on a piece of content. |
| | DeleteTagsOnContent (see page 1141) | Deletes all tags on a piece of content. |
| | DeleteTagsOnObject (see page 1141) | Deletes all tags on an object. |
| | DeleteTagsOnUser (see page 1142) | Deletes all tags applied to a user. To delete tags applied BY a given user, use DeleteTagsByUser (see page 1140) |
| | EditTag (see page 1142) | Updates an existing Tag. |
| | GetAllTagRelatedContent (see page 1142) | Returns a list of content related to the supplied Content (see page 1016) via a common Tag. For example, both content item one and content item two are tagged with "CMS" and are considered related. |
| | GetAllTagRelatedUsers (see page 1143) | Returns a list of users related to the supplied user via a common Tag. For example, both user one and user two are both tagged with "Engineer" and are considered related. |
| | GetAllTags (see page 1143) | This is the overview for the GetAllTags method overload. |
| | GetDefaultTags (see page 1145) | This is the overview for the GetDefaultTags method overload. |
| | GetTagByID (see page 1146) | Returns the Tag with a specified ID. |
| | GetTagByName (see page 1146) | Returns Tag with specified Text |
| | GetTagCloud (see page 1147) | Returns a TagCloud (see page 493) based upon the request. |
| | GetTagsForContent (see page 1147) | Returns an array of tags for a piece of content. |
| | GetTagsForObject (see page 1148) | This is the overview for the GetTagsForObject method overload. |
| | GetTagsForUser (see page 1149) | This is the overview for the GetTagsForUser method overload. |
| | RemoveTagAsDefault (see page 1150) | Removes a tag from the object's default tag list. |
| | SaveTagAsDefault (see page 1151) | Saves a tag to the the object's default list. |

## Tags.AddTag Method

Saves a Tag and returns new ID. If the tag already exists, the existing Tag ID is returned.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTag", Description = "Saves a Tag and returns new Id. If the tag
already exists, the existing Tag Id is returned.")]
public long AddTag(ref TagData tag);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTag", Description = "Saves a Tag and returns new Id. If the tag
already exists, the existing Tag Id is returned.")>
Public Function AddTag(ByRef tag As TagData) As long
```

### Parameters

| Parameters | Description |
|---|---|
| tag | Tag data to save. |

### Returns

An ID for the newly created tag or existing tag.

## AddTagToCommunityGroup Method

This is the overview for the AddTagToCommunityGroup method overload.

### Description

This method is overloaded.

- AddTagToCommunityGroup (String, Long) - applies a tag to a community group by providing the tag's text and the community group's ID.
- AddTagToCommunityGroup (String, Long, Int) - applies a tag in a specified language to a community group by providing the tag's text, the community group's ID, and the language ID.
- AddTagToCommunityGroup (String, Long, Long, Int) - applies a user created tag in a specified language to a community group by providing the tag's text, the community group's ID, the user ID and the language ID.

### Overload List

| | Name | Description |
|---|---|---|
| | Tags.AddTagToCommunityGroup (string, long) (see page 1132) | Applies a Tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language. |
| | Tags.AddTagToCommunityGroup (string, long, int) (see page 1133) | Applies a Tag to a community group in a specified language. If the tag text does not already exist, it is created. Uses the currently logged in user. The newly created ID is accessible via the returned TagData. |
| | Tags.AddTagToCommunityGroup (string, long, long, int) (see page 1133) | Applies a Tag to a community group in a specified language by a given language. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. |

## Tags.AddTagToCommunityGroup Method (string, long)

Applies a Tag to a community group. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToCommunityGroup", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.  Adds the tag for the current user and the Tag is saved
under the current site language.")]
public TagData AddTagToCommunityGroup(string tagText, long groupId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToCommunityGroup", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.  Adds the tag for the current user and the Tag is saved
under the current site language.")>
Public Function AddTagToCommunityGroup(tagText As string, groupId As long) As TagData
```

### Parameters

| Parameters | Description |
|---|---|
| tagText | The Text of the tag being applied to the community group. |
| groupId | The ID of the community group that is being tagged. |

### Returns

TagData structure representing the applied tag.

## Tags.AddTagToCommunityGroup Method (string, long, int)

Applies a Tag to a community group in a specified language. If the tag text does not already exist, it is created. Uses the currently logged in user. The newly created ID is accessible via the returned TagData.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToCommunityGroup with language", Description = "Applies a Tag
to a piece of content.  If the tag text does not already exist, it is created.  Uses the
currently logged in user. The newly created Id is accessible via the returned TagData. ")]
public TagData AddTagToCommunityGroup(string tagText, long groupId, int languageId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToCommunityGroup with language", Description = "Applies a Tag
to a piece of content.  If the tag text does not already exist, it is created.  Uses the
currently logged in user. The newly created Id is accessible via the returned TagData. ")>
Public Function AddTagToCommunityGroup(tagText As string, groupId As long, languageId As
Integer) As TagData
```

### Parameters

| Parameters | Description |
|---|---|
| tagText | Text of tag being applied to a community group. |
| groupId | The ID of the community that is being tagged. |
| languageId | The ID of the language the to apply to the tagText. Supply -1 for all languages. |

### Returns

TagData structure representing the applied tag.

## Tags.AddTagToCommunityGroup Method (string, long, long, int)

Applies a Tag to a community group in a specified language by a given language. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToCommunityGroup with user and language", Description =
"Applies a Tag to a piece of content.  If the tag text does not already exist, it is created.
The newly created Id is accessible via the returned TagData.")]
public TagData AddTagToCommunityGroup(string tagText, long groupId, long userId, int
languageId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToCommunityGroup with user and language", Description =
"Applies a Tag to a piece of content.  If the tag text does not already exist, it is created.
The newly created Id is accessible via the returned TagData.")>
Public Function AddTagToCommunityGroup(tagText As string, groupId As long, userId As long,
languageId As Integer) As TagData
```

### Parameters

| Parameters | Description |
|---|---|
| tagText | The Text of the tag being applied to a community group. |
| groupId | The ID of the community group that is being tagged. |
| userId | The ID of the user applying the tag. |
| languageId | The ID of the language being applied to the tagText. Supply -1 for all languages. |

### Returns

TagData structure representing the applied tag.

## AddTagToContent Method

This is the overview for the AddTagToContent method overload.

### Description

This method is overloaded.

- AddTagToContent (String, Long) - applies a tag to a piece of content based on the tag's text and the content's ID.
- AddTagToContent (String, Long, Long, Int) - applies a user created tag in a specified language to a content item by providing the tag's text, the content's ID, the user ID and the language ID.

### Overload List

| | Name | Description |
|---|---|---|
| | Tags.AddTagToContent (string, long) (see page 1134) | Applies a Tag to a content item. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language. |
| | Tags.AddTagToContent (string, long, long, int) (see page 1135) | Applies a Tag to a content item for a specified user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. |

## Tags.AddTagToContent Method (string, long)

Applies a Tag to a content item. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToContent", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData. Adds the tag for the current user and the Tag is saved
under the current site language.")]
public TagData AddTagToContent(string tagText, long contentId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToContent", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData. Adds the tag for the current user and the Tag is saved
under the current site language.")>
Public Function AddTagToContent(tagText As string, contentId As long) As TagData
```

### Parameters

| Parameters | Description |
|---|---|
| tagText | Text of the tag to apply to object. |
| contentId | The ID of the content item that is being tagged. |

### Returns

TagData structure representing the applied tag.

## Tags.AddTagToContent Method (string, long, long, int)

Applies a Tag to a content item for a specified user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToContent with user and language", Description = "Applies a
Tag to a piece of content.  If the tag text does not already exist, it is created.  The newly
created Id is accessible via the returned TagData.")]
public TagData AddTagToContent(string tagText, long contentId, long userId, int languageId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToContent with user and language", Description = "Applies a
Tag to a piece of content.  If the tag text does not already exist, it is created.  The newly
created Id is accessible via the returned TagData.")>
Public Function AddTagToContent(tagText As string, contentId As long, userId As long,
languageId As Integer) As TagData
```

### Parameters

| Parameters | Description |
|---|---|
| tagText | Text of tag to apply to object. |
| contentId | The ID of content item that is being tagged. |
| userId | The ID of the user applying the tag. |
| languageId | The ID of the language to apply to the tagText. Supply -1 for all languages. |

### Returns

TagData structure representing the applied tag.

## AddTagToObject Method

This is the overview for the AddTagToObject method overload.

### Description

This method is overloaded.

- AddTagToObject (String, Long, EkEnumeration.CMSObjectTypes) - applies a tag to an object based on the tag's text and the object's type and ID.
- AddTagToObject (String, Int, EkEnumeraiton.CMSObjectType, Long, Int) - applies a user created tag in a specified language to an object by providing the tag's text, the object's type and ID, and the language ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.AddTagToObject (string, long, EkEnumeration.CMSObjectTypes) (see page 1136) | Applies a Tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language. |
| | Tags.AddTagToObject (string, long, EkEnumeration.CMSObjectTypes, long, int) (see page 1136) | Applies a Tag to an object by a given user in a specified language. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. |

## Tags.AddTagToObject Method (string, long, EkEnumeration.CMSObjectTypes)

Applies a Tag to an object. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. Adds the tag for the current user and the Tag is saved under the current site language.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToObject", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.  Adds the tag for the current user and the Tag is saved
under the current site language.")]
public TagData AddTagToObject(string tagText, long objectId, EkEnumeration.CMSObjectTypes
objectType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToObject", Description = "Applies a Tag to a piece of
content.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.  Adds the tag for the current user and the Tag is saved
under the current site language.")>
Public Function AddTagToObject(tagText As string, objectId As long, objectType As
EkEnumeration.CMSObjectTypes) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | The text of a tag to being applied to an object. |
| objectId | The ID of the object being tagged. |
| objectType | The type of object represented by objectId. |

**Returns**

TagData structure representing the applied tag.

## Tags.AddTagToObject Method (string, long, EkEnumeration.CMSObjectTypes, long, int)

Applies a Tag to an object by a given user in a specified language. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToObject with user and language", Description = "Applies a Tag
to a piece of content.  If the tag text does not already exist, it is created.  The newly
created Id is accessible via the returned TagData.")]
public TagData AddTagToObject(string tagText, long objectId, EkEnumeration.CMSObjectTypes
```

```
objectType, long userId, int languageId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToObject with user and language", Description = "Applies a Tag
to a piece of content.  If the tag text does not already exist, it is created.  The newly
created Id is accessible via the returned TagData.")>
Public Function AddTagToObject(tagText As string, objectId As long, objectType As
EkEnumeration.CMSObjectTypes, userId As long, languageId As Integer) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | The text of the tag being applied to an object. |
| objectId | The ID of the object that is being tagged. |
| objectType | The type of object represented by objectId. |
| userId | The ID of the user applying the tag. |
| languageId | The ID of the language being applied to the tagText. Supply -1 for all languages. |

**Returns**

TagData structure representing the applied tag.

## AddTagToUser Method

This is the overview for the AddTagToUser method overload.

**Description**

This method is overloaded.

- AddTagToUser (Long, Long) - applies a tag to a user based on a tag's ID and the user's ID. Use this method when a tag already exists.
- AddTagToUser (String, Long) - applies a tag to a user based on the tag's text and the user's ID. Use this method when you are creating a new tag for a user.
- AddTagToUser (String, Long, Int) - applies a tag in a specified language to a user by providing the tag's text, the user ID and the language ID.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.AddTagToUser (long, long) (see page 1137) | Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. |
| | Tags.AddTagToUser (string, long) (see page 1138) | Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. The Tag is saved under the current site language. |
| | Tags.AddTagToUser (string, long, int) (see page 1138) | Applies a Tag to a user for a specified language. If the tag text does not already exist, it is created. The newly created Id is accessible via the returned TagData. |

## Tags.AddTagToUser Method (long, long)

Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToUser with tag id", Description = "Applies a Tag to a user.
If the tag text does not already exist, it is created.  The newly created Id is accessible via
the returned TagData.")]
public void AddTagToUser(long tagId, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToUser with tag id", Description = "Applies a Tag to a user.
If the tag text does not already exist, it is created.  The newly created Id is accessible via
the returned TagData.")>
Public Function AddTagToUser(tagId As long, userId As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| tagId | The ID of tag to apply to object. |
| userId | The ID of the user that is being tagged. |

## Tags.AddTagToUser Method (string, long)

Applies a Tag to a user. If the tag text does not already exist, it is created. The newly created ID is accessible via the returned TagData. The Tag is saved under the current site language.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToUser", Description = "Applies a Tag to a user.  If the tag
text does not already exist, it is created.  The newly created Id is accessible via the
returned TagData. The Tag is saved under the current site language.")]
public TagData AddTagToUser(string tagText, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToUser", Description = "Applies a Tag to a user.  If the tag
text does not already exist, it is created.  The newly created Id is accessible via the
returned TagData. The Tag is saved under the current site language.")>
Public Function AddTagToUser(tagText As string, userId As long) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | The text of the tag being applied to the user. |
| userId | The ID of the user that is being tagged. |

**Returns**

TagData structure representing the applied tag.

## Tags.AddTagToUser Method (string, long, int)

Applies a Tag to a user for a specified language. If the tag text does not already exist, it is created. The newly created Id is accessible via the returned TagData.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTagToUser with language", Description = "Applies a Tag to a
```

```
user.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.")]
public TagData AddTagToUser(string tagText, long userId, int languageId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTagToUser with language", Description = "Applies a Tag to a
user.  If the tag text does not already exist, it is created.  The newly created Id is
accessible via the returned TagData.")>
Public Function AddTagToUser(tagText As string, userId As long, languageId As Integer) As
TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | The text of the tag being applied to the user. |
| userId | The ID of the user that is being tagged. |
| languageId | The ID of the language being applied to the tagText. |

**Returns**

TagData structure representing the applied tag.

### Tags.DeleteTagById Method

Permanently deletes tag and all associations. Even deletes system tags so this should only be called if user has permission to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser (see page 1140).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagById", Description = "Permanently deletes tag and all
associations.  Even deletes system tags so this should only be called if user has permission
to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser.")]
public bool DeleteTagById(long tagId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagById", Description = "Permanently deletes tag and all
associations.  Even deletes system tags so this should only be called if user has permission
to delete system tags. If deleting tags used by a specific user, use DeleteTagsByUser.")>
Public Function DeleteTagById(tagId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| tagId | ID of tag to be deleted. |

### Tags.DeleteTagOnObject Method

Deletes a user's tag on an object. For example, A user had tagged content ID #101 with "CMS" and now wishes to delete that object tag.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagOnObject", Description = "Deletes a user's tag on an
object. For example, User A had tagged content 101 with \"CMS\" and now wishes to delete that
object tag.")]
public bool DeleteTagOnObject(long tagID, long ObjectID, EkEnumeration.CMSObjectTypes
ObjectType, long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagOnObject", Description = "Deletes a user's tag on an
object. For example, User A had tagged content 101 with \"CMS\" and now wishes to delete that
object tag.")>
Public Function DeleteTagOnObject(tagID As long, ObjectID As long, ObjectType As
EkEnumeration.CMSObjectTypes, userId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| tagID | The ID of tag to delete |
| ObjectID | The ID of the object. |
| ObjectType | The type of object associated with the ObjectId. |
| userId | The ID of the user who originally applied the tag and wants to delete it. |

**Returns**

True if operation was successful.

### Tags.DeleteTagsByUser Method

Deletes all tags applied by a user. To delete tags ON a user, use DeleteTagsOnUser (see page 1142).

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsByUser", Description = "Deletes all tags applied by a
user.")]
public bool DeleteTagsByUser(long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsByUser", Description = "Deletes all tags applied by a
user.")>
Public Function DeleteTagsByUser(userId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user whose tags are being deleted |

**Returns**

True if operation was successful.

### Tags.DeleteTagsOnCommunityGroup Method

Deletes all tags on a piece of content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsOnCommunityGroup", Description = "Deletes all tags on a
piece of content.")]
public bool DeleteTagsOnCommunityGroup(long groupId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsOnCommunityGroup", Description = "Deletes all tags on a
```

```
piece of content.")>
Public Function DeleteTagsOnCommunityGroup(groupId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| groupId | The ID of the community group whose tags are being deleted. |

**Returns**

True if operation was successful

### Tags.DeleteTagsOnContent Method

Deletes all tags on a piece of content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsOnContent", Description = "Deletes all tags on a piece of
content.")]
public bool DeleteTagsOnContent(long contentId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsOnContent", Description = "Deletes all tags on a piece of
content.")>
Public Function DeleteTagsOnContent(contentId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | The ID of content item. |

**Returns**

True if operation was successful.

### Tags.DeleteTagsOnObject Method

Deletes all tags on an object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsOnObject", Description = " Deletes all tags on an
object.")]
public bool DeleteTagsOnObject(long objectId, EkEnumeration.CMSObjectTypes objectType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsOnObject", Description = " Deletes all tags on an
object.")>
Public Function DeleteTagsOnObject(objectId As long, objectType As
EkEnumeration.CMSObjectTypes) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| ObjectID | The ID of the object. |
| ObjectType | The type of object associated with the ObjectId. |

### Returns

True if operation was successful.

## Tags.DeleteTagsOnUser Method

Deletes all tags applied to a user. To delete tags applied BY a given user, use DeleteTagsByUser (see page 1140)

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTagsOnUser", Description = "Deletes all tags applied to a
user. To delete tags applied BY a given user, use DeleteTagsByUser")]
public bool DeleteTagsOnUser(long userId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTagsOnUser", Description = "Deletes all tags applied to a
user. To delete tags applied BY a given user, use DeleteTagsByUser")>
Public Function DeleteTagsOnUser(userId As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| userId | The ID of the user whose tags are being deleted. |

### Returns

True if operation was successful

## Tags.EditTag Method

Updates an existing Tag.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EditTag", Description = "Updates an existing Tag.")]
public TagData EditTag(ref TagData tag);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "EditTag", Description = "Updates an existing Tag.")>
Public Function EditTag(ByRef tag As TagData) As TagData
```

### Parameters

| Parameters | Description |
|---|---|
| tag | Tag data to save. |

### Returns

The ID of a newly created tag or existing tag.

## Tags.GetAllTagRelatedContent Method

Returns a list of content related to the supplied Content (see page 1016) via a common Tag. For example, both content item one and content item two are tagged with "CMS" and are considered related.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTagRelatedContent", Description = "Returns a list of content
related to the supplied Content via a common Tag.  For example, both Content one and Content
Two are both tagged with \"CMS\" and are considered related.")]
public ContentData[] GetAllTagRelatedContent(long contentId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTagRelatedContent", Description = "Returns a list of content
related to the supplied Content via a common Tag.  For example, both Content one and Content
Two are both tagged with \"CMS\" and are considered related.")>
Public Function GetAllTagRelatedContent(contentId As long) As ContentData()
```

**Parameters**

| Parameters | Description |
|---|---|
| contentId | This is the content item for which to retrieve related content. |

**Returns**

Returns an array of partially populated ContentData (see page 1378) objects.

### Tags.GetAllTagRelatedUsers Method

Returns a list of users related to the supplied user via a common Tag. For example, both user one and user two are both tagged with "Engineer" and are considered related.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTagRelatedUsers", Description = "Returns a list of users
related to the supplied user via a common Tag.  For example, both user one and user Two are
both tagged with \"Engineer\" and are considered related.")]
public UserData[] GetAllTagRelatedUsers(long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTagRelatedUsers", Description = "Returns a list of users
related to the supplied user via a common Tag.  For example, both user one and user Two are
both tagged with \"Engineer\" and are considered related.")>
Public Function GetAllTagRelatedUsers(userId As long) As UserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | This is the user for which to retrieve related users. |

**Returns**

An array of partially populated UserData (see page 1419) objects.

### GetAllTags Method

This is the overview for the GetAllTags method overload.

**Description**

This method is overloaded.

- GetAllTags (PagingInfo) - retrieves all tags in the system. This method allows for paging.

- GetAllTags (TagRequestData, Long) - retrieves a list of tags in the system based on the requested information and the total number of tags requested.

**Overload List**

| | Name | Description |
|---|---|---|
| | Tags.GetAllTags (PagingInfo) (see page 1144) | Returns a list of all tags in the system. |
| | Tags.GetAllTags (TagRequestData, int) (see page 1144) | Returns a list of all tags in the system. |

## Tags.GetAllTags Method (PagingInfo)

Returns a list of all tags in the system.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTags", Description = "Returns a list of all tags in the
system.")]
public TagData[] GetAllTags(PagingInfo pagingInfo);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTags", Description = "Returns a list of all tags in the
system.")>
Public Function GetAllTags(pagingInfo As PagingInfo) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| pagingInfo | Paging information for call. |

**Returns**

An array of TagData objects.

## Tags.GetAllTags Method (TagRequestData, int)

Returns a list of all tags in the system.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllTags with rewuest data", Description = "Returns a list of all
tags in the system.")]
public TagData[] GetAllTags(TagRequestData request, ref int totalTags);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllTags with rewuest data", Description = "Returns a list of all
tags in the system.")>
Public Function GetAllTags(request As TagRequestData, ByRef totalTags As Integer) As TagData()
```

### Parameters

| Parameters | Description |
|---|---|
| request | Contains the parameters for the tags to retrieve. |
| totalTags | Returns the total number of tags available for the requested parameters. |

### Returns

An array of TagData objects.

## GetDefaultTags Method

This is the overview for the GetDefaultTags method overload.

### Description

This method is overloaded.

- GetDefaultTags (EkEnumeration.CMSObjectTypes) (see page 1145) - gets default tags for a specified object.
- GetDefaultTags (EkEnumeration.CMSObjectTypes, int) (see page 1146) - gets default tags in a specified language for an object.

### Overload List

| | Name | Description |
|---|---|---|
| | Tags.GetDefaultTags (EkEnumeration.CMSObjectTypes) (see page 1145) | Returns a list of all system defined tags in the system. |
| | Tags.GetDefaultTags (EkEnumeration.CMSObjectTypes, int) (see page 1146) | Returns a list of all system defined tags in the system. |

## Tags.GetDefaultTags Method (EkEnumeration.CMSObjectTypes)

Returns a list of all system defined tags in the system.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDefaultTags", Description = "Returns a list of all system defined
tags in the system.")]
public TagData[] GetDefaultTags(EkEnumeration.CMSObjectTypes objectType);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDefaultTags", Description = "Returns a list of all system defined
tags in the system.")>
Public Function GetDefaultTags(objectType As EkEnumeration.CMSObjectTypes) As TagData()
```

### Parameters

| Parameters | Description |
|---|---|
| objectType | The type of object for which to retrieve tags. For example, retrieve all the default tags for CommunityGroups (CMSObjectTypes.CommunityGroup (see page 1001)). |

### Returns

Collection (see page 88) of TagData objects

### Tags.GetDefaultTags Method (EkEnumeration.CMSObjectTypes, int)

Returns a list of all system defined tags in the system.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDefaultTags with language", Description = "Returns a list of all
system defined tags in the system.")]
public TagData[] GetDefaultTags(EkEnumeration.CMSObjectTypes objectType, int languageId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDefaultTags with language", Description = "Returns a list of all
system defined tags in the system.")>
Public Function GetDefaultTags(objectType As EkEnumeration.CMSObjectTypes, languageId As
Integer) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| objectType | The type of object for which to retrieve tags. For example, retrieve all the default tags for CommunityGroups (CMSObjectTypes.CommunityGroup (see page 1001)). |
| languageId | The language of the tags to retrieve. |

**Returns**

Collection (see page 88) of TagData objects.

### Tags.GetTagByID Method

Returns the Tag with a specified ID.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagByID", Description = "Returns the Tag with specified Id")]
public TagData GetTagByID(long tagId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagByID", Description = "Returns the Tag with specified Id")>
Public Function GetTagByID(tagId As long) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagId | The ID of the tag to retrieve. |

**Returns**

Returns a tag found with supplied ID or new tag if no tag is found.

### Tags.GetTagByName Method

Returns Tag with specified Text

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagByName", Description = "Returns Tag with specified Text")]
```

```
public TagData GetTagByName(string tagText);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagByName", Description = "Returns Tag with specified Text")>
Public Function GetTagByName(tagText As string) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | Text of the tag to retrieve. |

**Returns**

Returns a tag found with a supplied ID or new tag if no tag is found.

### Tags.GetTagCloud Method

Returns a TagCloud (see page 493) based upon the request.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagCloud", Description = "Returns a TagCloud based upon the
request.")]
public TagCloud GetTagCloud(TagCloudRequestData request, ref int totalTags);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagCloud", Description = "Returns a TagCloud based upon the
request.")>
Public Function GetTagCloud(request As TagCloudRequestData, ByRef totalTags As Integer) As
TagCloud
```

**Parameters**

| Parameters | Description |
|---|---|
| request | Contains the parameters for the tagcloud to retrieve. |
| totalTags | Returns the total number of tags available in the requested cloud. |

**Returns**

A TagCloud (see page 493) object representing the request.

### Tags.GetTagsForContent Method

Returns an array of tags for a piece of content.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagsForContent", Description = "Returns an array of tags for a
piece of content.")]
public TagData[] GetTagsForContent(long contentId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForContent", Description = "Returns an array of tags for a
piece of content.")>
Public Function GetTagsForContent(contentId As long) As TagData()
```

### Parameters

| Parameters | Description |
|---|---|
| contentId | The ID of the piece of content for which to retrieve tags. |

### Returns

An array of TagData objects that have been applied to the content.

## GetTagsForObject Method

This is the overview for the GetTagsForObject method overload.

### Description

This method is overloaded.

- GetTagsForObject (Long, EkEnumeration.CMSObjectsTypes) - returns an array of tags for a given object based on the object's ID and type, and the language ID.
- GetTagsForObject (Long EkEnumeration.CMSObjectTypes, Int) - returns an array of tags for a given object based on the object's ID and type.

### Overload List

| | Name | Description |
|---|---|---|
| | Tags.GetTagsForObject (long, EkEnumeration.CMSObjectTypes) (see page 1148) | Returns an array of tags for a given object. |
| | Tags.GetTagsForObject (long, EkEnumeration.CMSObjectTypes, int) (see page 1149) | Returns an array of tags for a given object. |

## Tags.GetTagsForObject Method (long, EkEnumeration.CMSObjectTypes)

Returns an array of tags for a given object.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagsForObject", Description = "Returns an array of tags for a
given object.")]
public TagData[] GetTagsForObject(long objectId, EkEnumeration.CMSObjectTypes objectType);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForObject", Description = "Returns an array of tags for a
given object.")>
Public Function GetTagsForObject(objectId As long, objectType As EkEnumeration.CMSObjectTypes)
As TagData()
```

### Parameters

| Parameters | Description |
|---|---|
| objectId | The object ID for which to retrieve tags. |
| objectType | The object type. |

### Returns

An array of TagData objects that have been applied to an object.

## Tags.GetTagsForObject Method (long, EkEnumeration.CMSObjectTypes, int)

Returns an array of tags for a given object.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagsForObject with language)", Description = "Returns an array of
tags for a given object.")]
public TagData[] GetTagsForObject(long objectId, EkEnumeration.CMSObjectTypes objectType, int
languageId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForObject with language)", Description = "Returns an array of
tags for a given object.")>
Public Function GetTagsForObject(objectId As long, objectType As EkEnumeration.CMSObjectTypes,
languageId As Integer) As TagData()
```

### Parameters

| Parameters | Description |
|---|---|
| objectId | The object ID for which to retrieve tags. |
| objectType | The object type. |
| languageId | The language ID of tags to retrieve. |

### Returns

An array of TagData objects that have been applied to Object.

## GetTagsForUser Method

This is the overview for the GetTagsForUser method overload.

### Description

This method is overloaded.

- GetTagsForUser (Long) - returns an array of tags for a user based on the user's ID.
- GetTagsForUser (Long, Int) - returns an array of tags for a specific language associated with a user based on the user's ID and language ID.

### Overload List

| | Name | Description |
|---|---|---|
| | Tags.GetTagsForUser (long) (see page 1150) | Returns an array of tags for a user. |
| | Tags.GetTagsForUser (long, int) (see page 1150) | Returns an array of tags for a user. |

### Tags.GetTagsForUser Method (long)

Returns an array of tags for a user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagsForUser", Description = "Returns an array of tags for a
user.")]
public TagData[] GetTagsForUser(long userId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForUser", Description = "Returns an array of tags for a
user.")>
Public Function GetTagsForUser(userId As long) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user for which to retrieve tags. |

**Returns**

An array of TagData objects that have been applied to the user.

### Tags.GetTagsForUser Method (long, int)

Returns an array of tags for a user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTagsForUser with language)", Description = "Returns an array of
tags for a user.")]
public TagData[] GetTagsForUser(long userId, int languageId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTagsForUser with language)", Description = "Returns an array of
tags for a user.")>
Public Function GetTagsForUser(userId As long, languageId As Integer) As TagData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userId | The ID of the user for which to retrieve tags. |
| languageId | Language of tags to retrieve. |

**Returns**

An array of TagData objects that have been applied to the user.

### Tags.RemoveTagAsDefault Method

Removes a tag from the object's default tag list.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RemoveTagAsDefault", Description = "")]
```

```
public void RemoveTagAsDefault(string tagText, int languageId, EkEnumeration.CMSObjectTypes
objectType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveTagAsDefault", Description = "")>
Public Function RemoveTagAsDefault(tagText As string, languageId As Integer, objectType As
EkEnumeration.CMSObjectTypes) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | The text of the tag. |
| languageId | The ID of the tag being removed. |
| objectType | The type of object. |

### Tags.SaveTagAsDefault Method

Saves a tag to the the object's default list.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SaveTagAsDefault", Description = "")]
public TagData SaveTagAsDefault(string tagText, int languageId, EkEnumeration.CMSObjectTypes
objectType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SaveTagAsDefault", Description = "")>
Public Function SaveTagAsDefault(tagText As string, languageId As Integer, objectType As
EkEnumeration.CMSObjectTypes) As TagData
```

**Parameters**

| Parameters | Description |
|---|---|
| tagText | The text of the tag. |
| languageId | The ID of the language to which the tag will be assigned. |
| objectType | The type of object. |

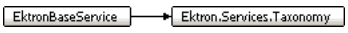
**Returns**

TagData structure representing the tag saved as default.

## Taxonomy Class

The summary description for Taxonomy.

**Class Hierarchy**




**C#**

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Taxonomy : EktronBaseService;
```

**Visual Basic**

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
```

```
Public Class Taxonomy
Inherits EktronBaseService
```

**File**

Taxonomy.cs

## Taxonomy Methods

The methods of the Taxonomy class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddTaxonomyItem (see page 1152) | This is the overview for the AddTaxonomyItem method overload. |
| | AddTaxonomySyncFolder (see page 1153) | Adds a folder entry into the taxonomy. |
| | CreateTaxonomy (see page 1154) | Creates a new taxonomy. |
| | DeleteTaxonomy (see page 1154) | Deletes the existing taxonomy. |
| | GetTaxonomyIdByPath (see page 1154) | Returns the taxonomy ID by a given path. |
| | LoadTaxonomy (see page 1155) | Returns the TaxonomyData with all its subcategories and its items. |
| | ReadAllAssignedCategory (see page 1155) | Returns all the assigned category for the content ID |
| | ReadTaxonomy (see page 1156) | Read the taxonomy data. |
| | RemoveTaxonomyFolder (see page 1156) | Removes the sync up folder from the taxonomy. |
| | RemoveTaxonomyItem (see page 1156) | Removes an item from the taxonomy. |
| | ReOrderTaxonomyItems (see page 1157) | Reorder the display of taxonomy items. |
| | UpdateTaxonomy (see page 1157) | Updates the existing taxonomy. |

### AddTaxonomyItem Method

This is the overview for the AddTaxonomyItem method overload.

**Description**

This method is overloaded.

- AddTaxonomyItem (TaxonomyContentRequest) (see page 1153)
- AddTaxonomyItem (TaxonomyRequest) (see page 1153)

**Overload List**

| | Name | Description |
|---|---|---|
| | Taxonomy.AddTaxonomyItem (TaxonomyContentRequest) (see page 1153) | Inserts an item into the taxonomy. |
| | Taxonomy.AddTaxonomyItem (TaxonomyRequest) (see page 1153) | Inserts an item into the taxonomy. |

### Taxonomy.AddTaxonomyItem Method (TaxonomyContentRequest)

Inserts an item into the taxonomy.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaxonomyItem ContentRequest", Description = "inserts item into
taxonomy")]
public void AddTaxonomyItem(TaxonomyContentRequest TaxonomyReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaxonomyItem ContentRequest", Description = "inserts item into
taxonomy")>
Public Function AddTaxonomyItem(TaxonomyReq As TaxonomyContentRequest) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomycontentrequest |

### Taxonomy.AddTaxonomyItem Method (TaxonomyRequest)

Inserts an item into the taxonomy.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaxonomyItem", Description = "inserts item into taxonomy")]
public void AddTaxonomyItem(TaxonomyRequest TaxonomyReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaxonomyItem", Description = "inserts item into taxonomy")>
Public Function AddTaxonomyItem(TaxonomyReq As TaxonomyRequest) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomyrequest |

### Taxonomy.AddTaxonomySyncFolder Method

Adds a folder entry into the taxonomy.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaxonomySyncFolder", Description = "adds folder entry")]
public void AddTaxonomySyncFolder(TaxonomySyncRequest TaxonomyReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaxonomySyncFolder", Description = "adds folder entry")>
Public Function AddTaxonomySyncFolder(TaxonomyReq As TaxonomySyncRequest) As void
```

### Parameters

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomyrequest |

## Taxonomy.CreateTaxonomy Method

Creates a new taxonomy.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CreateTaxonomy", Description = "creates new taxonomy")]
public long CreateTaxonomy(TaxonomyData TaxonomyReq);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CreateTaxonomy", Description = "creates new taxonomy")>
Public Function CreateTaxonomy(TaxonomyReq As TaxonomyData) As long
```

### Parameters

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomy data |

### Returns

new taxonomy ID

## Taxonomy.DeleteTaxonomy Method

Deletes the existing taxonomy.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaxonomy", Description = "deletes the existing taxonomy")]
public void DeleteTaxonomy(TaxonomyRequest TaxonomyReq);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaxonomy", Description = "deletes the existing taxonomy")>
Public Function DeleteTaxonomy(TaxonomyReq As TaxonomyRequest) As void
```

### Parameters

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomy request |

## Taxonomy.GetTaxonomyIdByPath Method

Returns the taxonomy ID by a given path.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaxonomyIdByPath", Description = "returns taxonomy id by given
path")]
public long GetTaxonomyIdByPath(string TaxonomyPath);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaxonomyIdByPath", Description = "returns taxonomy id by given
path")>
Public Function GetTaxonomyIdByPath(TaxonomyPath As string) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| path | the path to the taxonomy |

**Returns**

taxonomy id

### Taxonomy.LoadTaxonomy Method

Returns the TaxonomyData with all its subcategories and its items.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "LoadTaxonomy", Description = "returns taxonomy id by given path")]
public TaxonomyData LoadTaxonomy(ref TaxonomyRequest TaxonomyReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "LoadTaxonomy", Description = "returns taxonomy id by given path")>
Public Function LoadTaxonomy(ByRef TaxonomyReq As TaxonomyRequest) As TaxonomyData
```

**Parameters**

| Parameters | Description |
|---|---|
| TaxonomyReq | TaxonomyRequest (see page 1414) |

**Returns**

TaxonomyData

### Taxonomy.ReadAllAssignedCategory Method

Returns all the assigned category for the content ID

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ReadAllAssignedCategory", Description = "returns all the assigned
category for the content id")]
public TaxonomyBaseData[] ReadAllAssignedCategory(long ContentId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ReadAllAssignedCategory", Description = "returns all the assigned
category for the content id")>
Public Function ReadAllAssignedCategory(ContentId As long) As TaxonomyBaseData()
```

**Parameters**

| Parameters | Description |
|---|---|
| contentid | the ID for which the categories are returned |

### Returns

An array of taxonomybasedata

## Taxonomy.ReadTaxonomy Method

Read the taxonomy data.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ReadTaxonomy", Description = "read the taxonomy data")]
public TaxonomyData ReadTaxonomy(ref TaxonomyRequest TaxonomyReq);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ReadTaxonomy", Description = "read the taxonomy data")>
Public Function ReadTaxonomy(ByRef TaxonomyReq As TaxonomyRequest) As TaxonomyData
```

### Parameters

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomyrequest |

### Returns

taxonomydata

## Taxonomy.RemoveTaxonomyFolder Method

Removes the sync up folder from the taxonomy.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RemoveTaxonomyFolder", Description = "removes the sync up folder
from taxonomy")]
public void RemoveTaxonomyFolder(TaxonomySyncRequest TaxonomyReq);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveTaxonomyFolder", Description = "removes the sync up folder
from taxonomy")>
Public Function RemoveTaxonomyFolder(TaxonomyReq As TaxonomySyncRequest) As void
```

### Parameters

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomyrequest |

## Taxonomy.RemoveTaxonomyItem Method

Removes an item from the taxonomy.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RemoveTaxonomyItem", Description = "removes item from the taxonomy")]
public void RemoveTaxonomyItem(TaxonomyRequest TaxonomyReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RemoveTaxonomyItem", Description = "removes item from the taxonomy")>
Public Function RemoveTaxonomyItem(TaxonomyReq As TaxonomyRequest) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomyrequest |

### Taxonomy.ReOrderTaxonomyItems Method

Reorder the display of taxonomy items.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ReOrderTaxonomyItems", Description = "reorder display of taxonomy
items")]
public void ReOrderTaxonomyItems(TaxonomyRequest TaxonomyReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ReOrderTaxonomyItems", Description = "reorder display of taxonomy
items")>
Public Function ReOrderTaxonomyItems(TaxonomyReq As TaxonomyRequest) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomyrequest |

### Taxonomy.UpdateTaxonomy Method

Updates the existing taxonomy.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateTaxonomy", Description = "updates the existing taxonomy")]
public void UpdateTaxonomy(TaxonomyData TaxonomyReq);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateTaxonomy", Description = "updates the existing taxonomy")>
Public Function UpdateTaxonomy(TaxonomyReq As TaxonomyData) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| TaxonomyReq | taxonomydata |

## 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

## ThreadedDiscussion Methods

The methods of the ThreadedDiscussion class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddBoard (see page 1158) | Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns numeric ID of the new discussion board. |
| | AddForum (see page 1159) | Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns the numeric ID of new discussion forum. |
| | AddTopic (see page 1160) | This is the overview for the AddTopic method overload. |
| | DeleteBoard (see page 1161) | 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 1161) | 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 1162) | Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value. |
| | GetBoardCategories (see page 1162) | 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 1162) | 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 1163) | This is the overview for the GetDiscussionBoard method overload. |
| | GetDiscussionBoardString (see page 1164) | 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 1165) | Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum (see page 256) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned. |
| | GetRepliesForTopics (see page 1165) | Method returns a list of replies for a given forum ID.<br>**ArrayList Structure**<br>• ArrayListItem(j)(0) ' content_id<br>• ArrayListItem(j)(1) ' comment_total<br>• ArrayListItem(j)(2) ' comment_sum<br>• ArrayListItem(j)(3) ' last_posted<br>• ArrayListItem(j)(4) ' content_language |
| | GetTopic (see page 1166) | Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum (see page 256) 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 1166) | 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. |

### ThreadedDiscussion.AddBoard Method

Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns numeric ID of the new discussion

board.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddBoard", Description = "")]
public long AddBoard(long ParentID, string Name, string Title, bool RequiresAuthentication,
string StyleSheet, string[] Categories);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddBoard", Description = "")>
Public Function AddBoard(ParentID As long, Name As string, Title As string,
RequiresAuthentication As bool, StyleSheet As string, Categories As string[]) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| ParentID | The numeric ID of the folder to which the discussion board is being added. |
| Name | Name of the discussion board. |
| Title | Title of the discussion board. |
| RequiresAuthentication | Enter True if the discussion board requires authentication. If it does not require authentication, enter false. |
| StyleSheet | The stylesheet to apply to the discussion board. |
| Categories | String array of categories for the discussion board. |

**Returns**

Returns a numeric ID of new discussion board (Long).

### ThreadedDiscussion.AddForum Method

Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns the numeric ID of new discussion forum.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddForum", Description = "")]
public long AddForum(long BoardID, string Name, string Description, bool ModeratePosts, bool
LockForum, int SortOrder, long CategoryID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddForum", Description = "")>
Public Function AddForum(BoardID As long, Name As string, Description As string, ModeratePosts
As bool, LockForum As bool, SortOrder As Integer, CategoryID As long) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| BoardID | The numeric ID of the discussion board to which you want to add a forum. |
| Name | The name of forum. |
| Description | String value for the forum summary. |
| ModeratePosts | Are posts moderated? |
| LockForum | Is the forum locked? |
| SortOrder | Sort order for the new forum. |
| CategoryID | The numeric value of the Category ID the new forum belongs to. |

**Returns**

Returns the numeric ID of the new forum (Long).

## AddTopic Method

This is the overview for the AddTopic method overload.

### Description

This method is overloaded.

- AddTopic (Long, String, String) - add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as Long. This method is overloaded.
- AddTopic (Long, String, String, Int) - add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as Long.

### Overload List

| | Name | Description |
|---|---|---|
| | ThreadedDiscussion.AddTopic (long, string, string) (see page 1160) | Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns a numeric topic ID (Long). |
| | ThreadedDiscussion.AddTopic (long, string, string, int) (see page 1160) | Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns a numeric Topic ID (Long). |

## ThreadedDiscussion.AddTopic Method (long, string, string)

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns a numeric topic ID (Long).

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTopic", Description = "")]
public long AddTopic(long ForumID, string Title, string Message);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTopic", Description = "")>
Public Function AddTopic(ForumID As long, Title As string, Message As string) As long
```

### Parameters

| Parameters | Description |
|---|---|
| ForumID | A numeric ID of the forum to add topic to. |
| Title | String value of the topic title. |
| Message | String value of the topic message. |

### Returns

A numeric value for the topic ID (Long).

## ThreadedDiscussion.AddTopic Method (long, string, string, int)

Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns a numeric Topic ID (Long).

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "AddTopic With Language ID", Description = "")]
public long AddTopic(long ForumID, string Title, string Message, int LanguageId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTopic With Language ID", Description = "")>
Public Function AddTopic(ForumID As long, Title As string, Message As string, LanguageId As
Integer) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| ForumID | The numeric ID of the forum to add topic to. |
| Title | String value of the topic title. |
| Message | String value of the topic message. |
| LanguageId | The numeric ID of the Language ID. |

**Returns**

Returns a numeric value for the topic ID (Long).

### ThreadedDiscussion.DeleteBoard Method

Deletes a board with a given ID. This will delete the board as well as all forums, topics and posts under it. BoardID must be greater than 0.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteBoard", Description = "")]
public void DeleteBoard(long BoardID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteBoard", Description = "")>
Public Function DeleteBoard(BoardID As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| BoardID | The numeric ID of the board. This corresponds to the folder ID. |

### ThreadedDiscussion.DeleteForum Method

Deletes a forum with a given ID. This will delete the forum as well as all topics and posts under it. forumID must be greater than 0.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteForum", Description = "")]
public void DeleteForum(long forumID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteForum", Description = "")>
Public Function DeleteForum(forumID As long) As void
```

### Parameters

| Parameters | Description |
|---|---|
| forumID | The numeric ID of the forum. This corresponds to the folder ID. |

## ThreadedDiscussion.DeleteTopic Method

Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTopic", Description = "")]
public bool DeleteTopic(long topicid);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTopic", Description = "")>
Public Function DeleteTopic(topicid As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| post_id | Numeric value of the topic you wish to delete. |

### Returns

Boolean

## ThreadedDiscussion.GetBoardCategories Method

Method to get the Categories for a board by ID. Returns either zero length or populated array. See DiscussionCategory class for more information.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBoardCategories", Description = "")]
public DiscussionCategory[] GetBoardCategories(long boardID);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBoardCategories", Description = "")>
Public Function GetBoardCategories(boardID As long) As DiscussionCategory()
```

### Parameters

| Parameters | Description |
|---|---|
| boardID | The numeric 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBoardForums", Description = "")]
public DiscussionForum[] GetBoardForums(long boardID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBoardForums", Description = "")>
Public Function GetBoardForums(boardID As long) As DiscussionForum()
```

**Parameters**

| Parameters | Description |
|---|---|
| boardID | The numeric value of the board ID. |

**Returns**

Returns an array of type DiscussionForum.

## GetDiscussionBoard Method

This is the overview for the GetDiscussionBoard method overload.

**Description**

This method is overloaded.

- GetDiscussionBoard (Long) - get a discussion board by ID or path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.
- GetDiscussionBoard (String) - get a discussion board by path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

**Overload List**

| | Name | Description |
|---|---|---|
| | ThreadedDiscussion.GetDiscussionBoard (long) (see page 1163) | 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 1164) | 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 (long)

Method to get a board by ID. Returns a DiscussionBoard class for a given ID value. See DiscussionBoard class for more information on the properties of the returned object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDiscussionBoard By Board ID", Description = "")]
public DiscussionBoard GetDiscussionBoard(long boardID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "GetDiscussionBoard By Board ID", Description = "")>
Public Function GetDiscussionBoard(boardID As long) As DiscussionBoard
```

**Parameters**

| Parameters | Description |
|---|---|
| boardID | The numeric value of the board ID. |

**Returns**

Returns a class of type DiscussionBoard.

### ThreadedDiscussion.GetDiscussionBoard Method (string)

Method to get a discussion board by path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDiscussionBoard By Board Path", Description = "")]
public DiscussionBoard GetDiscussionBoard(string boardpath);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDiscussionBoard By Board Path", Description = "")>
Public Function GetDiscussionBoard(boardpath As string) As DiscussionBoard
```

**Parameters**

| Parameters | Description |
|---|---|
| boardpath | String value of the board path. Ex: \Ektron\CMS400.NET. |

**Returns**

Returns a class of type DiscussionBoard.

### ThreadedDiscussion.GetDiscussionBoardString Method

Method to get the rendered view of a discussion board by ID. Returns a string of the output based upon the input parameters. Display of the board can be controlled by using CSS stylesheets.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "", Description = "")]
public string GetDiscussionBoardString(long boardID, string URLpath);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "", Description = "")>
Public Function GetDiscussionBoardString(boardID As long, URLpath As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| boardID | The numeric 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 (see page 256) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetForum", Description = "")]
public DiscussionBoard GetForum(long forumid);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetForum", Description = "")>
Public Function GetForum(forumid As long) As DiscussionBoard
```

**Parameters**

| Parameters | Description |
|---|---|
| forumid | The numeric value of the forum ID. |

**Returns**

An instance of type DiscussionBoard.

### ThreadedDiscussion.GetRepliesForTopics Method

Method returns a list of replies for a given forum ID.

**ArrayList Structure**

- ArrayListItem(j)(0) ' content_id
- ArrayListItem(j)(1) ' comment_total
- ArrayListItem(j)(2) ' comment_sum
- ArrayListItem(j)(3) ' last_posted
- ArrayListItem(j)(4) ' content_language

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetRepliesForTopics", Description = "")]
public ArrayList GetRepliesForTopics(long ForumID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetRepliesForTopics", Description = "")>
Public Function GetRepliesForTopics(ForumID As long) As ArrayList
```

**Parameters**

| Parameters | Description |
|---|---|
| ForumID | The numeric value of the forum ID. |

**Returns**

An instance of type ArrayList.

**Remarks**

This is an older method and is deprecated. The information returned herein was merged into the higher performance GetForum

(see page 1165)(int forumID) method, and that should be used instead.

### ThreadedDiscussion.GetTopic Method

Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum (see page 256) 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTopic", Description = "")]
public DiscussionBoard GetTopic(long topicid);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTopic", Description = "")>
Public Function GetTopic(topicid As long) As DiscussionBoard
```

**Parameters**

| Parameters | Description |
|---|---|
| topicid | The numeric value of the Topic ID. |

**Returns**

An instance of type DiscussionBoard.

### ThreadedDiscussion.SearchReplies Method

Method to obtain an array of topic replies matching search criteria. Will return an instance of EkTasks, which is effectively an array of EkTask objects. If no results are returned, then a zero length EkTasks will be returned.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SearchReplies", Description = "")]
public EkTasks SearchReplies(long[] ForumIDs, string[] SearchTerms, EkEnumeration.SearchTypes
stType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SearchReplies", Description = "")>
Public Function SearchReplies(ForumIDs As long[], SearchTerms As string[], stType As
EkEnumeration.SearchTypes) As EkTasks
```

**Parameters**

| Parameters | Description |
|---|---|
| ForumIDs | A numeric (Long) array of the Forum (see page 256) 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. |

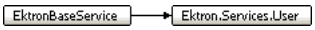
**Returns**

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

## User Methods

The methods of the User class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | ActivateUserAccount (see page 1169) | This is the overview for the ActivateUserAccount method overload. |
| | AddMembershipUser (see page 1170) | Creates new membership user. |
| | AddNewUser (see page 1170) | Adds a new user to the CMS. |
| | AddRolePermission (see page 1171) | Adds permission for the given role. |
| | AddUser (see page 1171) | Creates new cms user |
| | AddUserGroup (see page 1172) | Creates new user group. |
| | AddUserToGroup (see page 1172) | Adds an existing user to an existing group. |
| | AutoLogInUser (see page 1172) | Login (see page 337) the given user credentials. |
| | CreateBlankUserObject (see page 1173) | This is the overview for the CreateBlankUserObject method overload. |
| | DeleteUserByID (see page 1174) | 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 1174) | 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 1175) | Removes a user from the designated group. |
| | DeleteUserGroup (see page 1175) | Removes the group from the system permanently. |
| | EditUserCustomProperties (see page 1176) | This method creates the editable fields(UI) for user's custom properties. |
| | GetActiveUser (see page 1176) | Returns all of the user properties with or without preference. |
| | GetADMapping (see page 1176) | Loads the CMS and Active Directory (see page 189) name. |

| | | |
|---|---|---|
| | GetAllCmsUsers (see page 1177) | Loads all the CMS users. |
| | GetAllCustomProperties (see page 1178) | Returns array of all the custom properties defined in the system. |
| | GetAllUserByCustomProperty (see page 1178) | Returns all the users based on the given properties. |
| | GetAllUserGroups (see page 1179) | This is the overview for the GetAllUserGroups method overload. |
| | GetAllUsers (see page 1180) | This is the overview for the GetAllUsers method overload. |
| | GetGroupsUserIsIn (see page 1181) | Loads all the groups by user. |
| | GetMembershipUser (see page 1182) | Returns user data based on the user's login name (username). |
| | GetUser (see page 1182) | Loads the user details for the given user. |
| | GetUserbyUsername (see page 1183) | Loads the user details for the given user. |
| | GetUserCustomProperties (see page 1183) | Returns an array of custom properties for a given user's ID. |
| | GetUserEmailInfo (see page 1184) | Returns user data for a given user's ID. |
| | GetUserForFolderAdmin (see page 1184) | |
| | GetUserGroup (see page 1185) | Loads a given group's details. |
| | GetUserGroupForFolderAdmin (see page 1185) | Returns all the groups data for the administrator. |
| | GetUserPermissions (see page 1185) | Loads the user permissions as a array of UserPermissionData. |
| | GetUserPreference (see page 1186) | Loads the preferences for the given user. |
| | GetUsers (see page 1187) | This is the overview for the GetUsers method overload. |
| | GetUsersForTask (see page 1188) | Loads the users that have permissions to the given content. |
| | GetUsersNotInGroup (see page 1189) | Loads all the users that are not in the given group. |
| | IsAdmin (see page 1189) | Determines the if the current user is an Administrator. |
| | IsARoleMember (see page 1190) | This is the overview for the IsARoleMember method overload. |
| | IsARoleMemberForContent (see page 1193) | Determines whether a Role Member is allowed to work with a given piece of content as an administrator or not. Administrations privileges are for the given piece of content only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. |
| | IsARoleMemberForFolder (see page 1193) | 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 1194) | Used to log in the user. |
| | ResetMembershipUserPassword (see page 1194) | This is the overview for the ResetMembershipUserPassword method overload. |
| | UnsubscribeUser (see page 1195) | Unsubscribes a user from the system. |
| | UpdateCustomPropertiesItemOrder (see page 1196) | Updates the order of available custom properties. |
| | UpdateMapCoordinate (see page 1196) | Sets the map co-ordinate for the given user property. |
| | UpdateUser (see page 1197) | This is the overview for the UpdateUser method overload. |
| | UpDateUserGroup (see page 1198) | Updates the user group details. |
| | UpdateUserPreferences (see page 1198) | Updates the given user’s preference data. |
| | UserObject (see page 1199) | This is the overview for the UserObject method overload. |

## ActivateUserAccount Method

This is the overview for the ActivateUserAccount method overload.

### Description

This method is overloaded.

- ActivateUserAccount (int) - activates a deleted (deactivated) user into the current/active state using the UserId.
- ActivateUserAccount (string, string) (see page 1169) - activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter.

### Overload List

| | Name | Description |
|---|---|---|
| | User.ActivateUserAccount (long) (see page 1169) | Activates a deleted (deactivated) user into the current/active state using the UserId.<br>This function is overloaded. See Also: ActivateUserAccount(String, String) |
| | User.ActivateUserAccount (string, string) (see page 1169) | Activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter.<br>This function is overloaded. See Also: ActivateUserAccount(Long) |

## User.ActivateUserAccount Method (long)

Activates a deleted (deactivated) user into the current/active state using the UserId.

This function is overloaded. See Also: ActivateUserAccount(String, String)

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ActivateUserAccount By UserID", Description = "")]
public bool ActivateUserAccount(long UserId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ActivateUserAccount By UserID", Description = "")>
Public Function ActivateUserAccount(UserId As long) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| UserId | A unique ID for the required user |

### Returns

Boolean

### Remarks

A user with administrator privileges must be logged in to perform this operation.

## User.ActivateUserAccount Method (string, string)

Activates the deleted (deactivated) user into the current/active state using the Username and GUID (AccountID) parameter.

This function is overloaded. See Also: ActivateUserAccount(Long)

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "ActivateUserAccount By UserName and AccountID", Description = "")]
public bool ActivateUserAccount(string Username, string AccountID);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ActivateUserAccount By UserName and AccountID", Description = "")>
Public Function ActivateUserAccount(Username As string, AccountID As string) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddMembershipUser", Description = "")]
public UserData AddMembershipUser(UserData UserData);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddMembershipUser", Description = "")>
Public Function AddMembershipUser(UserData As UserData) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| UserData | An object of class which represents the corresponding user elements |

**Returns**

Ektron.Cms.UserData (see page 1419)

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

### User.AddNewUser Method

Adds a new user to the CMS.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddNewUser", Description = "")]
[Obsolete("Use \"AddMembershipUser\" instead.")]
public UserData AddNewUser(UserData UserData);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
```

```
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddNewUser", Description = "")>
<Obsolete("Use \"AddMembershipUser\" instead.")>
Public Function AddNewUser(UserData As UserData) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| UserData | An object of class which represents the corresponding user elements |

**Returns**

Ektron.Cms.UserData (see page 1419)

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

## User.AddRolePermission Method

Adds permission for the given role.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddRolePermission", Description = "")]
public bool AddRolePermission(string RoleName);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddRolePermission", Description = "")>
Public Function AddRolePermission(RoleName As string) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleName | Required role name. |

**Returns**

True if successfully added. Otherwise, returns false.

## User.AddUser Method

Creates new cms user

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUser", Description = "Creates new cms user.")]
public UserData AddUser(UserData user);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUser", Description = "Creates new cms user.")>
Public Function AddUser(user As UserData) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| user | Collection (see page 88) of User (see page 1167)'s info. |

### Returns

Ektron.Cms.UserData (see page 1419)

A user with administrator privileges must be logged in to perform this operation.

## User.AddUserGroup Method

Creates new user group.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUserGroup", Description = "")]
public void AddUserGroup(string GroupName, string GroupPath, string GroupDomain);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUserGroup", Description = "")>
Public Function AddUserGroup(GroupName As string, GroupPath As string, GroupDomain As string)
As void
```

### Parameters

| Parameters | Description |
|---|---|
| GroupName | The name of the group. |
| GroupPath | Group path, usually required for Active Directory (see page 189), otherwise CMS group assumed. |
| GroupDomain | Group domain, usually required for Active Directory (see page 189). |

### 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUserToGroup", Description = "")]
public void AddUserToGroup(long UserId, long GroupId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUserToGroup", Description = "")>
Public Function AddUserToGroup(UserId As long, GroupId As long) As void
```

### Parameters

| Parameters | Description |
|---|---|
| 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

Login (see page 337) the given user credentials.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AutoLogInUser", Description = "")]
public UserData AutoLogInUser(string SAMusername, string domain, string ServerName);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AutoLogInUser", Description = "")>
Public Function AutoLogInUser(SAMusername As string, domain As string, ServerName As string)
As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| SAMusername | The user's name. |
| domain | The domain. |
| ServerName | The server name. |

## CreateBlankUserObject Method

This is the overview for the CreateBlankUserObject method overload.

**Description**

This method is overloaded.

- CreateBlankUserObject () - returns the instance of the user data.
- CreateBlankUserObject (ref UserData) - sets the given user data values into the UserData class.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.CreateBlankUserObject () (see page 1173) | Returns the instance of the user data.<br>This function is overloaded. See Also: CreateBlankUserObject(UserData (see page 1419)) |
| | User.CreateBlankUserObject (UserData) (see page 1174) | Sets the given user data values into the UserData (see page 1419) class.<br>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 (see page 1419))

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CreateBlankUserObject", Description = "")]
public UserData CreateBlankUserObject();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CreateBlankUserObject", Description = "")>
Public Function CreateBlankUserObject() As UserData
```

### Returns

Ektron.Cms.UserData (see page 1419)

## User.CreateBlankUserObject Method (UserData)

Sets the given user data values into the UserData (see page 1419) class.

This function is overloaded. See Also: CreateBlankUserObject

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "CreateBlankUserObject By UserData", Description = "")]
public void CreateBlankUserObject(ref UserData UserData);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CreateBlankUserObject By UserData", Description = "")>
Public Function CreateBlankUserObject(ByRef UserData As UserData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserByID", Description = "")]
public void DeleteUserByID(long UserId);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserByID", Description = "")>
Public Function DeleteUserByID(UserId As long) As void
```

### Parameters

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserByIds", Description = "")]
public void DeleteUserByIds(string Ids);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserByIds", Description = "")>
Public Function DeleteUserByIds(Ids As string) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserFromGroup", Description = "")]
public void DeleteUserFromGroup(long UserId, long GroupId, bool ReturnDeleted);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserFromGroup", Description = "")>
Public Function DeleteUserFromGroup(UserId As long, GroupId As long, ReturnDeleted As bool) As
void
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserGroup", Description = "")]
public void DeleteUserGroup(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserGroup", Description = "")>
Public Function DeleteUserGroup(Id As long) As void
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EditUserCustomProperties", Description = "")]
public string EditUserCustomProperties(long UserId, bool ReadOnly);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "EditUserCustomProperties", Description = "")>
Public Function EditUserCustomProperties(UserId As long, ReadOnly As bool) As string
```

### Parameters

| Parameters | Description |
|---|---|
| UserId | User (see page 1167)'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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetActiveUser", Description = "")]
public UserData GetActiveUser(long Id, bool Preference);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetActiveUser", Description = "")>
Public Function GetActiveUser(Id As long, Preference As bool) As UserData
```

### Parameters

| Parameters | Description |
|---|---|
| Id | The user's ID. |
| Preference | Preference is returned only if the value true, otherwise nothing is returned for preference. |

### Returns

Ektron.Cms.UserData (see page 1419)

### Remarks

A user with administrator privileges must be logged in to perform this operation.

## User.GetADMapping Method

Loads the CMS and Active Directory (see page 189) name.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetADMapping", Description = "")]
public AdMappingData[] GetADMapping(long CallerId, string MapType, int Map, int Exclude, int
Editable);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetADMapping", Description = "")>
Public Function GetADMapping(CallerId As long, MapType As string, Map As Integer, Exclude As
Integer, Editable As Integer) As AdMappingData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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 (see page 1044) (0 – false, 1-true). |
| Editable | Flag (see page 1044)(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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCmsUsers", Description = "")]
public UserData[] GetAllCmsUsers(string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCmsUsers", Description = "")>
Public Function GetAllCmsUsers(OrderBy As string) As UserData()
```

### Parameters

| Parameters | Description |
|---|---|
| OrderBy | Used to arrange data in a specific order.<br>The string value can be:<br>• username<br>• firstname<br>• lastname<br>• userid<br>• language<br>• datecreated<br>• lastlogindate |

### Returns

Ektron.Cms.UserData (see page 1419)()

### 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllCustomProperties", Description = "")]
public CustomAttribute[] GetAllCustomProperties();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllCustomProperties", Description = "")>
Public Function GetAllCustomProperties() As CustomAttribute()
```

### Returns

Ektron.Cms.CustomAttribute()

## User.GetAllUserByCustomProperty Method

Returns all the users based on the given properties.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUserByCustomProperty", Description = "")]
public UserData[]
GetAllUserByCustomProperty(Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes PropType,
string PropValues);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUserByCustomProperty", Description = "")>
```

```
Public Function GetAllUserByCustomProperty(PropType As
Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes, PropValues As string) As UserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| PropType | |
| PropValues | Required property |

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

## GetAllUserGroups Method

This is the overview for the GetAllUserGroups method overload.

**Description**

This method is overloaded.

- GetAllUserGroups (EkEnumeration.UserTypes, string) (see page 1179) - loads all of the user groups from the system based on group type and order by information.
- GetAllUserGroups (string) (see page 1180) - loads all of the user groups from the system and allows you to specify order by information.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.GetAllUserGroups (EkEnumeration.UserTypes, string) (see page 1179) | Loads all of the user groups from the system.<br>This function is overloaded. See Also: GetAllUserGroups(String) |
| | User.GetAllUserGroups (string) (see page 1180) | Loads all of the user groups from the system.<br>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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUserGroups By GroupType and OrderBy", Description = "")]
public UserGroupData[] GetAllUserGroups(EkEnumeration.UserTypes GroupType, string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUserGroups By GroupType and OrderBy", Description = "")>
Public Function GetAllUserGroups(GroupType As EkEnumeration.UserTypes, OrderBy As string) As
UserGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| GroupType | GroupType: 1 = Membership (see page 350); 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUserGroups", Description = "")]
public UserGroupData[] GetAllUserGroups(string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUserGroups", Description = "")>
Public Function GetAllUserGroups(OrderBy As string) As UserGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| OrderBy | Arranges data in a specific order. Possible values are GroupName or GroupID. |

**Returns**

Ektron.Cms.UserGroupData()

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

## GetAllUsers Method

This is the overview for the GetAllUsers method overload.

**Description**

This method is overloaded.

GetAllUsers (Long, Long) - gets the User Group Data of all the allowed users in the folder specified.

GetAllUsers (ref UserRequestData) - returns the list of users from the database based on the UserRequestData class.

**Overload List**

| | Name | Description |
|---|---|---|
| | User.GetAllUsers (long, long) (see page 1181) | Gets the User (see page 1167) Group Data of all the allowed users in the folder specified. |
| | User.GetAllUsers (UserRequestData) (see page 1181) | Returns the list of users from the database based on the UserRequestData (see page 1424) class. |

### User.GetAllUsers Method (UserRequestData)

Returns the list of users from the database based on the UserRequestData (see page 1424) class.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUsers by UserRequest", Description = "")]
public UserData[] GetAllUsers(ref UserRequestData userRequest);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUsers by UserRequest", Description = "")>
Public Function GetAllUsers(ByRef userRequest As UserRequestData) As UserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| userRequest | list of constraints to load users |

**Returns**

An array of users as UserData (see page 1419).

### User.GetAllUsers Method (long, long)

Gets the User (see page 1167) Group Data of all the allowed users in the folder specified.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllUsers By CallerId and FolderID", Description = "")]
public UserGroupData[] GetAllUsers(long CallerID, long FolderId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUsers By CallerId and FolderID", Description = "")>
Public Function GetAllUsers(CallerID As long, FolderId As long) As UserGroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| CallerID | The user ID that is logged on to the system. |
| FolderId | The ID of the folder. |

**Returns**

Ektron.Cms.UserGroupData()

**Remarks**

User (see page 1167) login with administrator privileges are required to perform this operation.

### User.GetGroupsUserIsIn Method

Loads all the groups by user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetGroupsUserIsIn", Description = "")]
```

```
public GroupData[] GetGroupsUserIsIn(long UserId, string OrderBy);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetGroupsUserIsIn", Description = "")>
Public Function GetGroupsUserIsIn(UserId As long, OrderBy As string) As GroupData()
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMembershipUser", Description = "")]
public UserData GetMembershipUser(string Username);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMembershipUser", Description = "")>
Public Function GetMembershipUser(Username As string) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| Username | User (see page 1167)'s login name (username). This is required |

**Returns**

Ektron.Cms.UserData (see page 1419)

**Remarks**

User (see page 1167) login with administrator privileges are required to perform this operation.

### User.GetUser Method

Loads the user details for the given user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUser", Description = "")]
public UserData GetUser(long Id, bool Preference, bool ReturnDeleted);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUser", Description = "")>
```

```
Public Function GetUser(Id As long, Preference As bool, ReturnDeleted As bool) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The user ID. |
| Preference | Loads preference only if required (true = load preferences as a part of user data). |

**Returns**

Ektron.Cms.UserData (see page 1419)

**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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserbyUsername", Description = "")]
public UserData GetUserbyUsername(string username);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserbyUsername", Description = "")>
Public Function GetUserbyUsername(username As string) As UserData
```

**Parameters**

| Parameters | Description |
|---|---|
| username | The user username. |

**Returns**

Ektron.Cms.UserData (see page 1419)

**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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserCustomProperties", Description = "")]
public CustomAttribute[] GetUserCustomProperties(long UserId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserCustomProperties", Description = "")>
Public Function GetUserCustomProperties(UserId As long) As CustomAttribute()
```

**Parameters**

| Parameters | Description |
|---|---|
| UserId | User (see page 1167)'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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserEmailInfo", Description = "")]
public UserData GetUserEmailInfo(long Id);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserEmailInfo", Description = "")>
Public Function GetUserEmailInfo(Id As long) As UserData
```

### Parameters

| Parameters | Description |
|---|---|
| Id | The user's ID. |

### Returns

Object of UserData (see page 1419)

### Remarks

A user with administrator privileges must be logged in to perform this operation.

## User.GetUserForFolderAdmin Method

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserForFolderAdmin", Description = "")]
public UserData GetUserForFolderAdmin(long FolderId, long Id, bool Preference, bool
ReturnDeleted);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserForFolderAdmin", Description = "")>
Public Function GetUserForFolderAdmin(FolderId As long, Id As long, Preference As bool,
ReturnDeleted As bool) As UserData
```

### Parameters

| Parameters | Description |
|---|---|
| FolderId | The ID of the folder. |
| Id | |
| Preference | |
| ReturnDeleted | |

### User.GetUserGroup Method

Loads a given group's details.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserGroup", Description = "")]
public UserGroupData GetUserGroup(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserGroup", Description = "")>
Public Function GetUserGroup(Id As long) As UserGroupData
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserGroupForFolderAdmin", Description = "")]
public UserGroupData GetUserGroupForFolderAdmin(long FolderId, long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserGroupForFolderAdmin", Description = "")>
Public Function GetUserGroupForFolderAdmin(FolderId As long, Id As long) As UserGroupData
```

**Parameters**

| Parameters | Description |
|---|---|
| FolderId | The ID of the folder. |
| Id | The ID of the group. |

**Returns**

UserGroupData

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

### User.GetUserPermissions Method

Loads the user permissions as a array of UserPermissionData.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPermissions", Description = "")]
public UserPermissionData[] GetUserPermissions(long Id, string ItemType, long UserId, string
UserGroupList, EkContent.PermissionUserType PermissionType, EkContent.PermissionRequestType
PermissionRequest);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPermissions", Description = "")>
Public Function GetUserPermissions(Id As long, ItemType As string, UserId As long,
UserGroupList As string, PermissionType As EkContent.PermissionUserType, PermissionRequest As
EkContent.PermissionRequestType) As UserPermissionData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The item's numeric ID. |
| ItemType | The item's type. |
| UserId | The user's numeric 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPreference", Description = "")]
public UserPreferenceData GetUserPreference(long Id);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPreference", Description = "")>
Public Function GetUserPreference(Id As long) As UserPreferenceData
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The user's ID. |

**Returns**

Ektron.Cms.UserPreferenceData

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

## GetUsers Method

This is the overview for the GetUsers method overload.

### Description

This method is overloaded.

- GetUsers (Long, String) - get the users from a specific group or load and array of user group data.
- GetUser (String, String) - returns an array of user group data based on the userName and userNameType.

### Overload List

| | Name | Description |
|---|---|---|
| | User.GetUsers (long, string) (see page 1187) | Loads the all the users from the given group. |
| | User.GetUsers (string, string) (see page 1188) | Returns an array of user group data based on the userName and userNameType. |

## User.GetUsers Method (long, string)

Loads the all the users from the given group.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsers By GroupID and OrderBy", Description = "")]
public UserData[] GetUsers(long GroupId, string OrderBy);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsers By GroupID and OrderBy", Description = "")>
Public Function GetUsers(GroupId As long, OrderBy As string) As UserData()
```

### Parameters

| Parameters | Description |
|---|---|
| GroupId | The group's ID |
| OrderBy | The order of the users. The string value can be:<br>• username<br>• firstname<br>• lastname<br>• userid<br>• language<br>• datecreated<br>• lastlogindate |

### Returns

Ektron.Cms.UserData (see page 1419)()

### 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsers", Description = "")]
public UserGroupData[] GetUsers(string userNames, string userNameTypes);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsers", Description = "")>
Public Function GetUsers(userNames As string, userNameTypes As string) As UserGroupData()
```

### Parameters

| Parameters | Description |
|---|---|
| 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.

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsersForTask", Description = "")]
public UserGroupData[] GetUsersForTask(long CallerID, long ContentID);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsersForTask", Description = "")>
Public Function GetUsersForTask(CallerID As long, ContentID As long) As UserGroupData()
```

### Parameters

| Parameters | Description |
|---|---|
| CallerID | The user ID that is logged on to the system. |
| ContentID | The ID of the content. (Required) |

### Returns

Ektron.Cms.UserGroupData()

### Remarks

A user with administrator privileges must be logged in to perform this operation.

### User.GetUsersNotInGroup Method

Loads all the users that are not in the given group.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsersNotInGroup", Description = "")]
public UserData[] GetUsersNotInGroup(long Id, string OrderBy, int GroupType);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsersNotInGroup", Description = "")>
Public Function GetUsersNotInGroup(Id As long, OrderBy As string, GroupType As Integer) As
UserData()
```

**Parameters**

| Parameters | Description |
|---|---|
| Id | The group ID. |
| OrderBy | The order of the users. The string value can be:<br>• username<br>• firstname<br>• lastname<br>• userid<br>• language<br>• datecreated<br>• lastlogindate |

**Returns**

Ektron.Cms.UserData (see page 1419)()

**Description**

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

### User.IsAdmin Method

Determines the if the current user is an Administrator.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsAdmin", Description = "")]
public bool IsAdmin();
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsAdmin", Description = "")>
Public Function IsAdmin() As bool
```

**Returns**

True if the logged in user is administrator. Otherwise, returns false.

## IsARoleMember Method

This is the overview for the IsARoleMember method overload.

### Description

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method is overloaded.

### Overload List

| | Name | Description |
|---|---|---|
| | User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) (see page 1191) | Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.<br>This function is overloaded. See Also:<br>• CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean)<br>• CommonApi.IsARoleMember Method (Integer, Integer, Boolean)<br>• CommonApi.IsARoleMember Method (Integer) |
| | User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, long, bool) (see page 1191) | Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.<br>This function is overloaded. See Also:<br>• CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)<br>• CommonApi.IsARoleMember Method (Integer, Integer, Boolean)<br>• CommonApi.IsARoleMember Method (Integer) |
| | User.IsARoleMember (long) (see page 1192) | Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.<br>This function is overloaded. See Also:<br>• CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)<br>• CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean)<br>• CommonApi.IsARoleMember Method (Integer, Integer, Boolean) |

| | | |
|---|---|---|
| | User.IsARoleMember (long, long, bool) (see page 1192) | Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.<br>This function is overloaded. See Also:<br>• CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean)<br>• CommonApi.IsARoleMember Method (Integer)<br>• 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)
- CommonApi.IsARoleMember Method (Integer)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMember by CMSroleID", Description = "")]
public bool IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by CMSroleID", Description = "")>
Public Function IsARoleMember(RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | _nt_ |

### User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, long, bool)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- CommonApi.IsARoleMember Method (Integer, Integer, Boolean)
- CommonApi.IsARoleMember Method (Integer)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMember by CmsRoleID, UserID and GroupFlag", Description = "")]
public bool IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId, long UserId, bool
GroupFlag);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by CmsRoleID, UserID and GroupFlag", Description = "")>
Public Function IsARoleMember(RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds, UserId As
long, GroupFlag As bool) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | _nt_ |
| GroupFlag | _nt_ |
| nUserId | _nt_ |

## User.IsARoleMember Method (long)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean)
- CommonApi.IsARoleMember Method (Integer, Integer, Boolean)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMember by RoleID", Description = "")]
public bool IsARoleMember(long RoleId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by RoleID", Description = "")>
Public Function IsARoleMember(RoleId As long) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | _nt_ |

## User.IsARoleMember Method (long, long, bool)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

This function is overloaded. See Also:

- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean)
- CommonApi.IsARoleMember Method (Integer)
- CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMember by RoleID, UserId and GroupFlag", Description = "")]
public bool IsARoleMember(long RoleId, long UserId, bool GroupFlag);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by RoleID, UserId and GroupFlag", Description = "")>
Public Function IsARoleMember(RoleId As long, UserId As long, GroupFlag As bool) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | _nt_ |
| GroupFlag | _nt_ |
| nUserId | _nt_ |

## User.IsARoleMemberForContent Method

Determines whether a Role Member is allowed to work with a given piece of content as an administrator or not. Administrations privileges are for the given piece of content only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMemberForContent", Description = "")]
public bool IsARoleMemberForContent(long RoleId, long ContentId, long UserId, bool GroupFlag);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMemberForContent", Description = "")>
Public Function IsARoleMemberForContent(RoleId As long, ContentId As long, UserId As long,
GroupFlag As bool) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| RoleId | |
| ContentId | The numeric content 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMemberForFolder", Description = "")]
public bool IsARoleMemberForFolder(long RoleId, long FolderId, long UserId, bool GroupFlag);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMemberForFolder", Description = "")>
Public Function IsARoleMemberForFolder(RoleId As long, FolderId As long, UserId As long,
GroupFlag As bool) As bool
```

### Parameters

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "LogInUser", Description = "")]
public UserData LogInUser(string username, string Password, string ServerName, string Domain,
string Protocol);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "LogInUser", Description = "")>
Public Function LogInUser(username As string, Password As string, ServerName As string, Domain
As string, Protocol As string) As UserData
```

### Parameters

| Parameters | Description |
|---|---|
| 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 (see page 189). |
| Protocol | Required only when using a secure login or when you are using any valid Active Directory (see page 189) protocols. |

### Returns

Ektron.Cms.UserData (see page 1419)

## ResetMembershipUserPassword Method

This is the overview for the ResetMembershipUserPassword method overload.

### Description

This method is overloaded.

- ResetMembershipUserPassword (string) (see page 1195) - reset the password for the given membership user by providing the username.
- ResetMembershipUserPassword (string, string) (see page 1195) - reset the password for the given membership user by providing the username and verification account ID.

### Overload List

| | Name | Description |
|---|---|---|
| | User.ResetMembershipUserPassword (string) (see page 1195) | Reset the password for the given membership user. |
| | User.ResetMembershipUserPassword (string, string) (see page 1195) | Reset the password for the given membership user. |

### User.ResetMembershipUserPassword Method (string)

Reset the password for the given membership user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ResetMembershipUserPassword", Description = "")]
public string ResetMembershipUserPassword(string Username);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ResetMembershipUserPassword", Description = "")>
Public Function ResetMembershipUserPassword(Username As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Username | User (see page 1167)'s username to be reset |

**Returns**

String

**Remarks**

User (see page 1167) login is required to perform this operation.

### User.ResetMembershipUserPassword Method (string, string)

Reset the password for the given membership user.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ResetMembershipUserPasswordWithVerify", Description = "")]
public string ResetMembershipUserPassword(string Username, string AccountId);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ResetMembershipUserPasswordWithVerify", Description = "")>
Public Function ResetMembershipUserPassword(Username As string, AccountId As string) As string
```

**Parameters**

| Parameters | Description |
|---|---|
| Username | User (see page 1167)'s username to be reset |
| AccountId | If verification is turned on, specify verification id(email to a user) |

**Returns**

String

**Remarks**

User (see page 1167) login is required to perform this operation.

### User.UnsubscribeUser Method

Unsubscribes a user from the system.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UnsubscribeUser", Description = "")]
public bool UnsubscribeUser(string Username, string Password);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UnsubscribeUser", Description = "")>
Public Function UnsubscribeUser(Username As string, Password As string) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateCustomPropertiesItemOrder", Description = "")]
public bool UpdateCustomPropertiesItemOrder(string ItemOrder);
```

**Visual Basic**

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateCustomPropertiesItemOrder", Description = "")>
Public Function UpdateCustomPropertiesItemOrder(ItemOrder As string) As bool
```

**Parameters**

| Parameters | Description |
|---|---|
| ItemOrder | Order of the properties applicable to the user. |

**Returns**

Boolean

**Remarks**

A user with administrator privileges must be logged in to perform this operation.

## User.UpdateMapCoordinate Method

Sets the map co-ordinate for the given user property.

**C#**

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateMapCoordinate", Description = "")]
public void UpdateMapCoordinate(long userId, double latitude, double longitude);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateMapCoordinate", Description = "")>
Public Function UpdateMapCoordinate(userId As long, latitude As double, longitude As double)
As void
```

### Parameters

| Parameters | Description |
|---|---|
| userId | The user's ID. |
| attribId | Map (see page 340) latitude/longitude numeric ID. |
| value | Map (see page 340) latitude or longitude value. |

### Remarks

A user with administrator privileges must be logged in to perform this operation.

## UpdateUser Method

This is the overview for the UpdateUser method overload.

### Description

This method is overloaded.

- UpdateUser (Collection) (see page 1197) - updates the user details based on a collection.
- UpdateUser (UserData) (see page 1198) - updates the user details based on provided UserData.

### Overload List

| | Name | Description |
|---|---|---|
| | User.UpdateUser (Collection) (see page 1197) | Updates the user details. |
| | User.UpdateUser (UserData) (see page 1198) | 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")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateUser", Description = "")]
public void UpdateUser(UserData data);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateUser", Description = "")>
Public Function UpdateUser(data As UserData) As void
```

### Parameters

| Parameters | Description |
|---|---|
| data | A class of type UserData (see page 1419). |

### Remarks

User (see page 1167) 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 (see page 1419). |

### 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(long Id, Collection data);
```

### Visual Basic

```
Public Function UpdateUserPreferences(Id As long, 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

This is the overview for the UserObject method overload.

### Description

This method is overloaded.

- UserObject (Long)- returns user data based on the user's ID.
- UserObject (String) - returns user data based on the user's login or username.

### Overload List

| | Name | Description |
|---|---|---|
| | User.UserObject (long) (see page 1199) | Returns user data based on the user's ID.<br>This function is overloaded. See Also: UserObject(String) |
| | User.UserObject (string) (see page 1200) | Returns user data based on the user's login or username.<br>This function is overloaded. See Also: UserObject(Long) |

## User.UserObject Method (long)

Returns user data based on the user's ID.

This function is overloaded. See Also: UserObject(String)

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UserObject By User ID", Description = "")]
public UserData UserObject(long Id);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UserObject By User ID", Description = "")>
Public Function UserObject(Id As long) As UserData
```

### Parameters

| Parameters | Description |
|---|---|
| Id | The user's ID. |

### Returns

Ektron.Cms.UserData (see page 1419)

## User.UserObject Method (string)

Returns user data based on the user's login or username.

This function is overloaded. See Also: UserObject(Long)

### C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UserObject By UserName", Description = "")]
public UserData UserObject(string UserName);
```

### Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UserObject By UserName", Description = "")>
Public Function UserObject(UserName As string) As UserData
```

### Parameters

| Parameters | Description |
|---|---|
| sUserName | User (see page 1167)'s login or username. |

### Returns

Ektron.Cms.UserData (see page 1419)

# Data Classes

### Classes

| | Name | Description |
|---|---|---|
| | CommunityUserData (see page 1377) | UserData (see page 1419) object plus Community (see page 718) related properties. |
| | ContentData (see page 1378) | A data object that contains the properties that make up a content item. |
| | ContentRatingLabels (see page 1391) | Provides a convinient value class |
| | FormData (see page 1394) | Describes a form design.<br>Declaration Example:<br>`Dim formItem As Ektron.Cms.FormData` |
| | FormFieldDataItem (see page 1399) | Describes an instance of a data value from an individual form field. The data is a name/value pair. |
| | FormFieldDatalist (see page 1400) | Describes an individual datalist suitable as an enumeration for a select list. |
| | FormFieldDatalistItem (see page 1402) | Describes an individual datalist item (most importantly its value and display value). |
| | FormFieldDefinition (see page 1404) | Describes the definition of an individual form field. |
| | FormFieldList (see page 1407) | Describes the definition of all the fields in a specific form. |
| | FormRequestData (see page 1409) | A data object that contains information about a form. |
| | FormSubmittedData (see page 1412) | Describes the data of all the fields submitted from a form at a specific time. |
| | TaxonomyRequest (see page 1414) | This is class Ektron.Cms.TaxonomyRequest. |
| | UriFavorite (see page 1417) | A list of properties that make up a URI favorite data. |
| | UserData (see page 1419) | A data object that contains the properties that make up a user. |
| | UserRequestData (see page 1424) | A data object that contains request information for users. |

### Namespaces

| Name | Description |
|---|---|
| Commerce (see page 1201) | This section includes data classes for eCommerce (see page 573). These classes get or set information about the different types of data contained in eCommerce (see page 573). |
| Common (see page 1364) | This is namespace Ektron.Cms.Common. |

# Ektron.Cms.Commerce Namespace

This section includes data classes for eCommerce (see page 573). These classes get or set information about the different types of data contained in eCommerce (see page 573).

### Classes

| | Name | Description |
|---|---|---|
| | AddressData (see page 1206) | This is class Ektron.Cms.Commerce.AddressData. |
| | AdjustedBasketItem (see page 1208) | Class used to represent basket item price adjustments. |

| | | |
|---|---|---|
| | AppliedCouponData (see page 1211) | This is class Ektron.Cms.Commerce.AppliedCouponData. |
| | AppliedCouponDataCollection (see page 1213) | This is class Ektron.Cms.Commerce.AppliedCouponDataCollection. |
| | AppliedCouponEventArgs (see page 1214) | This is class Ektron.Cms.Commerce.AppliedCouponEventArgs. |
| | Basket (see page 1215) | This is class Ektron.Cms.Commerce.Basket. |
| | BasketCalculatorData (see page 1220) | This is class Ektron.Cms.Commerce.BasketCalculatorData. |
| | BasketData (see page 1224) | This is class Ektron.Cms.Commerce.BasketData. |
| | BasketItem (see page 1226) | Data class representing a basket line item. |
| | BasketItemCollection (see page 1231) | Collection (see page 88) class for BasketItems. |
| | BasketVariantData (see page 1233) | |
| | BundleData (see page 1236) | |
| | CatalogData (see page 1236) | |
| | ContractData (see page 1237) | This is class Ektron.Cms.Commerce.ContractData. |
| | CountryData (see page 1239) | This is class Ektron.Cms.Commerce.CountryData. |
| | CountryTaxRateData (see page 1242) | This is class Ektron.Cms.Commerce.CountryTaxRateData. |
| | CouponData (see page 1243) | This is class Ektron.Cms.Commerce.CouponData. |
| | CouponEntryData (see page 1247) | This is class Ektron.Cms.Commerce.CouponEntryData. |
| | CurrencyData (see page 1250) | Data class representing currencies in system. |
| | CurrencyPricingData (see page 1252) | Represents Pricing Information for agiven currency. |
| | CustomerData (see page 1255) | Data class representing Customers in system. |
| | Dimensions (see page 1259) | Class used to define dimension information. |
| | EntryAttributeData (see page 1262) | This is class Ektron.Cms.Commerce.EntryAttributeData. |
| | EntryData (see page 1264) | Base class for catalog entry data classes. |
| | EntryGroupData (see page 1276) | Data class representing Catalog Entry Groups in system. |
| | EntryGroupItemData (see page 1279) | This is class Ektron.Cms.Commerce.EntryGroupItemData. |
| | ExchangeRateData (see page 1281) | This is class Ektron.Cms.Commerce.ExchangeRateData. |
| | ImageMediaData (see page 1283) | This is class Ektron.Cms.ImageMediaData. |
| | InventoryData (see page 1284) | This is class Ektron.Cms.Commerce.InventoryData. |
| | KitConfigData (see page 1285) | Invitation class used for sending friend requests. |
| | KitData (see page 1287) | |
| | MediaData (see page 1288) | This is class Ektron.Cms.MediaData. |
| | MediaGalleryData (see page 1291) | This is class Ektron.Cms.MediaGalleryData. |
| | OptionGroupData (see page 1292) | Data class representing Option groups for Entry Kits. |
| | OptionGroupItemCollection (see page 1293) | Collection (see page 88) class for OptionGroupItemData (see page 1295). |
| | OptionGroupItemData (see page 1295) | Data class representing option group items in the CMS. Option Group Items are option items for a particular Kit Grouping. |
| | OrderData (see page 1298) | |

| | | |
|---|---|---|
| | OrderLineData (see page 1303) | |
| | OrderPartData (see page 1306) | data class representing order part sin the cms. |
| | PackageData (see page 1309) | Data class used for representing package sizes used for shipping items. |
| | PaymentAuthorization (see page 1311) | Data class representing payment authorization from gateway. |
| | PipelineStageData (see page 1313) | |
| | PostalCodeTaxRateData (see page 1314) | This is class Ektron.Cms.Commerce.PostalCodeTaxRateData. |
| | PricingData (see page 1316) | This is class Ektron.Cms.Commerce.PricingData. |
| | ProductData (see page 1321) | |
| | ProductTypeAttributeData (see page 1322) | This is class Ektron.Cms.Commerce.ProductTypeAttributeData. |
| | ProductTypeData (see page 1325) | Data class representing ecommerce product types. |
| | RecommendationItemData (see page 1326) | Data class representing entry reccomendations in ecommerce system. |
| | RegionData (see page 1330) | Data class for region |
| | RegionTaxRateData (see page 1333) | This is class Ektron.Cms.Commerce.RegionTaxRateData. |
| | ShippingBaseData (see page 1334) | Base data class for representing shipping method information. |
| | ShippingOptionData (see page 1335) | Data class representing a quote for a particular shipping method. |
| | TaxClassData (see page 1336) | Data object representing tax class information. |
| | TaxRateData (see page 1338) | This is class Ektron.Cms.Commerce.TaxRateData. |
| | TaxTypeData (see page 1341) | Data class representing Tax Types. |
| | ThumbnailData (see page 1343) | This is class Ektron.Cms.ThumbnailData. |
| | ThumbnailDefaultData (see page 1344) | This is class Ektron.Cms.ThumbnailDefaultData. |
| | TierPriceData (see page 1346) | This is class Ektron.Cms.Commerce.TierPriceData. |
| | ProductCouponEntryData (see page 1349) | This is class Ektron.Cms.Commerce.ProductCouponEntryData. |
| | WarehouseData (see page 1351) | This is class Ektron.Cms.Commerce.WarehouseData. |
| | Weight (see page 1353) | Class used to define mesurement information. |
| | ShippingMethodData (see page 1355) | Data class used to represent shipping methods defined in the cms. |

### Structs, Records, Enums

| | Name | Description |
|---|---|---|
| | AddressProperty (see page 1357) | Enumeration of Address properties used for sorting and filtering AddressData (see page 1206) objects. |
| | CountryProperty (see page 1357) | Enumeration of CountryData (see page 1239) properties used for sorting and filtering CountryData (see page 1239) objects. |
| | CouponProperty (see page 1358) | Enumeration of Coupon properties used for sorting and filtering CouponData (see page 1243) objects. |
| | CurrencyProperty (see page 1358) | Enumeration of CurrencyData (see page 1250) properties used for sorting and filtering CurrencyData (see page 1250) objects. |
| | CustomerProperty (see page 1358) | Enumeration of Customer properties used for sorting and filtering CustomerData (see page 1255) objects. |
| | EntryAttributeProperty (see page 1359) | Enumeration of EntryAttributeProperty properties available for sorting and filtering. |

| | | |
|---|---|---|
| | EntryGroupItemProperty (see page 1359) | Enumeration of Entry Group Item properties used for sorting and filtering EntryGroupItemData (see page 1279) objects. |
| | EntryGroupProperty (see page 1359) | Enumeration of EntryGroup properties used for sorting and filtering EntryGroupData (see page 1276) objects. |
| | EntryProperty (see page 1360) | Enumeration of EntryData (see page 1264) columns available for sorting and filtering. |
| | ExchangeRateProperty (see page 1360) | Enumeration of ExchangeRate columns available for sorting and filtering. |
| | LinearUnit (see page 1360) | Enumeration of units for measuring length. |
| | OrderProperty (see page 1360) | Enumeration of order properties used for sorting and filtering OrderData (see page 1298) objects. |
| | PackageProperty (see page 1361) | Enumeration representing properties of the packagedata class. |
| | ProductTypeAttributeProperty (see page 1361) | Enumeration of ProductTypeAttributeData (see page 1322) properties available for sorting and filtering. |
| | ProductTypeProperty (see page 1361) | Enumeration of ProductType properties used for sorting and filtering CustomerData (see page 1255) objects. |
| | RegionProperty (see page 1362) | Enumeration of Region properties used for sorting and filtering RegionData (see page 1330) objects. |
| | ShippingMethodProperty (see page 1362) | Enumeration representing properties of the ShippingMethodData (see page 1355) class. |
| | TaxClassProperty (see page 1362) | / Enumeration of TaxClass columns available for sorting and filtering. |
| | TaxRateProperty (see page 1363) | Enumeration of possible values to order Lists of TaxRateData (see page 1338) by. |
| | TaxRateType (see page 1363) | This Represents the default Tax rate Types supported by The CMS. Additonal tax Types may be supported by customers. |
| | TaxTypeProperty (see page 1363) | / Enumeration of TaxType columns available for sorting and filtering. |
| | WarehouseProperty (see page 1364) | Enumeration of Warehouse properties used for sorting and filtering WarehouseData (see page 1351) objects. |
| | WeightUnit (see page 1364) | Enumeration of units for measuring weight. |

# Classes

The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | AddressData (see page 1206) | This is class Ektron.Cms.Commerce.AddressData. |
| | AdjustedBasketItem (see page 1208) | Class used to represent basket item price adjustments. |
| | AppliedCouponData (see page 1211) | This is class Ektron.Cms.Commerce.AppliedCouponData. |
| | AppliedCouponDataCollection (see page 1213) | This is class Ektron.Cms.Commerce.AppliedCouponDataCollection. |
| | AppliedCouponEventArgs (see page 1214) | This is class Ektron.Cms.Commerce.AppliedCouponEventArgs. |
| | Basket (see page 1215) | This is class Ektron.Cms.Commerce.Basket. |
| | BasketCalculatorData (see page 1220) | This is class Ektron.Cms.Commerce.BasketCalculatorData. |
| | BasketData (see page 1224) | This is class Ektron.Cms.Commerce.BasketData. |
| | BasketItem (see page 1226) | Data class representing a basket line item. |
| | BasketItemCollection (see page 1231) | Collection (see page 88) class for BasketItems. |
| | BasketVariantData (see page 1233) | |
| | BundleData (see page 1236) | |

| | | |
|---|---|---|
| | CatalogData (see page 1236) | |
| | ContractData (see page 1237) | This is class Ektron.Cms.Commerce.ContractData. |
| | CountryData (see page 1239) | This is class Ektron.Cms.Commerce.CountryData. |
| | CountryTaxRateData (see page 1242) | This is class Ektron.Cms.Commerce.CountryTaxRateData. |
| | CouponData (see page 1243) | This is class Ektron.Cms.Commerce.CouponData. |
| | CouponEntryData (see page 1247) | This is class Ektron.Cms.Commerce.CouponEntryData. |
| | CurrencyData (see page 1250) | Data class representing currencies in system. |
| | CurrencyPricingData (see page 1252) | Represents Pricing Information for agiven currency. |
| | CustomerData (see page 1255) | Data class representing Customers in system. |
| | Dimensions (see page 1259) | Class used to define dimension information. |
| | EntryAttributeData (see page 1262) | This is class Ektron.Cms.Commerce.EntryAttributeData. |
| | EntryData (see page 1264) | Base class for catalog entry data classes. |
| | EntryGroupData (see page 1276) | Data class representing Catalog Entry Groups in system. |
| | EntryGroupItemData (see page 1279) | This is class Ektron.Cms.Commerce.EntryGroupItemData. |
| | ExchangeRateData (see page 1281) | This is class Ektron.Cms.Commerce.ExchangeRateData. |
| | ImageMediaData (see page 1283) | This is class Ektron.Cms.ImageMediaData. |
| | InventoryData (see page 1284) | This is class Ektron.Cms.Commerce.InventoryData. |
| | KitConfigData (see page 1285) | Invitation class used for sending friend requests. |
| | KitData (see page 1287) | |
| | MediaData (see page 1288) | This is class Ektron.Cms.MediaData. |
| | MediaGalleryData (see page 1291) | This is class Ektron.Cms.MediaGalleryData. |
| | OptionGroupData (see page 1292) | Data class representing Option groups for Entry Kits. |
| | OptionGroupItemCollection (see page 1293) | Collection (see page 88) class for OptionGroupItemData (see page 1295). |
| | OptionGroupItemData (see page 1295) | Data class representing option group items in the CMS. Option Group Items are option items for a particular Kit Grouping. |
| | OrderData (see page 1298) | |
| | OrderLineData (see page 1303) | |
| | OrderPartData (see page 1306) | data class representing order part sin the cms. |
| | PackageData (see page 1309) | Data class used for representing package sizes used for shipping items. |
| | PaymentAuthorization (see page 1311) | Data class representing payment authorization from gateway. |
| | PipelineStageData (see page 1313) | |
| | PostalCodeTaxRateData (see page 1314) | This is class Ektron.Cms.Commerce.PostalCodeTaxRateData. |
| | PricingData (see page 1316) | This is class Ektron.Cms.Commerce.PricingData. |
| | ProductData (see page 1321) | |
| | ProductTypeAttributeData (see page 1322) | This is class Ektron.Cms.Commerce.ProductTypeAttributeData. |

| | | |
|---|---|---|
| | ProductTypeData (see page 1325) | Data class representing ecommerce product types. |
| | RecommendationItemData (see page 1326) | Data class representing entry reccomendations in ecommerce system. |
| | RegionData (see page 1330) | Data class for region |
| | RegionTaxRateData (see page 1333) | This is class Ektron.Cms.Commerce.RegionTaxRateData. |
| | ShippingBaseData (see page 1334) | Base data class for representing shipping method information. |
| | ShippingOptionData (see page 1335) | Data class representing a quote for a particular shipping method. |
| | TaxClassData (see page 1336) | Data object representing tax class information. |
| | TaxRateData (see page 1338) | This is class Ektron.Cms.Commerce.TaxRateData. |
| | TaxTypeData (see page 1341) | Data class representing Tax Types. |
| | ThumbnailData (see page 1343) | This is class Ektron.Cms.ThumbnailData. |
| | ThumbnailDefaultData (see page 1344) | This is class Ektron.Cms.ThumbnailDefaultData. |
| | TierPriceData (see page 1346) | This is class Ektron.Cms.Commerce.TierPriceData. |
| | ProductCouponEntryData (see page 1349) | This is class Ektron.Cms.Commerce.ProductCouponEntryData. |
| | WarehouseData (see page 1351) | This is class Ektron.Cms.Commerce.WarehouseData. |
| | Weight (see page 1353) | Class used to define mesurement information. |
| | ShippingMethodData (see page 1355) | Data class used to represent shipping methods defined in the cms. |

# AddressData Class

### Class Hierarchy

Ektron.Cms.Commerce.AddressData

### C#

```
public class AddressData;
```

### Visual Basic

```
Public Class AddressData
```

### File

AddressData.cs

### Description

This is class Ektron.Cms.Commerce.AddressData.

# AddressData Properties

The properties of the AddressData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AddressLine1 (see page 1207) | Gets or sets Line of address. |

| | | |
|---|---|---|
| | AddressLine2 (see page 1207) | Gets or sets line 2 of address. |
| | City (see page 1207) | Gets or sets the city of the address. |
| | Company (see page 1207) | Gets or sets the name of the company for the address. |
| | Country (see page 1207) | Gets or sets the Country for this address. |
| | Id (see page 1208) | Gets or sets Id of Address. |
| | Name (see page 1208) | Gets or sets the name of this address. |
| | Phone (see page 1208) | Gets or sets the phone number associated with this address. |
| | PostalCode (see page 1208) | Gets or sets the postal code for this address. |
| | Region (see page 1208) | Gets or sets the region for this address. |

## AddressData.AddressLine1 Property

Gets or sets Line of address.

**C#**

```
public string AddressLine1;
```

**Visual Basic**

```
Public Property AddressLine1() As string
```

## AddressData.AddressLine2 Property

Gets or sets line 2 of address.

**C#**

```
public string AddressLine2;
```

**Visual Basic**

```
Public Property AddressLine2() As string
```

## AddressData.City Property

Gets or sets the city of the address.

**C#**

```
public string City;
```

**Visual Basic**

```
Public Property City() As string
```

## AddressData.Company Property

Gets or sets the name of the company for the address.

**C#**

```
public string Company;
```

**Visual Basic**

```
Public Property Company() As string
```

## AddressData.Country Property

Gets or sets the Country for this address.

**C#**

```
public CountryData Country;
```

**Visual Basic**

```
Public Property Country() As CountryData
```

## AddressData.Id Property

Gets or sets Id of Address.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## AddressData.Name Property

Gets or sets the name of this address.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

## AddressData.Phone Property

Gets or sets the phone number associated with this address.

**C#**

```
public string Phone;
```

**Visual Basic**

```
Public Property Phone() As string
```

## AddressData.PostalCode Property

Gets or sets the postal code for this address.

**C#**

```
public string PostalCode;
```

**Visual Basic**

```
Public Property PostalCode() As string
```

## AddressData.Region Property

Gets or sets the region for this address.

**C#**

```
public RegionData Region;
```

**Visual Basic**

```
Public Property Region() As RegionData
```

# AdjustedBasketItem Class

Class used to represent basket item price adjustments.

**Class Hierarchy**

**C#**

```
public class AdjustedBasketItem : BasketItem;
```

**Visual Basic**

```
Public Class AdjustedBasketItem
Inherits BasketItem
```

**File**

AdjustedBasketItem.cs

# AdjustedBasketItem Constructor

This is the overview for the AdjustedBasketItem constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | AdjustedBasketItem.AdjustedBasketItem () (see page 1209) | This is AdjustedBasketItem, a member of class AdjustedBasketItem. |
| | AdjustedBasketItem.AdjustedBasketItem (BasketItem) (see page 1209) | Creates a new AdjustedBasketItem based upon base basketitem. |

## AdjustedBasketItem.AdjustedBasketItem Constructor ()

**C#**

```
public AdjustedBasketItem();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is AdjustedBasketItem, a member of class AdjustedBasketItem.

## AdjustedBasketItem.AdjustedBasketItem Constructor (BasketItem)

Creates a new AdjustedBasketItem based upon base basketitem.

**C#**

```
public AdjustedBasketItem(BasketItem basketItem);
```

**Visual Basic**

```
Public Sub New(basketItem As BasketItem)
```

**Parameters**

| Parameters | Description |
|---|---|
| basketItem | |

# AdjustedBasketItem Properties

The properties of the AdjustedBasketItem class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AdjustedTotal (see page 1210) | Gets or sets the Total Adjusted price for the line item. This could reflect any coupon discounts, surcharges, etc applied to the item. |
| | ApplicableTaxRate (see page 1210) | Gets or sets the TaxRate class applicable to this item. |
| | AppliedCoupons (see page 1210) | Gets the list coupons applied to this basket item. |
| | HasFreeShipping (see page 1210) | Gets or sets whether or not this item has a free shipping coupon applied. |
| | TaxesApplied (see page 1211) | Gets or sets the amount of tax applicable to this item. |
| | TotalCouponDiscount (see page 1211) | Gets or sets the total coupon discount for this basket item. |

## AdjustedBasketItem.AdjustedTotal Property

Gets or sets the Total Adjusted price for the line item. This could reflect any coupon discounts, surcharges, etc applied to the item.

**C#**

```
public override decimal AdjustedTotal;
```

**Visual Basic**

```
Public override Property AdjustedTotal() As decimal
```

## AdjustedBasketItem.ApplicableTaxRate Property

Gets or sets the TaxRate class applicable to this item.

**C#**

```
public TaxRateData ApplicableTaxRate;
```

**Visual Basic**

```
Public Property ApplicableTaxRate() As TaxRateData
```

## AdjustedBasketItem.AppliedCoupons Property

Gets the list coupons applied to this basket item.

**C#**

```
public AppliedCouponDataCollection AppliedCoupons;
```

**Visual Basic**

```
Public Property AppliedCoupons() As AppliedCouponDataCollection
```

## AdjustedBasketItem.HasFreeShipping Property

Gets or sets whether or not this item has a free shipping coupon applied.

**C#**

```
public bool HasFreeShipping;
```

**Visual Basic**

```
Public Property HasFreeShipping() As bool
```

## AdjustedBasketItem.TaxesApplied Property

Gets or sets the amount of tax applicable to this item.

**C#**

```
public decimal TaxesApplied;
```

**Visual Basic**

```
Public Property TaxesApplied() As decimal
```

## AdjustedBasketItem.TotalCouponDiscount Property

Gets or sets the total coupon discount for this basket item.

**C#**

```
public decimal TotalCouponDiscount;
```

**Visual Basic**

```
Public Property TotalCouponDiscount() As decimal
```

# AppliedCouponData Class

**Class Hierarchy**

Ektron.Cms.Commerce.AppliedCouponData

**C#**

```
public class AppliedCouponData;
```

**Visual Basic**

```
Public Class AppliedCouponData
```

**File**

BasketItemCouponData.cs

**Description**

This is class Ektron.Cms.Commerce.AppliedCouponData.

# AppliedCouponData Constructor

This is the overview for the AppliedCouponData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | AppliedCouponData.AppliedCouponData () (see page 1212) | This is AppliedCouponData, a member of class AppliedCouponData. |
| | AppliedCouponData.AppliedCouponData (CouponData, decimal) (see page 1212) | |

## AppliedCouponData.AppliedCouponData Constructor ()

**C#**

```
public AppliedCouponData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is AppliedCouponData, a member of class AppliedCouponData.

## AppliedCouponData.AppliedCouponData Constructor (CouponData, decimal)

**C#**

```
public AppliedCouponData(CouponData coupon, decimal discount);
```

**Visual Basic**

```
Public Sub New(coupon As CouponData, discount As decimal)
```

**Parameters**

| Parameters | Description |
|---|---|
| coupon | sets the coupon being applied to the basketItem. |
| discount | sets the amount the coupon deducts from the price. |

# AppliedCouponData Properties

The properties of the AppliedCouponData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Coupon (see page 1212) | Gets or sets the coupon being applied to the basketItem. |
| | CouponDiscount (see page 1212) | Gets or sets the amount the coupon deducts from the price. |

## AppliedCouponData.Coupon Property

Gets or sets the coupon being applied to the basketItem.

**C#**

```
public CouponData Coupon;
```

**Visual Basic**

```
Public Property Coupon() As CouponData
```

## AppliedCouponData.CouponDiscount Property

Gets or sets the amount the coupon deducts from the price.

**C#**

```
public decimal CouponDiscount;
```

**Visual Basic**

```
Public Property CouponDiscount() As decimal
```

# AppliedCouponDataCollection Class

### Class Hierarchy

### C#

```
public class AppliedCouponDataCollection : Collection<AppliedCouponData>;
```

### Visual Basic

```
Public Class AppliedCouponDataCollection
Inherits Collection
```

### File

BasketItemCouponData.cs

### Description

This is class Ektron.Cms.Commerce.AppliedCouponDataCollection.

# AppliedCouponDataCollection Fields

The fields of the AppliedCouponDataCollection class are listed here.

### Public Fields

| | Name | Description |
|---|---|---|
| | ItemAdded (see page 1213) | This is ItemAdded, a member of class AppliedCouponDataCollection. |

## AppliedCouponDataCollection.ItemAdded Field

### C#

```
public EventHandler<AppliedCouponEventArgs> ItemAdded;
```

### Visual Basic

```
Public ItemAdded As EventHandler<AppliedCouponEventArgs>
```

### Description

This is ItemAdded, a member of class AppliedCouponDataCollection.

# AppliedCouponDataCollection Methods

The methods of the AppliedCouponDataCollection class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | FindByCouponId (see page 1213) | This is FindByCouponId, a member of class AppliedCouponDataCollection. |

## AppliedCouponDataCollection.FindByCouponId Method

### C#

```
public AppliedCouponData FindByCouponId(long id);
```

**Visual Basic**

```
Public Function FindByCouponId(id As long) As AppliedCouponData
```

**Description**

This is FindByCouponId, a member of class AppliedCouponDataCollection.

# AppliedCouponEventArgs Class

**Class Hierarchy**

**C#**

```
public class AppliedCouponEventArgs : EventArgs;
```

**Visual Basic**

```
Public Class AppliedCouponEventArgs
Inherits EventArgs
```

**File**

BasketItemCouponData.cs

**Description**

This is class Ektron.Cms.Commerce.AppliedCouponEventArgs.

# AppliedCouponEventArgs Constructor

This is the overview for the AppliedCouponEventArgs constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | AppliedCouponEventArgs.AppliedCouponEventArgs () (see page 1214) | This is AppliedCouponEventArgs, a member of class AppliedCouponEventArgs. |
| | AppliedCouponEventArgs.AppliedCouponEventArgs (AppliedCouponData) (see page 1214) | This is AppliedCouponEventArgs, a member of class AppliedCouponEventArgs. |

## AppliedCouponEventArgs.AppliedCouponEventArgs Constructor ()

**C#**

```
public AppliedCouponEventArgs();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is AppliedCouponEventArgs, a member of class AppliedCouponEventArgs.

## AppliedCouponEventArgs.AppliedCouponEventArgs Constructor (AppliedCouponData)

**C#**

```
public AppliedCouponEventArgs(AppliedCouponData appliedCoupon);
```

**Visual Basic**

```
Public Sub New(appliedCoupon As AppliedCouponData)
```

**Description**

This is AppliedCouponEventArgs, a member of class AppliedCouponEventArgs.

## AppliedCouponEventArgs Properties

The properties of the AppliedCouponEventArgs class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AppliedCoupon (see page 1215) | This is AppliedCoupon, a member of class AppliedCouponEventArgs. |

### AppliedCouponEventArgs.AppliedCoupon Property

**C#**

```
public AppliedCouponData AppliedCoupon;
```

**Visual Basic**

```
Public Property AppliedCoupon() As AppliedCouponData
```

**Description**

This is AppliedCoupon, a member of class AppliedCouponEventArgs.

# Basket Class

**Class Hierarchy**

**C#**

```
public abstract class Basket : BasketData, IBasket;
```

**Visual Basic**

```
Public abstract Class Basket
Inherits BasketData
Inherits IBasket
```

**File**

IBasket.cs

**Description**

This is class Ektron.Cms.Commerce.Basket.

## Basket Methods

The methods of the Basket class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddProduct (see page 1216) | This is the overview for the AddProduct method overload. |
| | AddProductVariant (see page 1217) | This is AddProductVariant, a member of class Basket. |
| | Decrement (see page 1217) | This is Decrement, a member of class Basket. |
| | GetItemIndex (see page 1217) | This is GetItemIndex, a member of class Basket. |
| | GetKitConfiguration (see page 1218) | This is GetKitConfiguration, a member of class Basket. |
| | GetKitConfigurationAsAdmin (see page 1218) | This is GetKitConfigurationAsAdmin, a member of class Basket. |
| | IncrementItemQuantity (see page 1218) | This is IncrementItemQuantity, a member of class Basket. |
| | MakeDefaultBasket (see page 1218) | This is MakeDefaultBasket, a member of class Basket. |
| | RemoveItem (see page 1218) | This is RemoveItem, a member of class Basket. |
| | SaveBasket (see page 1219) | This is SaveBasket, a member of class Basket. |
| | UpdateItemQuantity (see page 1219) | This is UpdateItemQuantity, a member of class Basket. |
| | UpdateKitConfiguration (see page 1219) | This is UpdateKitConfiguration, a member of class Basket. |

## AddProduct Method

This is the overview for the AddProduct method overload.

### Overload List

| | Name | Description |
|---|---|---|
| | Basket.AddProduct (long) (see page 1216) | This is AddProduct, a member of class Basket. |
| | Basket.AddProduct (long, int) (see page 1217) | Adds product to cart. |
| | Basket.AddProduct (long, KitConfigData) (see page 1216) | This is AddProduct, a member of class Basket. |

### Basket.AddProduct Method (long)

**C#**

```
public abstract long AddProduct(long productId);
```

**Visual Basic**

```
Public abstract Function AddProduct(productId As long) As long
```

**Description**

This is AddProduct, a member of class Basket.

### Basket.AddProduct Method (long, KitConfigData)

**C#**

```
public abstract long AddProduct(long productId, KitConfigData config);
```

**Visual Basic**

```
Public abstract Function AddProduct(productId As long, config As KitConfigData) As long
```

**Description**

This is AddProduct, a member of class Basket.

### Basket.AddProduct Method (long, int)

Adds product to cart.

**C#**

```
public abstract long AddProduct(long productId, int quantity);
```

**Visual Basic**

```
Public abstract Function AddProduct(productId As long, quantity As Integer) As long
```

**Parameters**

| Parameters | Description |
|---|---|
| productId | Id (see page 1225) of product to add to basket. |
| quantity | quantity of product to add to cart. |

**Returns**

returns basket item Id (see page 1225).

## Basket.AddProductVariant Method

**C#**

```
public abstract long AddProductVariant(long productId, long variantId);
```

**Visual Basic**

```
Public abstract Function AddProductVariant(productId As long, variantId As long) As long
```

**Description**

This is AddProductVariant, a member of class Basket.

## Basket.Decrement Method

**C#**

```
public abstract void Decrement(long basketItemId);
```

**Visual Basic**

```
Public abstract Function Decrement(basketItemId As long) As void
```

**Description**

This is Decrement, a member of class Basket.

## Basket.GetItemIndex Method

**C#**

```
public abstract int GetItemIndex(long basketItemId);
```

**Visual Basic**

```
Public abstract Function GetItemIndex(basketItemId As long) As Integer
```

**Description**

This is GetItemIndex, a member of class Basket.

## Basket.GetKitConfiguration Method

**C#**

```
public abstract KitConfigData GetKitConfiguration(long basketItemId);
```

**Visual Basic**

```
Public abstract Function GetKitConfiguration(basketItemId As long) As KitConfigData
```

**Description**

This is GetKitConfiguration, a member of class Basket.

## Basket.GetKitConfigurationAsAdmin Method

**C#**

```
public abstract KitConfigData GetKitConfigurationAsAdmin(long basketItemId);
```

**Visual Basic**

```
Public abstract Function GetKitConfigurationAsAdmin(basketItemId As long) As KitConfigData
```

**Description**

This is GetKitConfigurationAsAdmin, a member of class Basket.

## Basket.IncrementItemQuantity Method

**C#**

```
public abstract void IncrementItemQuantity(long basketItemId);
```

**Visual Basic**

```
Public abstract Function IncrementItemQuantity(basketItemId As long) As void
```

**Description**

This is IncrementItemQuantity, a member of class Basket.

## Basket.MakeDefaultBasket Method

**C#**

```
public abstract void MakeDefaultBasket();
```

**Visual Basic**

```
Public abstract Function MakeDefaultBasket() As void
```

**Description**

This is MakeDefaultBasket, a member of class Basket.

## Basket.RemoveItem Method

**C#**

```
public abstract void RemoveItem(long basketItemId);
```

**Visual Basic**

```
Public abstract Function RemoveItem(basketItemId As long) As void
```

**Description**

This is RemoveItem, a member of class Basket.

## Basket.SaveBasket Method

**C#**

```
public abstract void SaveBasket();
```

**Visual Basic**

```
Public abstract Function SaveBasket() As void
```

**Description**

This is SaveBasket, a member of class Basket.

## Basket.UpdateItemQuantity Method

**C#**

```
public abstract void UpdateItemQuantity(long basketItemId, int productQuantity);
```

**Visual Basic**

```
Public abstract Function UpdateItemQuantity(basketItemId As long, productQuantity As Integer)
As void
```

**Description**

This is UpdateItemQuantity, a member of class Basket.

## Basket.UpdateKitConfiguration Method

**C#**

```
public abstract long UpdateKitConfiguration(long productId, KitConfigData config);
```

**Visual Basic**

```
Public abstract Function UpdateKitConfiguration(productId As long, config As KitConfigData) As
long
```

**Description**

This is UpdateKitConfiguration, a member of class Basket.

# Basket Properties

The properties of the Basket class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | RequestInformation (see page 1220) | This is RequestInformation, a member of class Basket. |

### Basket.RequestInformation Property

**C#**

```
public abstract EkRequestInformation RequestInformation;
```

**Visual Basic**

```
Public abstract Property RequestInformation() As EkRequestInformation
```

**Description**

This is RequestInformation, a member of class Basket.

# BasketCalculatorData Class

**Class Hierarchy**

**C#**

```
public class BasketCalculatorData;
```

**Visual Basic**

```
Public Class BasketCalculatorData
```

**File**

BasketCalculatorData.cs

**Description**

This is class Ektron.Cms.Commerce.BasketCalculatorData.

## BasketCalculatorData Constructor

This is the overview for the BasketCalculatorData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | BasketCalculatorData.BasketCalculatorData () (see page 1220) | This is BasketCalculatorData, a member of class BasketCalculatorData. |
| | BasketCalculatorData.BasketCalculatorData (BasketData) (see page 1221) | This is BasketCalculatorData, a member of class BasketCalculatorData. |

### BasketCalculatorData.BasketCalculatorData Constructor ()

**C#**

```
public BasketCalculatorData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is BasketCalculatorData, a member of class BasketCalculatorData.

## BasketCalculatorData.BasketCalculatorData Constructor (BasketData)

**C#**

```
public BasketCalculatorData(BasketData basket);
```

**Visual Basic**

```
Public Sub New(basket As BasketData)
```

**Description**

This is BasketCalculatorData, a member of class BasketCalculatorData.

# BasketCalculatorData Properties

The properties of the BasketCalculatorData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AdjustedItems (see page 1221) | Gets the list of adjusted basket items in the current basket. |
| | Basket (see page 1221) | Gets or sets the Basket being calculated by the BasketCalculator. |
| | BasketCoupons (see page 1222) | Gets or sets BasketCoupons. These are all coupons that have been applied to the basket and their cumulative discount. If a coupon applies to more than one item, the discount amount for the coupon would be the total amount. |
| | BasketTotal (see page 1222) | Gets the basket total based upon item prices, coupons, taxes, shipping and any other adjustments. |
| | HasFreeShipping (see page 1222) | Gets or sets whether or not this basket has afree shipping coupon applied. |
| | ShipFrom (see page 1222) | Gets or sets the address from which the order would ship. This could be used for shipping, tax calculations, etc. If it's ommited, the shipping and tax information will not be calculated. |
| | ShippingRates (see page 1222) | Gets a list of shipping rates for the basket. |
| | ShipTo (see page 1222) | Gets or sets the address that the order is shipping to. This could be used for shipping, tax calculations, etc. If it's ommited, the shipping and tax information will not be calculated. |
| | TaxClassSubtotals (see page 1223) | Gets a dictionary representing the tax classes applicable to the basket with the amount of taxes applied to the basket for that tax class. |
| | TotalCouponDiscount (see page 1223) | Gets the total amount of coupons calculated for the current basket |
| | TotalShippingCost (see page 1223) | Gets the total amount of shipping costs calculated for the current basket. |
| | TotalTaxes (see page 1223) | Gets the total amount of taxes calculated for the current basket. |
| | ValidationResults (see page 1223) | Gets any ValidationResults encountered during calculation. |

## BasketCalculatorData.AdjustedItems Property

Gets the list of adjusted basket items in the current basket.

**C#**

```
public List<AdjustedBasketItem> AdjustedItems;
```

**Visual Basic**

```
Public Property AdjustedItems() As List<AdjustedBasketItem>
```

## BasketCalculatorData.Basket Property

Gets or sets the Basket being calculated by the BasketCalculator.

**C#**

```
public BasketData Basket;
```

**Visual Basic**

```
Public Property Basket() As BasketData
```

## BasketCalculatorData.BasketCoupons Property

Gets or sets BasketCoupons. These are all coupons that have been applied to the basket and their cumulative discount. If a coupon applies to more than one item, the discount amount for the coupon would be the total amount.

**C#**

```
public AppliedCouponDataCollection BasketCoupons;
```

**Visual Basic**

```
Public Property BasketCoupons() As AppliedCouponDataCollection
```

## BasketCalculatorData.BasketTotal Property

Gets the basket total based upon item prices, coupons, taxes, shipping and any other adjustments.

**C#**

```
public decimal BasketTotal;
```

**Visual Basic**

```
Public Property BasketTotal() As decimal
```

## BasketCalculatorData.HasFreeShipping Property

Gets or sets whether or not this basket has afree shipping coupon applied.

**C#**

```
public bool HasFreeShipping;
```

**Visual Basic**

```
Public Property HasFreeShipping() As bool
```

## BasketCalculatorData.ShipFrom Property

Gets or sets the address from which the order would ship. This could be used for shipping, tax calculations, etc. If it's ommited, the shipping and tax information will not be calculated.

**C#**

```
public AddressData ShipFrom;
```

**Visual Basic**

```
Public Property ShipFrom() As AddressData
```

## BasketCalculatorData.ShippingRates Property

Gets a list of shipping rates for the basket.

**C#**

```
public List<ShippingOptionData> ShippingRates;
```

**Visual Basic**

```
Public Property ShippingRates() As List<ShippingOptionData>
```

## BasketCalculatorData.ShipTo Property

Gets or sets the address that the order is shipping to. This could be used for shipping, tax calculations, etc. If it's ommited, the

shipping and tax information will not be calculated.

**C#**

```
public AddressData ShipTo;
```

**Visual Basic**

```
Public Property ShipTo() As AddressData
```

## BasketCalculatorData.TaxClassSubtotals Property

Gets a dictionary representing the tax classes applicable to the basket with the amount of taxes applied to the basket for that tax class.

**C#**

```
public Dictionary<TaxClassData, decimal> TaxClassSubtotals;
```

**Visual Basic**

```
Public Property TaxClassSubtotals() As Dictionary<TaxClassData, decimal>
```

## BasketCalculatorData.TotalCouponDiscount Property

Gets the total amount of coupons calculated for the current basket

**C#**

```
public decimal TotalCouponDiscount;
```

**Visual Basic**

```
Public Property TotalCouponDiscount() As decimal
```

## BasketCalculatorData.TotalShippingCost Property

Gets the total amount of shipping costs calculated for the current basket.

**C#**

```
public decimal TotalShippingCost;
```

**Visual Basic**

```
Public Property TotalShippingCost() As decimal
```

## BasketCalculatorData.TotalTaxes Property

Gets the total amount of taxes calculated for the current basket.

**C#**

```
public decimal TotalTaxes;
```

**Visual Basic**

```
Public Property TotalTaxes() As decimal
```

## BasketCalculatorData.ValidationResults Property

Gets any ValidationResults encountered during calculation.

**C#**

```
public ValidationResults ValidationResults;
```

**Visual Basic**

```
Public Property ValidationResults() As ValidationResults
```

# BasketData Class

### Class Hierarchy

### C#

```
public class BasketData;
```

### Visual Basic

```
Public Class BasketData
```

### File

BasketData.cs

### Description

This is class Ektron.Cms.Commerce.BasketData.

## BasketData Properties

The properties of the BasketData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | DateCreated (see page 1224) | Gets or sets the date created. |
| | DateModified (see page 1224) | Gets or sets the date modified. |
| | Id (see page 1225) | Gets or sets the Id of the Basket (see page 1215). |
| | IsDefault (see page 1225) | Gets or sets value indicating whether or not this basket is the default basket for customers. |
| | Items (see page 1225) | Gets the collection items in the basket. |
| | Name (see page 1225) | Gets or sets the name of the basket. |
| | ShippingAddressId (see page 1225) | Gets or sets the Id (see page 1225) of the address this basket will be shipping to. This is used for calculating taxes, shipping, etc. |
| | ShippingMethodId (see page 1226) | Gets or sets the id of the shipping method to be used for the basket. This is used for calculating shipping costs. |
| | ShopperId (see page 1226) | Gets or sets the Shoppervisitor Id (see page 1225) using the basket. This Id (see page 1225) is used when customer is not logged in. |
| | Subtotal (see page 1226) | Gets the basket subtotal. |
| | UserId (see page 1226) | Gets or sets the Id (see page 1225) of the user asscoiated with the basket. |

## BasketData.DateCreated Property

Gets or sets the date created.

### C#

```
public DateTime DateCreated;
```

### Visual Basic

```
Public Property DateCreated() As DateTime
```

## BasketData.DateModified Property

Gets or sets the date modified.

**C#**

```
public DateTime DateModified;
```

**Visual Basic**

```
Public Property DateModified() As DateTime
```

## BasketData.Id Property

Gets or sets the Id of the Basket (see page 1215).

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## BasketData.IsDefault Property

Gets or sets value indicating whether or not this basket is the default basket for customers.

**C#**

```
public bool IsDefault;
```

**Visual Basic**

```
Public Property IsDefault() As bool
```

## BasketData.Items Property

Gets the collection items in the basket.

**C#**

```
public BasketItemCollection Items;
```

**Visual Basic**

```
Public Property Items() As BasketItemCollection
```

## BasketData.Name Property

Gets or sets the name of the basket.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

## BasketData.ShippingAddressId Property

Gets or sets the Id (see page 1225) of the address this basket will be shipping to. This is used for calculating taxes, shipping, etc.

**C#**

```
public long ShippingAddressId;
```

**Visual Basic**

```
Public Property ShippingAddressId() As long
```

## BasketData.ShippingMethodId Property

Gets or sets the id of the shipping method to be used for the basket. This is used for calculating shipping costs.

**C#**

```
public long ShippingMethodId;
```

**Visual Basic**

```
Public Property ShippingMethodId() As long
```

## BasketData.ShopperId Property

Gets or sets the Shoppervisitor Id (see page 1225) using the basket. This Id (see page 1225) is used when customer is not logged in.

**C#**

```
public string ShopperId;
```

**Visual Basic**

```
Public Property ShopperId() As string
```

## BasketData.Subtotal Property

Gets the basket subtotal.

**C#**

```
public decimal Subtotal;
```

**Visual Basic**

```
Public Property Subtotal() As decimal
```

## BasketData.UserId Property

Gets or sets the Id (see page 1225) of the user asscoiated with the basket.

**C#**

```
public long UserId;
```

**Visual Basic**

```
Public Property UserId() As long
```

# BasketItem Class

Data class representing a basket line item.

**Class Hierarchy**

**C#**

```
public class BasketItem;
```

**Visual Basic**

```
Public Class BasketItem
```

**File**

BasketItem.cs

# BasketItem Constructor

This is the overview for the BasketItem constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | BasketItem.BasketItem () (see page 1227) | This is BasketItem, a member of class BasketItem. |
| | BasketItem.BasketItem (BasketItem) (see page 1227) | constructor used for cloning Existing BasketItem. |

## BasketItem.BasketItem Constructor ()

**C#**

```
public BasketItem();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is BasketItem, a member of class BasketItem.

## BasketItem.BasketItem Constructor (BasketItem)

constructor used for cloning Existing BasketItem.

**C#**

```
public BasketItem(BasketItem basketItem);
```

**Visual Basic**

```
Public Sub New(basketItem As BasketItem)
```

**Parameters**

| Parameters | Description |
|---|---|
| basketItem | |

# BasketItem Properties

The properties of the BasketItem class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AdjustedTotal (see page 1228) | Gets or sets the Total Adjusted price for the line item. |
| | Configuration (see page 1228) | Gets or sets any applicable Kit configuarations for the product in the basket for this line item. |
| | Dimensions (see page 1228) | Gets or sets the dimensions of the product. Used for shipping. |
| | Id (see page 1228) | Gets or sets the Id of this line item. |
| | IsTangible (see page 1228) | Indicates if this product is a tangible good. |
| | ListPrice (see page 1229) | Gets or sets the list price for the line item. |
| | Pricing (see page 1229) | The item pricing, including currencies and quantity. |

| | | |
|---|---|---|
| | ProductId (see page 1229) | Gets or sets the Product (see page 412) Id (see page 1228) associated with this this line item. |
| | ProductLanguage (see page 1229) | Gets or sets the language for the product in the basket for this line item. |
| | ProductSku (see page 1229) | Gets or sets the Sku for the product in the basket for this line item. |
| | ProductTitle (see page 1230) | Gets or sets the Product (see page 412) Title for the product in the basket for this line item. |
| | ProductType (see page 1230) | Gets or sets the type of the product in the basket for this line item. |
| | Quantity (see page 1230) | Gets or sets the quantity of the product in the basket for this line item. |
| | Quicklink (see page 1230) | Gets or sets the product's Quicklink. |
| | SalePrice (see page 1230) | Sets the base sale price for the line item. The returned sale price will reflect any adjustments based upon kit options or product variants selected. |
| | TaxClassId (see page 1231) | Gets or sets the tax class Id (see page 1228) for the product in the basket for this line item. |
| | Variant (see page 1231) | Gets or sets and applicable varioant information for the product in the basket for this line item. |
| | Weight (see page 1231) | Gets or sets the weight of the product. Used for shipping. |

## BasketItem.AdjustedTotal Property

Gets or sets the Total Adjusted price for the line item.

**C#**

```
public virtual decimal AdjustedTotal;
```

**Visual Basic**

```
Public virtual Property AdjustedTotal() As decimal
```

## BasketItem.Configuration Property

Gets or sets any applicable Kit configuarations for the product in the basket for this line item.

**C#**

```
public KitConfigData Configuration;
```

**Visual Basic**

```
Public Property Configuration() As KitConfigData
```

## BasketItem.Dimensions Property

Gets or sets the dimensions of the product. Used for shipping.

**C#**

```
public Dimensions Dimensions;
```

**Visual Basic**

```
Public Property Dimensions() As Dimensions
```

## BasketItem.Id Property

Gets or sets the Id of this line item.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## BasketItem.IsTangible Property

Indicates if this product is a tangible good.

**C#**

```
public bool IsTangible;
```

**Visual Basic**

```
Public Property IsTangible() As bool
```

**Returns**

Boolean value indicating if this product is a tangible good.

## BasketItem.ListPrice Property

Gets or sets the list price for the line item.

**C#**

```
public decimal ListPrice;
```

**Visual Basic**

```
Public Property ListPrice() As decimal
```

## BasketItem.Pricing Property

The item pricing, including currencies and quantity.

**C#**

```
public PricingData Pricing;
```

**Visual Basic**

```
Public Property Pricing() As PricingData
```

**Returns**

PricingData (see page 1316) of the basket item.

## BasketItem.ProductId Property

Gets or sets the Product (see page 412) Id (see page 1228) associated with this this line item.

**C#**

```
public long ProductId;
```

**Visual Basic**

```
Public Property ProductId() As long
```

## BasketItem.ProductLanguage Property

Gets or sets the language for the product in the basket for this line item.

**C#**

```
public long ProductLanguage;
```

**Visual Basic**

```
Public Property ProductLanguage() As long
```

## BasketItem.ProductSku Property

Gets or sets the Sku for the product in the basket for this line item.

**C#**

```
public string ProductSku;
```

**Visual Basic**

```
Public Property ProductSku() As string
```

## BasketItem.ProductTitle Property

Gets or sets the Product (see page 412) Title for the product in the basket for this line item.

**C#**

```
public string ProductTitle;
```

**Visual Basic**

```
Public Property ProductTitle() As string
```

## BasketItem.ProductType Property

Gets or sets the type of the product in the basket for this line item.

**C#**

```
public EkEnumeration.CatalogEntryType ProductType;
```

**Visual Basic**

```
Public Property ProductType() As EkEnumeration.CatalogEntryType
```

## BasketItem.Quantity Property

Gets or sets the quantity of the product in the basket for this line item.

**C#**

```
public int Quantity;
```

**Visual Basic**

```
Public Property Quantity() As Integer
```

## BasketItem.Quicklink Property

Gets or sets the product's Quicklink.

**C#**

```
public string Quicklink;
```

**Visual Basic**

```
Public Property Quicklink() As string
```

## BasketItem.SalePrice Property

Sets the base sale price for the line item. The returned sale price will reflect any adjustments based upon kit options or product variants selected.

**C#**

```
public decimal SalePrice;
```

**Visual Basic**

```
Public Property SalePrice() As decimal
```

## BasketItem.TaxClassId Property

Gets or sets the tax class Id (see page 1228) for the product in the basket for this line item.

**C#**

```
public long TaxClassId;
```

**Visual Basic**

```
Public Property TaxClassId() As long
```

## BasketItem.Variant Property

Gets or sets and applicable varioant information for the product in the basket for this line item.

**C#**

```
public BasketVariantData Variant;
```

**Visual Basic**

```
Public Property Variant() As BasketVariantData
```

## BasketItem.Weight Property

Gets or sets the weight of the product. Used for shipping.

**C#**

```
public Weight Weight;
```

**Visual Basic**

```
Public Property Weight() As Weight
```

# BasketItemCollection Class

Collection (see page 88) class for BasketItems.

**Class Hierarchy**

**C#**

```
public class BasketItemCollection : Collection<BasketItem>;
```

**Visual Basic**

```
Public Class BasketItemCollection
Inherits Collection
```

**File**

BasketItem.cs

# BasketItemCollection.BasketItemCollection Constructor

**C#**

```
public BasketItemCollection();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is BasketItemCollection, a member of class BasketItemCollection.

# BasketItemCollection Methods

The methods of the BasketItemCollection class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | AddRange (see page 1232) | This is AddRange, a member of class BasketItemCollection. |
| | Find (see page 1232) | This is Find, a member of class BasketItemCollection. |
| | FindAll (see page 1232) | This is FindAll, a member of class BasketItemCollection. |
| | FindIndex (see page 1232) | This is FindIndex, a member of class BasketItemCollection. |

## BasketItemCollection.AddRange Method

**C#**

```
public void AddRange(IEnumerable<BasketItem> collection);
```

**Visual Basic**

```
Public Function AddRange(collection As IEnumerable<BasketItem>) As void
```

**Description**

This is AddRange, a member of class BasketItemCollection.

## BasketItemCollection.Find Method

**C#**

```
public BasketItem Find(Predicate<BasketItem> match);
```

**Visual Basic**

```
Public Function Find(match As Predicate<BasketItem>) As BasketItem
```

**Description**

This is Find, a member of class BasketItemCollection.

## BasketItemCollection.FindAll Method

**C#**

```
public List<BasketItem> FindAll(Predicate<BasketItem> match);
```

**Visual Basic**

```
Public Function FindAll(match As Predicate<BasketItem>) As List<BasketItem>
```

**Description**

This is FindAll, a member of class BasketItemCollection.

## BasketItemCollection.FindIndex Method

**C#**

```
public int FindIndex(Predicate<BasketItem> match);
```

**Visual Basic**

```
Public Function FindIndex(match As Predicate<BasketItem>) As Integer
```

**Description**

This is FindIndex, a member of class BasketItemCollection.

# BasketVariantData Class

**Class Hierarchy**

**C#**

```
public class BasketVariantData;
```

**Visual Basic**

```
Public Class BasketVariantData
```

**File**

VariantData.cs

## BasketVariantData.BasketVariantData Constructor

**C#**

```
public BasketVariantData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is BasketVariantData, a member of class BasketVariantData.

## BasketVariantData Properties

The properties of the BasketVariantData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Dimensions (see page 1233) | Gets or sets the dimensions of the product. Used for shipping. |
| | Id (see page 1234) | This is Id, a member of class BasketVariantData. |
| | Language (see page 1234) | This is Language, a member of class BasketVariantData. |
| | ListPrice (see page 1234) | This is ListPrice, a member of class BasketVariantData. |
| | Pricing (see page 1234) | The variant pricing, including currencies and quantity. |
| | SalePrice (see page 1235) | This is SalePrice, a member of class BasketVariantData. |
| | Sku (see page 1235) | This is Sku, a member of class BasketVariantData. |
| | TaxClassId (see page 1235) | Gets or set the TaxClassId of the Variant. |
| | Title (see page 1235) | This is Title, a member of class BasketVariantData. |
| | Weight (see page 1235) | Gets or sets the weight of the product. Used for shipping. |

### BasketVariantData.Dimensions Property

Gets or sets the dimensions of the product. Used for shipping.

**C#**

```
public Dimensions Dimensions;
```

**Visual Basic**

```
Public Property Dimensions() As Dimensions
```

## BasketVariantData.Id Property

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Description**

This is Id, a member of class BasketVariantData.

## BasketVariantData.Language Property

**C#**

```
public long Language;
```

**Visual Basic**

```
Public Property Language() As long
```

**Description**

This is Language, a member of class BasketVariantData.

## BasketVariantData.ListPrice Property

**C#**

```
public decimal ListPrice;
```

**Visual Basic**

```
Public Property ListPrice() As decimal
```

**Description**

This is ListPrice, a member of class BasketVariantData.

## BasketVariantData.Pricing Property

The variant pricing, including currencies and quantity.

**C#**

```
public PricingData Pricing;
```

**Visual Basic**

```
Public Property Pricing() As PricingData
```

**Returns**

PricingData (see page 1316) of the variant.

## BasketVariantData.SalePrice Property

**C#**

```
public decimal SalePrice;
```

**Visual Basic**

```
Public Property SalePrice() As decimal
```

**Description**

This is SalePrice, a member of class BasketVariantData.

## BasketVariantData.Sku Property

**C#**

```
public string Sku;
```

**Visual Basic**

```
Public Property Sku() As string
```

**Description**

This is Sku, a member of class BasketVariantData.

## BasketVariantData.TaxClassId Property

Gets or set the TaxClassId of the Variant.

**C#**

```
public long TaxClassId;
```

**Visual Basic**

```
Public Property TaxClassId() As long
```

## BasketVariantData.Title Property

**C#**

```
public string Title;
```

**Visual Basic**

```
Public Property Title() As string
```

**Description**

This is Title, a member of class BasketVariantData.

## BasketVariantData.Weight Property

Gets or sets the weight of the product. Used for shipping.

**C#**

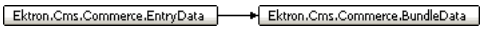
```
public Weight Weight;
```

**Visual Basic**

```
Public Property Weight() As Weight
```

# BundleData Class

### Class Hierarchy




### C#

```
public class BundleData : EntryData;
```

### Visual Basic

```
Public Class BundleData
Inherits EntryData
```

### File

BundleData.cs

# BundleData.BundleData Constructor

### C#

```
public BundleData();
```

### Visual Basic

```
Public Sub New()
```

### Description

This is BundleData, a member of class BundleData.

# BundleData Properties

The properties of the BundleData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | BundledItems (see page 1236) | This is BundledItems, a member of class BundleData. |

## BundleData.BundledItems Property

### C#

```
public List<EntryData> BundledItems;
```

### Visual Basic

```
Public Property BundledItems() As List<EntryData>
```

### Description

This is BundledItems, a member of class BundleData.

# CatalogData Class

### Class Hierarchy

**C#**

```
public class CatalogData;
```

**Visual Basic**

```
Public Class CatalogData
```

**File**

CatalogData.cs

## CatalogData Properties

The properties of the CatalogData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Id (see page 1237) | This is Id, a member of class CatalogData. |
| | Name (see page 1237) | This is Name, a member of class CatalogData. |

### CatalogData.Id Property

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Description**

This is Id, a member of class CatalogData.

### CatalogData.Name Property

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

**Description**

This is Name, a member of class CatalogData.

# ContractData Class

**Class Hierarchy**

**C#**

```
public class ContractData;
```

**Visual Basic**

```
Public Class ContractData
```

**File**

ContractData.cs

**Description**

This is class Ektron.Cms.Commerce.ContractData.

# ContractData Constructor

This is the overview for the ContractData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | ContractData.ContractData () (see page 1238) | This is ContractData, a member of class ContractData. |
| | ContractData.ContractData (string) (see page 1238) | This is ContractData, a member of class ContractData. |

## ContractData.ContractData Constructor ()

**C#**

```
public ContractData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is ContractData, a member of class ContractData.

## ContractData.ContractData Constructor (string)

**C#**

```
public ContractData(string contractName);
```

**Visual Basic**

```
Public Sub New(contractName As string)
```

**Description**

This is ContractData, a member of class ContractData.

# ContractData Properties

The properties of the ContractData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Id (see page 1238) | Gets or sets the Id of the contract. |
| | Name (see page 1239) | Gets or sets the contract name. |

## ContractData.Id Property

Gets or sets the Id of the contract.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

### ContractData.Name Property

Gets or sets the contract name.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

## CountryData Class

**Class Hierarchy**

**C#**

```
public class CountryData;
```

**Visual Basic**

```
Public Class CountryData
```

**File**

CountryData.cs

**Description**

This is class Ektron.Cms.Commerce.CountryData.

## CountryData Constructor

This is the overview for the CountryData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | CountryData.CountryData () (see page 1239) | This is CountryData, a member of class CountryData. |
| | CountryData.CountryData (int, string, string, string, bool) (see page 1240) | |

### CountryData.CountryData Constructor ()

**C#**

```
public CountryData();
```

**Visual Basic**

```
Public Sub New()
```

### Description

This is CountryData, a member of class CountryData.

## CountryData.CountryData Constructor (int, string, string, string, bool)

### C#

```
public CountryData(int id, string name, string shortIsoCode, string longIsoCode, bool enabled);
```

### Visual Basic

```
Public Sub New(id As Integer, name As string, shortIsoCode As string, longIsoCode As string,
enabled As bool)
```

### Parameters

| Parameters | Description |
|---|---|
| id | Id (see page 1241) of the country. The Id (see page 1241) is the ISO numeric code of the country |
| name | name of the country |
| shortIsoCode | 2 letter ISO code of the country |
| longIsoCode | 3 letter ISO code of the country |
| enabled | indicates whether or not the country is enabled for use |

# CountryData Methods

The methods of the CountryData class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | ToString (see page 1240) | Returns Name (see page 1241) of Country. |
| | Validate (see page 1240) | Validates the CountryData (see page 1239) object is valid for saving. |

## CountryData.ToString Method

Returns Name (see page 1241) of Country.

### C#

```
public override string ToString();
```

### Visual Basic

```
Public override Function ToString() As string
```

## CountryData.Validate Method

Validates the CountryData (see page 1239) object is valid for saving.

### C#

```
public ValidationResults Validate();
```

### Visual Basic

```
Public Function Validate() As ValidationResults
```

### Returns

Returns a ValidationResult object with any validation errors.

# CountryData Properties

The properties of the CountryData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Enabled (see page 1241) | Gets or sets whether or not the country is enabled for use. |
| | Id (see page 1241) | Gets or sets the Id of the country. The Id is the ISO numeric code of the country. |
| | LongIsoCode (see page 1241) | Gets or sets the 3 letter ISO code of the country. |
| | Name (see page 1241) | Gets or sets the name of the country. |
| | ShortIsoCode (see page 1241) | Gets or sets the 2 letter ISO code of the country. |

## CountryData.Enabled Property

Gets or sets whether or not the country is enabled for use.

**C#**

```
public bool Enabled;
```

**Visual Basic**

```
Public Property Enabled() As bool
```

## CountryData.Id Property

Gets or sets the Id of the country. The Id is the ISO numeric code of the country.

**C#**

```
public int Id;
```

**Visual Basic**

```
Public Property Id() As Integer
```

## CountryData.LongIsoCode Property

Gets or sets the 3 letter ISO code of the country.

**C#**

```
public string LongIsoCode;
```

**Visual Basic**

```
Public Property LongIsoCode() As string
```

## CountryData.Name Property

Gets or sets the name of the country.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

## CountryData.ShortIsoCode Property

Gets or sets the 2 letter ISO code of the country.

**C#**

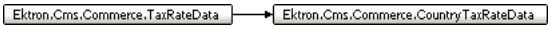
```
public string ShortIsoCode;
```

**Visual Basic**

```
Public Property ShortIsoCode() As string
```

# CountryTaxRateData Class

**Class Hierarchy**




**C#**

```
public class CountryTaxRateData : TaxRateData;
```

**Visual Basic**

```
Public Class CountryTaxRateData
Inherits TaxRateData
```

**File**

CountryTaxRateData.cs

**Description**

This is class Ektron.Cms.Commerce.CountryTaxRateData.

# CountryTaxRateData Constructor

This is the overview for the CountryTaxRateData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | CountryTaxRateData.CountryTaxRateData () (see page 1242) | This is CountryTaxRateData, a member of class CountryTaxRateData. |
| | CountryTaxRateData.CountryTaxRateData (long, long, decimal) (see page 1243) | This is CountryTaxRateData, a member of class CountryTaxRateData. |
| | CountryTaxRateData.CountryTaxRateData (long, long, decimal, long) (see page 1243) | This is CountryTaxRateData, a member of class CountryTaxRateData. |

## CountryTaxRateData.CountryTaxRateData Constructor ()

**C#**

```
public CountryTaxRateData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is CountryTaxRateData, a member of class CountryTaxRateData.

## CountryTaxRateData.CountryTaxRateData Constructor (long, long, decimal)

**C#**

```
public CountryTaxRateData(long countryId, long taxClassId, decimal rate);
```

**Visual Basic**

```
Public Sub New(countryId As long, taxClassId As long, rate As decimal)
```

**Description**

This is CountryTaxRateData, a member of class CountryTaxRateData.

## CountryTaxRateData.CountryTaxRateData Constructor (long, long, decimal, long)

**C#**

```
public CountryTaxRateData(long countryId, long taxClassId, decimal rate, long id);
```

**Visual Basic**

```
Public Sub New(countryId As long, taxClassId As long, rate As decimal, id As long)
```

**Description**

This is CountryTaxRateData, a member of class CountryTaxRateData.

# CountryTaxRateData Properties

The properties of the CountryTaxRateData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CountryData (see page 1243) | Gets or sets the CountryData object associated with this tax rate. |

## CountryTaxRateData.CountryData Property

Gets or sets the CountryData object associated with this tax rate.

**C#**

```
public CountryData CountryData;
```

**Visual Basic**

```
Public Property CountryData() As CountryData
```

# CouponData Class

**Class Hierarchy**

Ektron.Cms.Commerce.CouponData

**C#**

```
public class CouponData;
```

**Visual Basic**

```
Public Class CouponData
```

**File**

CouponData.cs

**Description**

This is class Ektron.Cms.Commerce.CouponData.

# CouponData.CouponData Constructor

**C#**

```
public CouponData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is CouponData, a member of class CouponData.

# CouponData Properties

The properties of the CouponData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Code (see page 1244) | Gets or sets the coupon code. |
| | CouponType (see page 1245) | Gets or sets the coupon type. |
| | CurrencyId (see page 1245) | Gets or sets the Id (see page 1246) of the currency this coupon applies to. |
| | Description (see page 1245) | Gets or sets the description of the coupon. |
| | DiscountType (see page 1245) | Gets or sets the Coupon Discount Type. |
| | DiscountValue (see page 1245) | Gets or sets the value of the discount. For a fixed amount, this would be the amount (i.e. 5.00). For percentage, it would be the discount percent value in whole numbers (i.e. 15% = 15.00). |
| | ExpirationDate (see page 1245) | Gets or sets the date the coupon expires. |
| | Id (see page 1246) | Gets or sets the Id of the coupon. |
| | IsActive (see page 1246) | Determines whether or not this coupon is currently active. Deleted coupons remain in the system and are marked inactive. |
| | IsCombinable (see page 1246) | Determins whether or not this coupon can be combined with other coupons. If true, this coupon can be combined with other "Combinable" coupons. |
| | MaximumAmount (see page 1246) | Gets or sets the maximum order amount that this coupon can be applied to. |
| | MaximumUses (see page 1246) | Gets or sets the maximim number of times this coupon can be used before it expires. |
| | MinimumAmount (see page 1247) | Gets or sets the minimum order amount required before the coupon is applicable. |
| | OnePerCustomer (see page 1247) | Gets or sets the OnePerCuster value. If true, this coupon can only be used once by each customer. |
| | StartDate (see page 1247) | Gets or sets the date the coupon becomes valid. |

## CouponData.Code Property

Gets or sets the coupon code.

**C#**

```
public string Code;
```

**Visual Basic**

```
Public Property Code() As string
```

## CouponData.CouponType Property

Gets or sets the coupon type.

**C#**

```
public EkEnumeration.CouponType CouponType;
```

**Visual Basic**

```
Public Property CouponType() As EkEnumeration.CouponType
```

## CouponData.CurrencyId Property

Gets or sets the Id (see page 1246) of the currency this coupon applies to.

**C#**

```
public int CurrencyId;
```

**Visual Basic**

```
Public Property CurrencyId() As Integer
```

## CouponData.Description Property

Gets or sets the description of the coupon.

**C#**

```
public string Description;
```

**Visual Basic**

```
Public Property Description() As string
```

## CouponData.DiscountType Property

Gets or sets the Coupon Discount Type.

**C#**

```
public EkEnumeration.CouponDiscountType DiscountType;
```

**Visual Basic**

```
Public Property DiscountType() As EkEnumeration.CouponDiscountType
```

## CouponData.DiscountValue Property

Gets or sets the value of the discount. For a fixed amount, this would be the amount (i.e. 5.00). For percentage, it would be the discount percent value in whole numbers (i.e. 15% = 15.00).

**C#**

```
public decimal DiscountValue;
```

**Visual Basic**

```
Public Property DiscountValue() As decimal
```

## CouponData.ExpirationDate Property

Gets or sets the date the coupon expires.

**C#**

```
public DateTime ExpirationDate;
```

**Visual Basic**

```
Public Property ExpirationDate() As DateTime
```

## CouponData.Id Property

Gets or sets the Id of the coupon.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## CouponData.IsActive Property

Determines whether or not this coupon is currently active. Deleted coupons remain in the system and are marked inactive.

**C#**

```
public bool IsActive;
```

**Visual Basic**

```
Public Property IsActive() As bool
```

## CouponData.IsCombinable Property

Determins whether or not this coupon can be combined with other coupons. If true, this coupon can be combined with other "Combinable" coupons.

**C#**

```
public bool IsCombinable;
```

**Visual Basic**

```
Public Property IsCombinable() As bool
```

## CouponData.MaximumAmount Property

Gets or sets the maximum order amount that this coupon can be applied to.

**C#**

```
public decimal MaximumAmount;
```

**Visual Basic**

```
Public Property MaximumAmount() As decimal
```

## CouponData.MaximumUses Property

Gets or sets the maximim number of times this coupon can be used before it expires.

**C#**

```
public int MaximumUses;
```

**Visual Basic**

```
Public Property MaximumUses() As Integer
```

## CouponData.MinimumAmount Property

Gets or sets the minimum order amount required before the coupon is applicable.

**C#**

```
public decimal MinimumAmount;
```

**Visual Basic**

```
Public Property MinimumAmount() As decimal
```

## CouponData.OnePerCustomer Property

Gets or sets the OnePerCuster value. If true, this coupon can only be used once by each customer.

**C#**

```
public bool OnePerCustomer;
```

**Visual Basic**

```
Public Property OnePerCustomer() As bool
```

## CouponData.StartDate Property

Gets or sets the date the coupon becomes valid.

**C#**

```
public DateTime StartDate;
```

**Visual Basic**

```
Public Property StartDate() As DateTime
```

# CouponEntryData Class

**Class Hierarchy**

**C#**

```
public class CouponEntryData;
```

**Visual Basic**

```
Public Class CouponEntryData
```

**File**

CouponEntryData.cs

**Description**

This is class Ektron.Cms.Commerce.CouponEntryData.

# CouponEntryData Constructor

This is the overview for the CouponEntryData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | CouponEntryData.CouponEntryData () (see page 1248) | This is CouponEntryData, a member of class CouponEntryData. |
| | CouponEntryData.CouponEntryData (long, EkEnumeration.CMSObjectTypes, long, string, bool) (see page 1248) | This is CouponEntryData, a member of class CouponEntryData. |

## CouponEntryData.CouponEntryData Constructor ()

**C#**

```
public CouponEntryData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is CouponEntryData, a member of class CouponEntryData.

## CouponEntryData.CouponEntryData Constructor (long, EkEnumeration.CMSObjectTypes, long, string, bool)

**C#**

```
public CouponEntryData(long objectId, EkEnumeration.CMSObjectTypes objectType, long couponId,
string title, bool isIncluded);
```

**Visual Basic**

```
Public Sub New(objectId As long, objectType As EkEnumeration.CMSObjectTypes, couponId As long,
title As string, isIncluded As bool)
```

**Description**

This is CouponEntryData, a member of class CouponEntryData.

# CouponEntryData Properties

The properties of the CouponEntryData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CouponId (see page 1248) | Gets or sets the Id of the coupon |
| | DataState (see page 1249) | Gets or sets the current saved state for the CouponEntry. Used by the Cms (see page 7) internally. |
| | IsIncluded (see page 1249) | Gets or sets whether or not the Object is included in the coupon. If true, the object is included in the coupon. If false, the coupon is NOT included in the coupon. |
| | ObjectId (see page 1249) | Gets or sets the Id of object the coupon is applied to. |
| | ObjectType (see page 1249) | Gets or sets the type of object this coupon is applied to. |
| | Path (see page 1249) | Gets or sets the Path of the object. |
| | Title (see page 1250) | Gets or sets the Title of the object. |

## CouponEntryData.CouponId Property

Gets or sets the Id of the coupon

**C#**

```
public long CouponId;
```

**Visual Basic**

```
Public Property CouponId() As long
```

## CouponEntryData.DataState Property

Gets or sets the current saved state for the CouponEntry. Used by the Cms (see page 7) internally.

**C#**

```
public EkEnumeration.DataState DataState;
```

**Visual Basic**

```
Public Property DataState() As EkEnumeration.DataState
```

## CouponEntryData.IsIncluded Property

Gets or sets whether or not the Object is included in the coupon. If true, the object is included in the coupon. If false, the coupon is NOT included in the coupon.

**C#**

```
public bool IsIncluded;
```

**Visual Basic**

```
Public Property IsIncluded() As bool
```

## CouponEntryData.ObjectId Property

Gets or sets the Id of object the coupon is applied to.

**C#**

```
public long ObjectId;
```

**Visual Basic**

```
Public Property ObjectId() As long
```

## CouponEntryData.ObjectType Property

Gets or sets the type of object this coupon is applied to.

**C#**

```
public EkEnumeration.CMSObjectTypes ObjectType;
```

**Visual Basic**

```
Public Property ObjectType() As EkEnumeration.CMSObjectTypes
```

## CouponEntryData.Path Property

Gets or sets the Path of the object.

**C#**

```
public string Path;
```

**Visual Basic**

```
Public Property Path() As string
```

## CouponEntryData.Title Property

Gets or sets the Title of the object.

**C#**

```
public string Title;
```

**Visual Basic**

```
Public Property Title() As string
```

# CurrencyData Class

Data class representing currencies in system.

**Class Hierarchy**

**C#**

```
public class CurrencyData;
```

**Visual Basic**

```
Public Class CurrencyData
```

**File**

CurrencyData.cs

# CurrencyData Properties

The properties of the CurrencyData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AlphaIsoCode (see page 1250) | Gets or sets the 3 letter ISO code of the currency. |
| | CultureCode (see page 1251) | Gets or sets the culture code associated with this currency. This is used for formatting currency values for this currency. |
| | CurrencySymbol (see page 1251) | Read only function that gets the symbol for this currency. |
| | Enabled (see page 1251) | Gets or sets whether or not the currency is enabled for use. |
| | Id (see page 1251) | Gets or sets the Id of the currency. The Id is the ISO numeric code of the currency. |
| | ISOCurrencySymbol (see page 1251) | Read only function that gets the ISO symbol for this currency. |
| | Name (see page 1252) | Gets or sets the name of the currency. |

## CurrencyData.AlphaIsoCode Property

Gets or sets the 3 letter ISO code of the currency.

**C#**

```
public string AlphaIsoCode;
```

**Visual Basic**

```
Public Property AlphaIsoCode() As string
```

## CurrencyData.CultureCode Property

Gets or sets the culture code associated with this currency. This is used for formatting currency values for this currency.

**C#**

```
public string CultureCode;
```

**Visual Basic**

```
Public Property CultureCode() As string
```

## CurrencyData.CurrencySymbol Property

Read only function that gets the symbol for this currency.

**C#**

```
public string CurrencySymbol;
```

**Visual Basic**

```
Public Property CurrencySymbol() As string
```

**Example**

$

## CurrencyData.Enabled Property

Gets or sets whether or not the currency is enabled for use.

**C#**

```
public bool Enabled;
```

**Visual Basic**

```
Public Property Enabled() As bool
```

## CurrencyData.Id Property

Gets or sets the Id of the currency. The Id is the ISO numeric code of the currency.

**C#**

```
public int Id;
```

**Visual Basic**

```
Public Property Id() As Integer
```

## CurrencyData.ISOCurrencySymbol Property

Read only function that gets the ISO symbol for this currency.

**C#**

```
public string ISOCurrencySymbol;
```

**Visual Basic**

```
Public Property ISOCurrencySymbol() As string
```

**Example**

USD

## CurrencyData.Name Property

Gets or sets the name of the currency.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

# CurrencyPricingData Class

Represents Pricing Information for agiven currency.

**Class Hierarchy**

**C#**

```
public class CurrencyPricingData;
```

**Visual Basic**

```
Public Class CurrencyPricingData
```

**File**

CurrencyPricingData.cs

# CurrencyPricingData Constructor

This is the overview for the CurrencyPricingData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | CurrencyPricingData.CurrencyPricingData () (see page 1252) | This is CurrencyPricingData, a member of class CurrencyPricingData. |
| | CurrencyPricingData.CurrencyPricingData (long, decimal, decimal) (see page 1253) | Creates a CurrencyPricingData with a single tierprice of quantity 1 and the supplied sale price. |
| | CurrencyPricingData.CurrencyPricingData (long, decimal, EkEnumeration.PricingType) (see page 1253) | This is CurrencyPricingData, a member of class CurrencyPricingData. |

## CurrencyPricingData.CurrencyPricingData Constructor ()

**C#**

```
public CurrencyPricingData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is CurrencyPricingData, a member of class CurrencyPricingData.

## CurrencyPricingData.CurrencyPricingData Constructor (long, decimal, EkEnumeration.PricingType)

**C#**

```
public CurrencyPricingData(long currencyId, decimal listPrice, EkEnumeration.PricingType
pricingType);
```

**Visual Basic**

```
Public Sub New(currencyId As long, listPrice As decimal, pricingType As
EkEnumeration.PricingType)
```

**Description**

This is CurrencyPricingData, a member of class CurrencyPricingData.

## CurrencyPricingData.CurrencyPricingData Constructor (long, decimal, decimal)

Creates a CurrencyPricingData with a single tierprice of quantity 1 and the supplied sale price.

**C#**

```
public CurrencyPricingData(long currencyId, decimal listPrice, decimal salePrice);
```

**Visual Basic**

```
Public Sub New(currencyId As long, listPrice As decimal, salePrice As decimal)
```

**Parameters**

| Parameters | Description |
|---|---|
| currencyId | Id of currency to create pricing for. |
| listPrice | List Price for currency. |
| salePrice | sale Price for single tier item. |

# CurrencyPricingData Methods

The methods of the CurrencyPricingData class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Clone (see page 1253) | This is Clone, a member of class CurrencyPricingData. |
| | GetSalePrice (see page 1254) | This is GetSalePrice, a member of class CurrencyPricingData. |
| | Validate (see page 1254) | This is Validate, a member of class CurrencyPricingData. |

## CurrencyPricingData.Clone Method

**C#**

```
public CurrencyPricingData Clone();
```

**Visual Basic**

```
Public Function Clone() As CurrencyPricingData
```

**Description**

This is Clone, a member of class CurrencyPricingData.

## CurrencyPricingData.GetSalePrice Method

**C#**

```
public decimal GetSalePrice(int quantity);
```

**Visual Basic**

```
Public Function GetSalePrice(quantity As Integer) As decimal
```

**Description**

This is GetSalePrice, a member of class CurrencyPricingData.

## CurrencyPricingData.Validate Method

**C#**

```
public ValidationResults Validate();
```

**Visual Basic**

```
Public Function Validate() As ValidationResults
```

**Description**

This is Validate, a member of class CurrencyPricingData.

# CurrencyPricingData Properties

The properties of the CurrencyPricingData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AlphaIsoCode (see page 1254) | Gets or sets the alpha iso code of the currency associated with these prices. |
| | CurrencyId (see page 1254) | Gets or sets the Id of currency that this pricing represents. |
| | ListPrice (see page 1255) | Gets or sets the list price for this currency. |
| | NumericIsoCode (see page 1255) | Gets or sets the numeric iso code of the currency associated with these prices. |
| | PricingType (see page 1255) | Gets or sets the PricingType forthe currency. |
| | SalePrice (see page 1255) | Gets the default sale price for a single quantity. |
| | TierPrices (see page 1255) | Gets all the tier prices for this currency. |

## CurrencyPricingData.AlphaIsoCode Property

Gets or sets the alpha iso code of the currency associated with these prices.

**C#**

```
public string AlphaIsoCode;
```

**Visual Basic**

```
Public Property AlphaIsoCode() As string
```

## CurrencyPricingData.CurrencyId Property

Gets or sets the Id of currency that this pricing represents.

**C#**

```
public long CurrencyId;
```

**Visual Basic**

```
Public Property CurrencyId() As long
```

### CurrencyPricingData.ListPrice Property

Gets or sets the list price for this currency.

**C#**

```
public decimal ListPrice;
```

**Visual Basic**

```
Public Property ListPrice() As decimal
```

### CurrencyPricingData.NumericIsoCode Property

Gets or sets the numeric iso code of the currency associated with these prices.

**C#**

```
public int NumericIsoCode;
```

**Visual Basic**

```
Public Property NumericIsoCode() As Integer
```

### CurrencyPricingData.PricingType Property

Gets or sets the PricingType forthe currency.

**C#**

```
public Common.EkEnumeration.PricingType PricingType;
```

**Visual Basic**

```
Public Property PricingType() As Common.EkEnumeration.PricingType
```

### CurrencyPricingData.SalePrice Property

Gets the default sale price for a single quantity.

**C#**

```
public decimal SalePrice;
```

**Visual Basic**

```
Public Property SalePrice() As decimal
```

### CurrencyPricingData.TierPrices Property

Gets all the tier prices for this currency.

**C#**

```
public List<TierPriceData> TierPrices;
```

**Visual Basic**

```
Public Property TierPrices() As List<TierPriceData>
```

## CustomerData Class

Data class representing Customers in system.

### Class Hierarchy

### C#

```
public class CustomerData;
```

### Visual Basic

```
Public Class CustomerData
```

### File

CustomerData.cs

## CustomerData Properties

The properties of the CustomerData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AverageOrderValue (see page 1256) | Gets the average order placed by this customer. |
| | BillingAddressId (see page 1256) | Gets or sets the Address Id (see page 1257) of the billing address for the customer. |
| | DisplayName (see page 1257) | Gets or sets the display name of the customer. |
| | EmailAddress (see page 1257) | Gets or sets the email address of the customer. |
| | FirstName (see page 1257) | Gets or sets the first name of the customer. |
| | Id (see page 1257) | Gets or sets the Id of the customer. |
| | IsDeleted (see page 1257) | Gets the IsDeleted status of the customer. |
| | IsMembershipUser (see page 1257) | Gets or sets the IsMembershipUser status of the customer. |
| | LastName (see page 1258) | Gets or sets the last of the customer. |
| | Password (see page 1258) | Gets or sets the password for the customer. |
| | ShippingAddressId (see page 1258) | Gets or sets Address Id (see page 1257) of the shipping address for this customer. |
| | TotalOrders (see page 1258) | Gets or sets the toal number of orders placed by the customer. |
| | TotalOrderValue (see page 1258) | Gets or sets the total value of orders placed by this customer. |
| | UserId (see page 1258) | Gets or sets the Id (see page 1257) of the Cms (see page 7) User (see page 913) object associated with this customer. |
| | UserName (see page 1259) | Gets or sets the username for the customer. |

## CustomerData.AverageOrderValue Property

Gets the average order placed by this customer.

### C#

```
public decimal AverageOrderValue;
```

### Visual Basic

```
Public Property AverageOrderValue() As decimal
```

## CustomerData.BillingAddressId Property

Gets or sets the Address Id (see page 1257) of the billing address for the customer.

**C#**

```
public long BillingAddressId;
```

**Visual Basic**

```
Public Property BillingAddressId() As long
```

## CustomerData.DisplayName Property

Gets or sets the display name of the customer.

**C#**

```
public string DisplayName;
```

**Visual Basic**

```
Public Property DisplayName() As string
```

## CustomerData.EmailAddress Property

Gets or sets the email address of the customer.

**C#**

```
public string EmailAddress;
```

**Visual Basic**

```
Public Property EmailAddress() As string
```

## CustomerData.FirstName Property

Gets or sets the first name of the customer.

**C#**

```
public string FirstName;
```

**Visual Basic**

```
Public Property FirstName() As string
```

## CustomerData.Id Property

Gets or sets the Id of the customer.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## CustomerData.IsDeleted Property

Gets the IsDeleted status of the customer.

**C#**

```
public bool IsDeleted;
```

**Visual Basic**

```
Public Property IsDeleted() As bool
```

## CustomerData.IsMembershipUser Property

Gets or sets the IsMembershipUser status of the customer.

**C#**

```
public bool IsMembershipUser;
```

**Visual Basic**

```
Public Property IsMembershipUser() As bool
```

## CustomerData.LastName Property

Gets or sets the last of the customer.

**C#**

```
public string LastName;
```

**Visual Basic**

```
Public Property LastName() As string
```

## CustomerData.Password Property

Gets or sets the password for the customer.

**C#**

```
public string Password;
```

**Visual Basic**

```
Public Property Password() As string
```

## CustomerData.ShippingAddressId Property

Gets or sets Address Id (see page 1257) of the shipping address for this customer.

**C#**

```
public long ShippingAddressId;
```

**Visual Basic**

```
Public Property ShippingAddressId() As long
```

## CustomerData.TotalOrders Property

Gets or sets the toal number of orders placed by the customer.

**C#**

```
public int TotalOrders;
```

**Visual Basic**

```
Public Property TotalOrders() As Integer
```

## CustomerData.TotalOrderValue Property

Gets or sets the total value of orders placed by this customer.

**C#**

```
public decimal TotalOrderValue;
```

**Visual Basic**

```
Public Property TotalOrderValue() As decimal
```

## CustomerData.UserId Property

Gets or sets the Id (see page 1257) of the Cms (see page 7) User (see page 913) object associated with this customer.

**C#**

```
public long UserId;
```

**Visual Basic**

```
Public Property UserId() As long
```

### CustomerData.UserName Property

Gets or sets the username for the customer.

**C#**

```
public string UserName;
```

**Visual Basic**

```
Public Property UserName() As string
```

## Dimensions Class

Class used to define dimension information.

**Class Hierarchy**

**C#**

```
public class Dimensions : IComparable<Dimensions>;
```

**Visual Basic**

```
Public Class Dimensions
Inherits IComparable
```

**File**

Measure.cs

## Dimensions Constructor

This is the overview for the Dimensions constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Dimensions.Dimensions () (see page 1259) | This is Dimensions, a member of class Dimensions. |
| | Dimensions.Dimensions (LinearUnit, float, float, float) (see page 1260) | |

### Dimensions.Dimensions Constructor ()

**C#**

```
public Dimensions();
```

**Visual Basic**

```
Public Sub New()
```

### Description

This is Dimensions, a member of class Dimensions.

## Dimensions.Dimensions Constructor (LinearUnit, float, float, float)

### C#

```
public Dimensions(LinearUnit units, float length, float width, float height);
```

### Visual Basic

```
Public Sub New(units As LinearUnit, length As float, width As float, height As float)
```

### Parameters

| Parameters | Description |
|---|---|
| units | Linear Units (see page 1262) that define the dimensions. |
| length | Length (see page 1261) of the package in Units (see page 1262) |
| width | Length (see page 1261) of the package in Units (see page 1262) |
| height | Length (see page 1261) of the package in Units (see page 1262) |

# Dimensions Methods

The methods of the Dimensions class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | Clone (see page 1260) | This is Clone, a member of class Dimensions. |
| | CompareTo (see page 1260) | Compares DImensions based upon CubicUnits (see page 1261) |
| | ConvertUnits (see page 1261) | Converts the current Dimensions (see page 1259) object to a new object in the supplied units. |

## Dimensions.Clone Method

### C#

```
public Dimensions Clone();
```

### Visual Basic

```
Public Function Clone() As Dimensions
```

### Description

This is Clone, a member of class Dimensions.

## Dimensions.CompareTo Method

Compares DImensions based upon CubicUnits (see page 1261)

### C#

```
public int CompareTo(Dimensions other);
```

### Visual Basic

```
Public Function CompareTo(other As Dimensions) As Integer
```

### Parameters

| Parameters | Description |
|---|---|
| other | |

## Dimensions.ConvertUnits Method

Converts the current Dimensions (see page 1259) object to a new object in the supplied units.

**C#**

```
public Dimensions ConvertUnits(LinearUnit toUnit);
```

**Visual Basic**

```
Public Function ConvertUnits(toUnit As LinearUnit) As Dimensions
```

**Parameters**

| Parameters | Description |
|---|---|
| toUnit | LinearUnit (see page 1360) to convert to |

# Dimensions Properties

The properties of the Dimensions class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CubicUnits (see page 1261) | Gets the dimensions cubic units. |
| | Height (see page 1261) | Gets or sets the Length (see page 1261) of the package in Units (see page 1262) |
| | Length (see page 1261) | Gets or sets the Length of the package in Units (see page 1262). |
| | Units (see page 1262) | Gets or sets the Linear Units that define the dimensions. |
| | Width (see page 1262) | Gets or sets the Length (see page 1261) of the package in Units (see page 1262) |

## Dimensions.CubicUnits Property

Gets the dimensions cubic units.

**C#**

```
public float CubicUnits;
```

**Visual Basic**

```
Public Property CubicUnits() As float
```

## Dimensions.Height Property

Gets or sets the Length (see page 1261) of the package in Units (see page 1262)

**C#**

```
public float Height;
```

**Visual Basic**

```
Public Property Height() As float
```

## Dimensions.Length Property

Gets or sets the Length of the package in Units (see page 1262).

**C#**

```
public float Length;
```

**Visual Basic**

```
Public Property Length() As float
```

## Dimensions.Units Property

Gets or sets the Linear Units that define the dimensions.

**C#**

```
public LinearUnit Units;
```

**Visual Basic**

```
Public Property Units() As LinearUnit
```

## Dimensions.Width Property

Gets or sets the Length (see page 1261) of the package in Units (see page 1262)

**C#**

```
public float Width;
```

**Visual Basic**

```
Public Property Width() As float
```

# EntryAttributeData Class

**Class Hierarchy**

**C#**

```
public class EntryAttributeData : ProductTypeAttributeData;
```

**Visual Basic**

```
Public Class EntryAttributeData
Inherits ProductTypeAttributeData
```

**File**

ProductAttributeData.cs

**Description**

This is class Ektron.Cms.Commerce.EntryAttributeData.

# EntryAttributeData Constructor

This is the overview for the EntryAttributeData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | EntryAttributeData.EntryAttributeData () (see page 1263) | This is EntryAttributeData, a member of class EntryAttributeData. |
| | EntryAttributeData.EntryAttributeData (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus) (see page 1263) | This is EntryAttributeData, a member of class EntryAttributeData. |

| | | |
|---|---|---|
| | EntryAttributeData.EntryAttributeData (string, long, object, int, EkEnumeration.ProductTypeAttributeDataType, string, EkEnumeration.ProductTypeAttributeStatus) (see page 1263) | This is EntryAttributeData, a member of class EntryAttributeData. |

## EntryAttributeData.EntryAttributeData Constructor ()

**C#**

```
public EntryAttributeData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is EntryAttributeData, a member of class EntryAttributeData.

## EntryAttributeData.EntryAttributeData Constructor (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus)

**C#**

```
private EntryAttributeData(string name, int displayOrder,
EkEnumeration.ProductTypeAttributeDataType dataType, object defaultValue,
EkEnumeration.ProductTypeAttributeStatus activeStatus);
```

**Visual Basic**

```
Private Sub New(name As string, displayOrder As Integer, dataType As
EkEnumeration.ProductTypeAttributeDataType, defaultValue As object, activeStatus As
EkEnumeration.ProductTypeAttributeStatus)
```

**Description**

This is EntryAttributeData, a member of class EntryAttributeData.

## EntryAttributeData.EntryAttributeData Constructor (string, long, object, int, EkEnumeration.ProductTypeAttributeDataType, string, EkEnumeration.ProductTypeAttributeStatus)

**C#**

```
public EntryAttributeData(string name, long attributeTypeId, object currentValue, int
displayOrder, EkEnumeration.ProductTypeAttributeDataType dataType, string defaultValue,
EkEnumeration.ProductTypeAttributeStatus activeStatus);
```

**Visual Basic**

```
Public Sub New(name As string, attributeTypeId As long, currentValue As object, displayOrder
As Integer, dataType As EkEnumeration.ProductTypeAttributeDataType, defaultValue As string,
activeStatus As EkEnumeration.ProductTypeAttributeStatus)
```

**Description**

This is EntryAttributeData, a member of class EntryAttributeData.

# EntryAttributeData Properties

The properties of the EntryAttributeData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AttributeTypeId (see page 1264) | This is AttributeTypeId, a member of class EntryAttributeData. |
| | CurrentValue (see page 1264) | This is CurrentValue, a member of class EntryAttributeData. |
| | VersionId (see page 1264) | This is VersionId, a member of class EntryAttributeData. |

## EntryAttributeData.AttributeTypeId Property

**C#**

```
public long AttributeTypeId;
```

**Visual Basic**

```
Public Property AttributeTypeId() As long
```

**Description**

This is AttributeTypeId, a member of class EntryAttributeData.

## EntryAttributeData.CurrentValue Property

**C#**

```
public object CurrentValue;
```

**Visual Basic**

```
Public Property CurrentValue() As object
```

**Description**

This is CurrentValue, a member of class EntryAttributeData.

## EntryAttributeData.VersionId Property

**C#**

```
public long VersionId;
```

**Visual Basic**

```
Public Property VersionId() As long
```

**Description**

This is VersionId, a member of class EntryAttributeData.

# EntryData Class

Base class for catalog entry data classes.

### Class Hierarchy

Ektron.Cms.Commerce.EntryData

**C#**

```
public abstract class EntryData;
```

**Visual Basic**

```
Public abstract Class EntryData
```

**File**

EntryData.cs

# EntryData Constructor

This is the overview for the EntryData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | EntryData.EntryData () (see page 1265) | This is EntryData, a member of class EntryData. |
| | EntryData.EntryData (EkEnumeration.CatalogEntryType) (see page 1265) | This is EntryData, a member of class EntryData. |
| | EntryData.EntryData (int, EkEnumeration.CatalogEntryType) (see page 1265) | This is EntryData, a member of class EntryData. |

## EntryData.EntryData Constructor ()

**C#**

```
public EntryData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is EntryData, a member of class EntryData.

## EntryData.EntryData Constructor (EkEnumeration.CatalogEntryType)

**C#**

```
public EntryData(EkEnumeration.CatalogEntryType entryType);
```

**Visual Basic**

```
Public Sub New(entryType As EkEnumeration.CatalogEntryType)
```

**Description**

This is EntryData, a member of class EntryData.

## EntryData.EntryData Constructor (int, EkEnumeration.CatalogEntryType)

**C#**

```
public EntryData(int languageId, EkEnumeration.CatalogEntryType entryType);
```

### Visual Basic

```
Public Sub New(languageId As Integer, entryType As EkEnumeration.CatalogEntryType)
```

### Description

This is EntryData, a member of class EntryData.

## EntryData Properties

The properties of the EntryData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Attributes (see page 1267) | Gets or sets the attributes of the Catalog Entry. |
| | Comment (see page 1267) | This is Comment, a member of class EntryData. |
| | ContentStatus (see page 1267) | The status of the content information. |
| | CurrencyId (see page 1267) | Gets or sets the Id (see page 1269) of the currency used for the current Sale and List prices. |
| | DateModified (see page 1268) | This is DateModified, a member of class EntryData. |
| | Dimensions (see page 1268) | Gets or sets the dimensions of the product. Used for shipping. |
| | DisableInventoryManagement (see page 1268) | Gets or sets the disable inventory management flag. If true, this product does not track inventory and no inventory management is performed. |
| | EndDate (see page 1268) | Get or sets the enddate of the EntryData (see page 1264) object. |
| | EndDateAction (see page 1268) | This is EndDateAction, a member of class EntryData. |
| | EntryType (see page 1269) | The type of entry. |
| | FolderId (see page 1269) | Gets or sets the folder ID of the EntryData (see page 1264) object. |
| | GoLive (see page 1269) | Get or sets the golive of the EntryData (see page 1264) object. |
| | Html (see page 1269) | Get or sets the html/xml of the EntryData (see page 1264) object. |
| | Id (see page 1269) | Gets or sets the ID of the EntryData (see page 1264) object. |
| | Image (see page 1270) | This is Image, a member of class EntryData. |
| | ImageThumbnail (see page 1270) | This is ImageThumbnail, a member of class EntryData. |
| | IsArchived (see page 1270) | Get or sets the Archive flag of the EntryData (see page 1264) object. |
| | IsBuyable (see page 1270) | Get or sets whether the entry is buyable. |
| | IsMarkedForDeletion (see page 1271) | Get or sets the MarkForDeletion flag on the EntryData (see page 1264) object. |
| | IsPublished (see page 1271) | This is IsPublished, a member of class EntryData. |
| | IsTangible (see page 1271) | Indicates if this product is a tangible good. |
| | LanguageId (see page 1271) | Gets or sets the language of the EntryData (see page 1264) object. |
| | LastEditorFirstName (see page 1271) | This is LastEditorFirstName, a member of class EntryData. |
| | LastEditorId (see page 1272) | This is LastEditorId, a member of class EntryData. |
| | LastEditorLastName (see page 1272) | This is LastEditorLastName, a member of class EntryData. |
| | ListPrice (see page 1272) | Gets the current List price from the Pricing (see page 1273) object. To set an item's price, Use the Pricing (see page 1273) property. Create a TierPrice with a quantity of 1. Use the PricingData (see page 1316)(CurrencyId (see page 1267), SalePrice (see page 1274), ListPrice) constructor. |
| | Media (see page 1272) | This is Media, a member of class EntryData. |
| | Metadata (see page 1273) | This is Metadata, a member of class EntryData. |
| | NumberSold (see page 1273) | Gets or sets the Number Sold for this entry. |
| | Pricing (see page 1273) | The entry pricing, including currencies and quantity. |
| | ProductType (see page 1273) | This is ProductType, a member of class EntryData. |
| | QuantityMultiple (see page 1273) | The has these number of items in it. |

| | | |
|---|---|---|
| | Quicklink (see page 1274) | Gets or sets the entry's Quicklink. |
| | SalePrice (see page 1274) | Gets the current Sale price from the Pricing (see page 1273) object. To set an item's sale price, Use the Pricing (see page 1273) property. Create a TierPrice with a quantity of 1. Use the PricingData (see page 1316)(CurrencyId (see page 1267), SalePrice, ListPrice (see page 1272)) constructor. |
| | Sku (see page 1274) | Get or sets the sku of the EntryData (see page 1264) object. |
| | Status (see page 1274) | The status of the entry information. |
| | StatusLanguage (see page 1275) | The language that the entry is checked out to, if any. |
| | Summary (see page 1275) | Get or sets the summary of the EntryData (see page 1264) object. |
| | TaxClassId (see page 1275) | This is TaxClassId, a member of class EntryData. |
| | TemplateId (see page 1275) | This is TemplateId, a member of class EntryData. |
| | Title (see page 1275) | Get or sets the title of the EntryData (see page 1264) object. |
| | Url (see page 1276) | Get or sets the url of the EntryData (see page 1264) object. |
| | VersionId (see page 1276) | The version Id (see page 1269) of the catalog entry. 0 is the current version. |
| | Weight (see page 1276) | Gets or sets the weight of the product. Used for shipping. |

## EntryData.Attributes Property

Gets or sets the attributes of the Catalog Entry.

**C#**

```
public List<EntryAttributeData> Attributes;
```

**Visual Basic**

```
Public Property Attributes() As List<EntryAttributeData>
```

## EntryData.Comment Property

**C#**

```
public string Comment;
```

**Visual Basic**

```
Public Property Comment() As string
```

**Description**

This is Comment, a member of class EntryData.

## EntryData.ContentStatus Property

The status of the content information.

**C#**

```
public string ContentStatus;
```

**Visual Basic**

```
Public Property ContentStatus() As string
```

## EntryData.CurrencyId Property

Gets or sets the Id (see page 1269) of the currency used for the current Sale and List prices.

**C#**

```
public long CurrencyId;
```

**Visual Basic**

```
Public Property CurrencyId() As long
```

## EntryData.DateModified Property

**C#**

```
public DateTime DateModified;
```

**Visual Basic**

```
Public Property DateModified() As DateTime
```

**Description**

This is DateModified, a member of class EntryData.

## EntryData.Dimensions Property

Gets or sets the dimensions of the product. Used for shipping.

**C#**

```
public Dimensions Dimensions;
```

**Visual Basic**

```
Public Property Dimensions() As Dimensions
```

## EntryData.DisableInventoryManagement Property

Gets or sets the disable inventory management flag. If true, this product does not track inventory and no inventory management is performed.

**C#**

```
public bool DisableInventoryManagement;
```

**Visual Basic**

```
Public Property DisableInventoryManagement() As bool
```

## EntryData.EndDate Property

Get or sets the enddate of the EntryData (see page 1264) object.

**C#**

```
public DateTime EndDate;
```

**Visual Basic**

```
Public Property EndDate() As DateTime
```

**Returns**

Datetime of the entry enddate.

## EntryData.EndDateAction Property

**C#**

```
public int EndDateAction;
```

**Visual Basic**

```
Public Property EndDateAction() As Integer
```

**Description**

This is EndDateAction, a member of class EntryData.

## EntryData.EntryType Property

The type of entry.

**C#**

```
public virtual EkEnumeration.CatalogEntryType EntryType;
```

**Visual Basic**

```
Public virtual Property EntryType() As EkEnumeration.CatalogEntryType
```

**Returns**

Integer of the entry quantity multiple.

## EntryData.FolderId Property

Gets or sets the folder ID of the EntryData (see page 1264) object.

**C#**

```
public long FolderId;
```

**Visual Basic**

```
Public Property FolderId() As long
```

**Returns**

ID of the entry folder.

## EntryData.GoLive Property

Get or sets the golive of the EntryData (see page 1264) object.

**C#**

```
public DateTime GoLive;
```

**Visual Basic**

```
Public Property GoLive() As DateTime
```

**Returns**

Datetime of the entry golive.

## EntryData.Html Property

Get or sets the html/xml of the EntryData (see page 1264) object.

**C#**

```
public string Html;
```

**Visual Basic**

```
Public Property Html() As string
```

**Returns**

String of the entry html.

## EntryData.Id Property

Gets or sets the ID of the EntryData (see page 1264) object.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Returns**

ID of the entry title.

## EntryData.Image Property

**C#**

```
public string Image;
```

**Visual Basic**

```
Public Property Image() As string
```

**Description**

This is Image, a member of class EntryData.

## EntryData.ImageThumbnail Property

**C#**

```
public string ImageThumbnail;
```

**Visual Basic**

```
Public Property ImageThumbnail() As string
```

**Description**

This is ImageThumbnail, a member of class EntryData.

## EntryData.IsArchived Property

Get or sets the Archive flag of the EntryData (see page 1264) object.

**C#**

```
public bool IsArchived;
```

**Visual Basic**

```
Public Property IsArchived() As bool
```

**Returns**

Boolean.

## EntryData.IsBuyable Property

Get or sets whether the entry is buyable.

**C#**

```
public bool IsBuyable;
```

**Visual Basic**

```
Public Property IsBuyable() As bool
```

**Returns**

Boolean of the entry flag Buyable.

## EntryData.IsMarkedForDeletion Property

Get or sets the MarkForDeletion flag on the EntryData (see page 1264) object.

**C#**

```
public bool IsMarkedForDeletion;
```

**Visual Basic**

```
Public Property IsMarkedForDeletion() As bool
```

**Returns**

Boolean of the entry flag MarkForDeletion.

## EntryData.IsPublished Property

**C#**

```
public bool IsPublished;
```

**Visual Basic**

```
Public Property IsPublished() As bool
```

**Description**

This is IsPublished, a member of class EntryData.

## EntryData.IsTangible Property

Indicates if this product is a tangible good.

**C#**

```
public bool IsTangible;
```

**Visual Basic**

```
Public Property IsTangible() As bool
```

**Returns**

Boolean value indicating if this product is a tangible good.

## EntryData.LanguageId Property

Gets or sets the language of the EntryData (see page 1264) object.

**C#**

```
public int LanguageId;
```

**Visual Basic**

```
Public Property LanguageId() As Integer
```

## EntryData.LastEditorFirstName Property

**C#**

```
public string LastEditorFirstName;
```

**Visual Basic**

```
Public Property LastEditorFirstName() As string
```

**Description**

This is LastEditorFirstName, a member of class EntryData.

## EntryData.LastEditorId Property

**C#**

```
public long LastEditorId;
```

**Visual Basic**

```
Public Property LastEditorId() As long
```

**Description**

This is LastEditorId, a member of class EntryData.

## EntryData.LastEditorLastName Property

**C#**

```
public string LastEditorLastName;
```

**Visual Basic**

```
Public Property LastEditorLastName() As string
```

**Description**

This is LastEditorLastName, a member of class EntryData.

## EntryData.ListPrice Property

Gets the current List price from the Pricing (see page 1273) object. To set an item's price, Use the Pricing (see page 1273) property. Create a TierPrice with a quantity of 1. Use the PricingData (see page 1316)(CurrencyId (see page 1267), SalePrice (see page 1274), ListPrice) constructor.

**C#**

```
public decimal ListPrice;
```

**Visual Basic**

```
Public Property ListPrice() As decimal
```

## EntryData.Media Property

**C#**

```
public MediaGalleryData Media;
```

**Visual Basic**

```
Public Property Media() As MediaGalleryData
```

**Description**

This is Media, a member of class EntryData.

## EntryData.Metadata Property

**C#**

```
public List<ContentMetaData> Metadata;
```

**Visual Basic**

```
Public Property Metadata() As List<ContentMetaData>
```

**Description**

This is Metadata, a member of class EntryData.

## EntryData.NumberSold Property

Gets or sets the Number Sold for this entry.

**C#**

```
public long NumberSold;
```

**Visual Basic**

```
Public Property NumberSold() As long
```

**Returns**

A content item's Quicklink (see page 1274) information.

## EntryData.Pricing Property

The entry pricing, including currencies and quantity.

**C#**

```
public PricingData Pricing;
```

**Visual Basic**

```
Public Property Pricing() As PricingData
```

**Returns**

PricingData (see page 1316) of the catalog entry.

## EntryData.ProductType Property

**C#**

```
public ProductTypeData ProductType;
```

**Visual Basic**

```
Public Property ProductType() As ProductTypeData
```

**Description**

This is ProductType, a member of class EntryData.

## EntryData.QuantityMultiple Property

The has these number of items in it.

**C#**

```
public int QuantityMultiple;
```

**Visual Basic**

```
Public Property QuantityMultiple() As Integer
```

**Returns**

Integer of the entry quantity multiple.

## EntryData.Quicklink Property

Gets or sets the entry's Quicklink.

**C#**

```
public string Quicklink;
```

**Visual Basic**

```
Public Property Quicklink() As string
```

**Returns**

A content item's Quicklink information.

## EntryData.SalePrice Property

Gets the current Sale price from the Pricing (see page 1273) object. To set an item's sale price, Use the Pricing (see page 1273) property. Create a TierPrice with a quantity of 1. Use the PricingData (see page 1316)(CurrencyId (see page 1267), SalePrice, ListPrice (see page 1272)) constructor.

**C#**

```
public decimal SalePrice;
```

**Visual Basic**

```
Public Property SalePrice() As decimal
```

## EntryData.Sku Property

Get or sets the sku of the EntryData (see page 1264) object.

**C#**

```
public string Sku;
```

**Visual Basic**

```
Public Property Sku() As string
```

**Returns**

String of the entry sku.

## EntryData.Status Property

The status of the entry information.

**C#**

```
public string Status;
```

**Visual Basic**

```
Public Property Status() As string
```

## EntryData.StatusLanguage Property

The language that the entry is checked out to, if any.

**C#**

```
public int StatusLanguage;
```

**Visual Basic**

```
Public Property StatusLanguage() As Integer
```

## EntryData.Summary Property

Get or sets the summary of the EntryData (see page 1264) object.

**C#**

```
public string Summary;
```

**Visual Basic**

```
Public Property Summary() As string
```

**Returns**

String of the entry summary.

## EntryData.TaxClassId Property

**C#**

```
public long TaxClassId;
```

**Visual Basic**

```
Public Property TaxClassId() As long
```

**Description**

This is TaxClassId, a member of class EntryData.

## EntryData.TemplateId Property

**C#**

```
public long TemplateId;
```

**Visual Basic**

```
Public Property TemplateId() As long
```

**Description**

This is TemplateId, a member of class EntryData.

## EntryData.Title Property

Get or sets the title of the EntryData (see page 1264) object.

**C#**

```
public string Title;
```

**Visual Basic**

```
Public Property Title() As string
```

**Returns**

String of the entry title.

## EntryData.Url Property

Get or sets the url of the EntryData (see page 1264) object.

**C#**

```
public string Url;
```

**Visual Basic**

```
Public Property Url() As string
```

**Returns**

String of the entry url.

## EntryData.VersionId Property

The version Id (see page 1269) of the catalog entry. 0 is the current version.

**C#**

```
public long VersionId;
```

**Visual Basic**

```
Public Property VersionId() As long
```

## EntryData.Weight Property

Gets or sets the weight of the product. Used for shipping.

**C#**

```
public Weight Weight;
```

**Visual Basic**

```
Public Property Weight() As Weight
```

# EntryGroupData Class

Data class representing Catalog Entry Groups in system.

**Class Hierarchy**

**C#**

```
public class EntryGroupData;
```

**Visual Basic**

```
Public Class EntryGroupData
```

**File**

EntryGroupData.cs

# EntryGroupData Constructor

This is the overview for the EntryGroupData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | EntryGroupData.EntryGroupData () (see page 1277) | This is EntryGroupData, a member of class EntryGroupData. |
| | EntryGroupData.EntryGroupData (long, string, string, long, EkEnumeration.EntryGroupType) (see page 1277) | |

## EntryGroupData.EntryGroupData Constructor ()

**C#**

```
public EntryGroupData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is EntryGroupData, a member of class EntryGroupData.

## EntryGroupData.EntryGroupData Constructor (long, string, string, long, EkEnumeration.EntryGroupType)

**C#**

```
public EntryGroupData(long id, string groupName, string description, long entryId,
EkEnumeration.EntryGroupType groupType);
```

**Visual Basic**

```
Public Sub New(id As long, groupName As string, description As string, entryId As long,
groupType As EkEnumeration.EntryGroupType)
```

**Parameters**

| Parameters | Description |
|---|---|
| id | id of entry group. |
| description | description of entry group. |
| entryId | Id (see page 1278) of entry owning group |
| groupname | name of entry group. |
| grouptype | type of entry group. |

# EntryGroupData Properties

The properties of the EntryGroupData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Description (see page 1278) | Gets or sets the description of Entry Group. |
| | DisplayOrder (see page 1278) | Gets or sets the display order of the entry group. |

| | | |
|---|---|---|
| | EntryId (see page 1278) | Gets or sets the Id (see page 1278) of Catalog Entry that owns the Entry Group. |
| | GroupType (see page 1278) | Gets or sets the type of Entry Group. |
| | Id (see page 1278) | Gets or sets the Id of Entry Group. |
| | Name (see page 1278) | Gets or sets the Name of Entry Group. |

## EntryGroupData.Description Property

Gets or sets the description of Entry Group.

**C#**

```
public string Description;
```

**Visual Basic**

```
Public Property Description() As string
```

## EntryGroupData.DisplayOrder Property

Gets or sets the display order of the entry group.

**C#**

```
public int DisplayOrder;
```

**Visual Basic**

```
Public Property DisplayOrder() As Integer
```

## EntryGroupData.EntryId Property

Gets or sets the Id (see page 1278) of Catalog Entry that owns the Entry Group.

**C#**

```
public long EntryId;
```

**Visual Basic**

```
Public Property EntryId() As long
```

## EntryGroupData.GroupType Property

Gets or sets the type of Entry Group.

**C#**

```
public EkEnumeration.EntryGroupType GroupType;
```

**Visual Basic**

```
Public Property GroupType() As EkEnumeration.EntryGroupType
```

## EntryGroupData.Id Property

Gets or sets the Id of Entry Group.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## EntryGroupData.Name Property

Gets or sets the Name of Entry Group.

**C#**

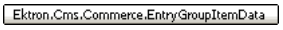
```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

# EntryGroupItemData Class

**Class Hierarchy**




**C#**

```
public class EntryGroupItemData;
```

**Visual Basic**

```
Public Class EntryGroupItemData
```

**File**

EntryGroupItemData.cs

**Description**

This is class Ektron.Cms.Commerce.EntryGroupItemData.

## EntryGroupItemData Constructor

This is the overview for the EntryGroupItemData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | EntryGroupItemData.EntryGroupItemData () (see page 1279) | This is EntryGroupItemData, a member of class EntryGroupItemData. |
| | EntryGroupItemData.EntryGroupItemData (long, long, long) (see page 1279) | This is EntryGroupItemData, a member of class EntryGroupItemData. |

### EntryGroupItemData.EntryGroupItemData Constructor ()

**C#**

```
public EntryGroupItemData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is EntryGroupItemData, a member of class EntryGroupItemData.

### EntryGroupItemData.EntryGroupItemData Constructor (long, long, long)

**C#**

```
public EntryGroupItemData(long id, long groupId, long entryId);
```

**Visual Basic**

```
Public Sub New(id As long, groupId As long, entryId As long)
```

**Description**

This is EntryGroupItemData, a member of class EntryGroupItemData.

# EntryGroupItemData Properties

The properties of the EntryGroupItemData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DisplayOrder (see page 1280) | This is DisplayOrder, a member of class EntryGroupItemData. |
| | EntryId (see page 1280) | This is EntryId, a member of class EntryGroupItemData. |
| | GroupId (see page 1280) | This is GroupId, a member of class EntryGroupItemData. |
| | Id (see page 1281) | This is Id, a member of class EntryGroupItemData. |

## EntryGroupItemData.DisplayOrder Property

**C#**

```
public int DisplayOrder;
```

**Visual Basic**

```
Public Property DisplayOrder() As Integer
```

**Description**

This is DisplayOrder, a member of class EntryGroupItemData.

## EntryGroupItemData.EntryId Property

**C#**

```
public long EntryId;
```

**Visual Basic**

```
Public Property EntryId() As long
```

**Description**

This is EntryId, a member of class EntryGroupItemData.

## EntryGroupItemData.GroupId Property

**C#**

```
public long GroupId;
```

**Visual Basic**

```
Public Property GroupId() As long
```

**Description**

This is GroupId, a member of class EntryGroupItemData.

## EntryGroupItemData.Id Property

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Description**

This is Id, a member of class EntryGroupItemData.

# ExchangeRateData Class

**Class Hierarchy**

**C#**

```
public class ExchangeRateData;
```

**Visual Basic**

```
Public Class ExchangeRateData
```

**File**

ExchangeRateData.cs

**Description**

This is class Ektron.Cms.Commerce.ExchangeRateData.

## ExchangeRateData.ExchangeRateData Constructor

**C#**

```
public ExchangeRateData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is ExchangeRateData, a member of class ExchangeRateData.

## ExchangeRateData Methods

The methods of the ExchangeRateData class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | ConvertPrice (see page 1281) | Converts a pric ein the base currency to a price in the exchange currency using the rate defined. |
| | Validate (see page 1282) | Validates the ExchangeRate data object for saving. |

## ExchangeRateData.ConvertPrice Method

Converts a pric ein the base currency to a price in the exchange currency using the rate defined.

**C#**

```
public decimal ConvertPrice(decimal basePrice);
```

**Visual Basic**

```
Public Function ConvertPrice(basePrice As decimal) As decimal
```

**Parameters**

| Parameters | Description |
|---|---|
| basePrice | price to convert. |

## ExchangeRateData.Validate Method

Validates the ExchangeRate data object for saving.

**C#**

```
public ValidationResults Validate();
```

**Visual Basic**

```
Public Function Validate() As ValidationResults
```

**Returns**

returns ValidationResult object with results.

# ExchangeRateData Properties

The properties of the ExchangeRateData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | BaseCurrencyId (see page 1282) | Gets or sets the Currency Id for the base currency. This is the currency that is being used as the basis for the exchange rate. i.e. If you are converting from USD to AUD, USD is the base currency |
| | EffectiveDate (see page 1282) | Gets or sets the date and time in which this rate becomes effective. |
| | ExchangeCurrencyId (see page 1283) | Gets or sets the Currency Id for the exchange currency. This is the currency that the currency amount is being converted to. If you are converting from USD to AUD, AUD is the exchange currency. |
| | Rate (see page 1283) | Gets or sets the actual exchange rate between to currencies. This is the rate applied to base currency to convert it to exchange currency. |

## ExchangeRateData.BaseCurrencyId Property

Gets or sets the Currency Id for the base currency. This is the currency that is being used as the basis for the exchange rate. i.e. If you are converting from USD to AUD, USD is the base currency

**C#**

```
public int BaseCurrencyId;
```

**Visual Basic**

```
Public Property BaseCurrencyId() As Integer
```

## ExchangeRateData.EffectiveDate Property

Gets or sets the date and time in which this rate becomes effective.

**C#**

```
public DateTime? EffectiveDate;
```

**Visual Basic**

```
Public Property EffectiveDate() As DateTime?
```

## ExchangeRateData.ExchangeCurrencyId Property

Gets or sets the Currency Id for the exchange currency. This is the currency that the currency amount is being converted to. If you are converting from USD to AUD, AUD is the exchange currency.

**C#**

```
public int ExchangeCurrencyId;
```

**Visual Basic**

```
Public Property ExchangeCurrencyId() As Integer
```

## ExchangeRateData.Rate Property

Gets or sets the actual exchange rate between to currencies. This is the rate applied to base currency to convert it to exchange currency.

**C#**

```
public decimal Rate;
```

**Visual Basic**

```
Public Property Rate() As decimal
```

# ImageMediaData Class

**Class Hierarchy**

**C#**

```
public class ImageMediaData : MediaData;
```

**Visual Basic**

```
Public Class ImageMediaData
Inherits MediaData
```

**File**

ImageMediaData.cs

**Description**

This is class Ektron.Cms.ImageMediaData.

# ImageMediaData Properties

The properties of the ImageMediaData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Thumbnails (see page 1284) | This is Thumbnails, a member of class ImageMediaData. |

## ImageMediaData.Thumbnails Property

**C#**

```
public List<ThumbnailData> Thumbnails;
```

**Visual Basic**

```
Public Property Thumbnails() As List<ThumbnailData>
```

**Description**

This is Thumbnails, a member of class ImageMediaData.

# InventoryData Class

**Class Hierarchy**

**C#**

```
public class InventoryData;
```

**Visual Basic**

```
Public Class InventoryData
```

**File**

InventoryData.cs

**Description**

This is class Ektron.Cms.Commerce.InventoryData.

# InventoryData Properties

The properties of the InventoryData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | EntryId (see page 1284) | Gets or sets the Id of entry that this inventory data represents. |
| | ReorderLevel (see page 1285) | Gets or sets the inventory level at which stock should be reordered. |
| | UnitsInStock (see page 1285) | Gets or sets the the number of units currently in stock for this product. |
| | UnitsOnOrder (see page 1285) | Gets or sets The number of units currently on order for this product. This is the number of units that the store has ordered to replenish inventory and NOT the number of units that are currently on order by customers. |

## InventoryData.EntryId Property

Gets or sets the Id of entry that this inventory data represents.

**C#**

```
public long EntryId;
```

**Visual Basic**

```
Public Property EntryId() As long
```

## InventoryData.ReorderLevel Property

Gets or sets the inventory level at which stock should be reordered.

**C#**

```
public int ReorderLevel;
```

**Visual Basic**

```
Public Property ReorderLevel() As Integer
```

## InventoryData.UnitsInStock Property

Gets or sets the the number of units currently in stock for this product.

**C#**

```
public int UnitsInStock;
```

**Visual Basic**

```
Public Property UnitsInStock() As Integer
```

## InventoryData.UnitsOnOrder Property

Gets or sets The number of units currently on order for this product. This is the number of units that the store has ordered to replenish inventory and NOT the number of units that are currently on order by customers.

**C#**

```
public int UnitsOnOrder;
```

**Visual Basic**

```
Public Property UnitsOnOrder() As Integer
```

# KitConfigData Class

Invitation class used for sending friend requests.

**Class Hierarchy**

**C#**

```
public class KitConfigData;
```

**Visual Basic**

```
Public Class KitConfigData
```

**File**

KitConfigData.cs

# KitConfigData.KitConfigData Constructor

**C#**

```
public KitConfigData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is KitConfigData, a member of class KitConfigData.

# KitConfigData Methods

The methods of the KitConfigData class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | GetGroupIndex (see page 1286) | This is GetGroupIndex, a member of class KitConfigData. |

## KitConfigData.GetGroupIndex Method

**C#**

```
public int GetGroupIndex(long groupId);
```

**Visual Basic**

```
Public Function GetGroupIndex(groupId As long) As Integer
```

**Description**

This is GetGroupIndex, a member of class KitConfigData.

# KitConfigData Properties

The properties of the KitConfigData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | BasketItemId (see page 1286) | This is BasketItemId, a member of class KitConfigData. |
| | Groups (see page 1286) | This is Groups, a member of class KitConfigData. |
| | Id (see page 1287) | This is Id, a member of class KitConfigData. |
| | PriceModifier (see page 1287) | This is PriceModifier, a member of class KitConfigData. |

## KitConfigData.BasketItemId Property

**C#**

```
public long BasketItemId;
```

**Visual Basic**

```
Public Property BasketItemId() As long
```

**Description**

This is BasketItemId, a member of class KitConfigData.

## KitConfigData.Groups Property

**C#**

```
public List<OptionGroupData> Groups;
```

**Visual Basic**

```
Public Property Groups() As List<OptionGroupData>
```

**Description**

This is Groups, a member of class KitConfigData.

## KitConfigData.Id Property

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Description**

This is Id, a member of class KitConfigData.

## KitConfigData.PriceModifier Property

**C#**

```
public decimal PriceModifier;
```

**Visual Basic**

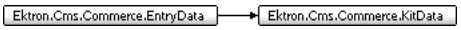
```
Public Property PriceModifier() As decimal
```

**Description**

This is PriceModifier, a member of class KitConfigData.

# KitData Class

**Class Hierarchy**




**C#**

```
public class KitData : EntryData;
```

**Visual Basic**

```
Public Class KitData
Inherits EntryData
```

**File**

KitData.cs

# KitData.KitData Constructor

**C#**

```
public KitData();
```

**Visual Basic**

```
Public Sub New()
```

### Description

This is KitData, a member of class KitData.

## KitData Properties

The properties of the KitData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | OptionGroups (see page 1288) | This is OptionGroups, a member of class KitData. |

## KitData.OptionGroups Property

### C#

```
public List<OptionGroupData> OptionGroups;
```

### Visual Basic

```
Public Property OptionGroups() As List<OptionGroupData>
```

### Description

This is OptionGroups, a member of class KitData.

# MediaData Class

### Class Hierarchy

### C#

```
public abstract class MediaData : IMediaData;
```

### Visual Basic

```
Public abstract Class MediaData
Inherits IMediaData
```

### File

MediaData.cs

### Description

This is class Ektron.Cms.MediaData.

## MediaData Properties

The properties of the MediaData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Alt (see page 1289) | This is Alt, a member of class MediaData. |
| | FileName (see page 1289) | This is FileName, a member of class MediaData. |
| | FilePath (see page 1289) | This is FilePath, a member of class MediaData. |
| | Height (see page 1289) | This is Height, a member of class MediaData. |

| | | |
|---|---|---|
| | Id (see page 1289) | This is Id, a member of class MediaData. |
| | MimeType (see page 1290) | This is MimeType, a member of class MediaData. |
| | Sequence (see page 1290) | This is Sequence, a member of class MediaData. |
| | Title (see page 1290) | This is Title, a member of class MediaData. |
| | Width (see page 1290) | This is Width, a member of class MediaData. |

## MediaData.Alt Property

**C#**

```
public string Alt;
```

**Visual Basic**

```
Public Property Alt() As string
```

**Description**

This is Alt, a member of class MediaData.

## MediaData.FileName Property

**C#**

```
public string FileName;
```

**Visual Basic**

```
Public Property FileName() As string
```

**Description**

This is FileName, a member of class MediaData.

## MediaData.FilePath Property

**C#**

```
public string FilePath;
```

**Visual Basic**

```
Public Property FilePath() As string
```

**Description**

This is FilePath, a member of class MediaData.

## MediaData.Height Property

**C#**

```
public int Height;
```

**Visual Basic**

```
Public Property Height() As Integer
```

**Description**

This is Height, a member of class MediaData.

## MediaData.Id Property

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Description**

This is Id, a member of class MediaData.

## MediaData.MimeType Property

**C#**

```
public string MimeType;
```

**Visual Basic**

```
Public Property MimeType() As string
```

**Description**

This is MimeType, a member of class MediaData.

## MediaData.Sequence Property

**C#**

```
public int Sequence;
```

**Visual Basic**

```
Public Property Sequence() As Integer
```

**Description**

This is Sequence, a member of class MediaData.

## MediaData.Title Property

**C#**

```
public string Title;
```

**Visual Basic**

```
Public Property Title() As string
```

**Description**

This is Title, a member of class MediaData.

## MediaData.Width Property

**C#**

```
public int Width;
```

**Visual Basic**

```
Public Property Width() As Integer
```

**Description**

This is Width, a member of class MediaData.

# MediaGalleryData Class

### Class Hierarchy

### C#

```
public class MediaGalleryData : IMediaGallery;
```

### Visual Basic

```
Public Class MediaGalleryData
Inherits IMediaGallery
```

### File

MediaGalleryData.cs

### Description

This is class Ektron.Cms.MediaGalleryData.

# MediaGalleryData Properties

The properties of the MediaGalleryData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Images (see page 1291) | This is Images, a member of class MediaGalleryData. |
| | ObjectId (see page 1291) | This is ObjectId, a member of class MediaGalleryData. |

## MediaGalleryData.Images Property

### C#

```
public List<ImageMediaData> Images;
```

### Visual Basic

```
Public Property Images() As List<ImageMediaData>
```

### Description

This is Images, a member of class MediaGalleryData.

## MediaGalleryData.ObjectId Property

### C#

```
public long ObjectId;
```

### Visual Basic

```
Public Property ObjectId() As long
```

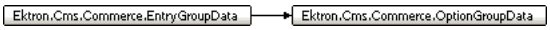
### Description

This is ObjectId, a member of class MediaGalleryData.

# OptionGroupData Class

Data class representing Option groups for Entry Kits.

**Class Hierarchy**




**C#**

```
public class OptionGroupData : EntryGroupData;
```

**Visual Basic**

```
Public Class OptionGroupData
Inherits EntryGroupData
```

**File**

OptionGroupData.cs

# OptionGroupData Constructor

This is the overview for the OptionGroupData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | OptionGroupData.OptionGroupData () (see page 1292) | This is OptionGroupData, a member of class OptionGroupData. |
| | OptionGroupData.OptionGroupData (long, string, string, string, long) (see page 1292) | |
| | OptionGroupData.OptionGroupData (string, string) (see page 1293) | |

## OptionGroupData.OptionGroupData Constructor ()

**C#**

```
public OptionGroupData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is OptionGroupData, a member of class OptionGroupData.

## OptionGroupData.OptionGroupData Constructor (long, string, string, string, long)

**C#**

```
public OptionGroupData(long id, string groupName, string description, string image, long
entryId);
```

**Visual Basic**

```
Public Sub New(id As long, groupName As string, description As string, image As string,
entryId As long)
```

### Parameters

| Parameters | Description |
|---|---|
| id | Id (see page 1278) of option group. |
| groupName | Name (see page 1278) of the option group. |
| description | Description (see page 1278) of the OptionGroup |
| image | path to image associated with option group. |
| entryId | Id (see page 1278) of entry owning this option group. |

## OptionGroupData.OptionGroupData Constructor (string, string)

**C#**

```
public OptionGroupData(string groupName, string description);
```

**Visual Basic**

```
Public Sub New(groupName As string, description As string)
```

### Parameters

| Parameters | Description |
|---|---|
| groupName | Name (see page 1278) of the option group. |
| description | Description (see page 1278) of the OptionGroup |
| image | path to image associated with option group. |

# OptionGroupData Properties

The properties of the OptionGroupData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Image (see page 1293) | Gets or sets the path to the image associated with option group. |
| | Options (see page 1293) | gets the list of options in this option groups. |

## OptionGroupData.Image Property

Gets or sets the path to the image associated with option group.

**C#**

```
public string Image;
```

**Visual Basic**

```
Public Property Image() As string
```

## OptionGroupData.Options Property

gets the list of options in this option groups.

**C#**

```
public OptionGroupItemCollection Options;
```

**Visual Basic**

```
Public Property Options() As OptionGroupItemCollection
```

# OptionGroupItemCollection Class

Collection (see page 88) class for OptionGroupItemData (see page 1295).

### Class Hierarchy

### C#

```
public class OptionGroupItemCollection : Collection<OptionGroupItemData>;
```

### Visual Basic

```
Public Class OptionGroupItemCollection
Inherits Collection
```

### File

OptionGroupItemData.cs

## OptionGroupItemCollection.OptionGroupItemCollection Constructor

### C#

```
public OptionGroupItemCollection();
```

### Visual Basic

```
Public Sub New()
```

### Description

This is OptionGroupItemCollection, a member of class OptionGroupItemCollection.

## OptionGroupItemCollection Methods

The methods of the OptionGroupItemCollection class are listed here.

### Public Methods

| | Name | Description |
|---|---|---|
| | AddRange (see page 1294) | This is AddRange, a member of class OptionGroupItemCollection. |
| | Find (see page 1294) | This is Find, a member of class OptionGroupItemCollection. |
| | FindAll (see page 1295) | This is FindAll, a member of class OptionGroupItemCollection. |
| | FindIndex (see page 1295) | This is FindIndex, a member of class OptionGroupItemCollection. |

## OptionGroupItemCollection.AddRange Method

### C#

```
public void AddRange(IEnumerable<OptionGroupItemData> collection);
```

### Visual Basic

```
Public Function AddRange(collection As IEnumerable<OptionGroupItemData>) As void
```

### Description

This is AddRange, a member of class OptionGroupItemCollection.

## OptionGroupItemCollection.Find Method

### C#

```
public OptionGroupItemData Find(Predicate<OptionGroupItemData> match);
```

### Visual Basic

```
Public Function Find(match As Predicate<OptionGroupItemData>) As OptionGroupItemData
```

**Description**

This is Find, a member of class OptionGroupItemCollection.

## OptionGroupItemCollection.FindAll Method

**C#**

```
public List<OptionGroupItemData> FindAll(Predicate<OptionGroupItemData> match);
```

**Visual Basic**

```
Public Function FindAll(match As Predicate<OptionGroupItemData>) As List<OptionGroupItemData>
```

**Description**

This is FindAll, a member of class OptionGroupItemCollection.

## OptionGroupItemCollection.FindIndex Method

**C#**

```
public int FindIndex(Predicate<OptionGroupItemData> match);
```

**Visual Basic**

```
Public Function FindIndex(match As Predicate<OptionGroupItemData>) As Integer
```

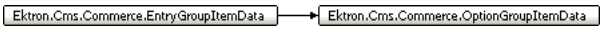
**Description**

This is FindIndex, a member of class OptionGroupItemCollection.

# OptionGroupItemData Class

Data class representing option group items in the CMS. Option Group Items are option items for a particular Kit Grouping.

**Class Hierarchy**




**C#**

```
public class OptionGroupItemData : EntryGroupItemData;
```

**Visual Basic**

```
Public Class OptionGroupItemData
Inherits EntryGroupItemData
```

**File**

OptionGroupItemData.cs

# OptionGroupItemData Constructor

This is the overview for the OptionGroupItemData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | OptionGroupItemData.OptionGroupItemData () (see page 1296) | This is OptionGroupItemData, a member of class OptionGroupItemData. |

| | | |
|---|---|---|
| | OptionGroupItemData.OptionGroupItemData (long, long, string, decimal) (see page 1296) | |
| | OptionGroupItemData.OptionGroupItemData (string, decimal) (see page 1296) | |

## OptionGroupItemData.OptionGroupItemData Constructor ()

**C#**

```
public OptionGroupItemData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is OptionGroupItemData, a member of class OptionGroupItemData.

## OptionGroupItemData.OptionGroupItemData Constructor (long, long, string, decimal)

**C#**

```
public OptionGroupItemData(long id, long groupId, string optionName, decimal
priceModification);
```

**Visual Basic**

```
Public Sub New(id As long, groupId As long, optionName As string, priceModification As decimal)
```

**Parameters**

| Parameters | Description |
|---|---|
| id | Id (see page 1281) of option group item. |
| groupId | Id (see page 1281) of option group this item belongs to. |
| optionName | Name (see page 1297) of the option. |
| priceModification | Price modifier for this option. |

## OptionGroupItemData.OptionGroupItemData Constructor (string, decimal)

**C#**

```
public OptionGroupItemData(string optionName, decimal priceModification);
```

**Visual Basic**

```
Public Sub New(optionName As string, priceModification As decimal)
```

**Parameters**

| Parameters | Description |
|---|---|
| optionName | Name (see page 1297) of the option. |
| priceModification | Price modifier for this option. |

# OptionGroupItemData Properties

The properties of the OptionGroupItemData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CurrencyId (see page 1297) | Gets or sets the currency Id (see page 1281). |

| | | |
|---|---|---|
| | ExtraText (see page 1297) | Gets or sets any extra test associated with this option item. |
| | Name (see page 1297) | Gets or sets the name of the option item. |
| | PriceModification (see page 1297) | Gets or sets the price modification value for this option. i.e. this option increases the price $10.00. |
| | PriceModificationType (see page 1297) | Gets or sets the proce modification type for this option. i.e. is the price modification a percentage or fixed amount. |

## OptionGroupItemData.CurrencyId Property

Gets or sets the currency Id (see page 1281).

**C#**

```
public int CurrencyId;
```

**Visual Basic**

```
Public Property CurrencyId() As Integer
```

## OptionGroupItemData.ExtraText Property

Gets or sets any extra test associated with this option item.

**C#**

```
public string ExtraText;
```

**Visual Basic**

```
Public Property ExtraText() As string
```

## OptionGroupItemData.Name Property

Gets or sets the name of the option item.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

## OptionGroupItemData.PriceModification Property

Gets or sets the price modification value for this option. i.e. this option increases the price $10.00.

**C#**

```
public decimal PriceModification;
```

**Visual Basic**

```
Public Property PriceModification() As decimal
```

## OptionGroupItemData.PriceModificationType Property

Gets or sets the proce modification type for this option. i.e. is the price modification a percentage or fixed amount.

**C#**

```
public EkEnumeration.PriceModificationType PriceModificationType;
```

**Visual Basic**

```
Public Property PriceModificationType() As EkEnumeration.PriceModificationType
```

# OrderData Class

### Class Hierarchy

### C#

```
public class OrderData;
```

### Visual Basic

```
Public Class OrderData
```

### File

OrderData.cs

# OrderData Properties

The properties of the OrderData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | BillingAddressId (see page 1299) | This is BillingAddressId, a member of class OrderData. |
| | ClientIP (see page 1299) | This is ClientIP, a member of class OrderData. |
| | CouponTotal (see page 1299) | Gets or sets the total coupon discount for the order. |
| | CouponUsed (see page 1299) | This is CouponUsed, a member of class OrderData. |
| | Currency (see page 1299) | This is Currency, a member of class OrderData. |
| | Customer (see page 1300) | This is Customer, a member of class OrderData. |
| | CustomerType (see page 1300) | This is CustomerType, a member of class OrderData. |
| | DateCompleted (see page 1300) | This is DateCompleted, a member of class OrderData. |
| | DateCreated (see page 1300) | This is DateCreated, a member of class OrderData. |
| | DateRequired (see page 1300) | This is DateRequired, a member of class OrderData. |
| | HandlingTotal (see page 1301) | Gets or sets the total handling charges for the order. |
| | HttpHost (see page 1301) | This is HttpHost, a member of class OrderData. |
| | Id (see page 1301) | This is Id, a member of class OrderData. |
| | OrderTotal (see page 1301) | Gets or sets the Order Total. |
| | Parts (see page 1301) | This is Parts, a member of class OrderData. |
| | ShippingTotal (see page 1301) | Gets or sets the total shipping charged for order. |
| | SpecialInstructions (see page 1302) | This is SpecialInstructions, a member of class OrderData. |
| | StageId (see page 1302) | This is StageId, a member of class OrderData. |
| | StageName (see page 1302) | This is StageName, a member of class OrderData. |
| | Status (see page 1302) | This is Status, a member of class OrderData. |
| | Subtotal (see page 1302) | Gets or sets the Order SubTotal. |
| | TaxTotal (see page 1303) | Gets or sets the total tax amount for the order. |
| | VisitorId (see page 1303) | This is VisitorId, a member of class OrderData. |

## OrderData.BillingAddressId Property

**C#**

```
public long BillingAddressId;
```

**Visual Basic**

```
Public Property BillingAddressId() As long
```

**Description**

This is BillingAddressId, a member of class OrderData.

## OrderData.ClientIP Property

**C#**

```
public string ClientIP;
```

**Visual Basic**

```
Public Property ClientIP() As string
```

**Description**

This is ClientIP, a member of class OrderData.

## OrderData.CouponTotal Property

Gets or sets the total coupon discount for the order.

**C#**

```
public decimal CouponTotal;
```

**Visual Basic**

```
Public Property CouponTotal() As decimal
```

## OrderData.CouponUsed Property

**C#**

```
public bool CouponUsed;
```

**Visual Basic**

```
Public Property CouponUsed() As bool
```

**Description**

This is CouponUsed, a member of class OrderData.

## OrderData.Currency Property

**C#**

```
public CurrencyData Currency;
```

**Visual Basic**

```
Public Property Currency() As CurrencyData
```

**Description**

This is Currency, a member of class OrderData.

## OrderData.Customer Property

**C#**

```
public CustomerData Customer;
```

**Visual Basic**

```
Public Property Customer() As CustomerData
```

**Description**

This is Customer, a member of class OrderData.

## OrderData.CustomerType Property

**C#**

```
public EkEnumeration.CustomerType CustomerType;
```

**Visual Basic**

```
Public Property CustomerType() As EkEnumeration.CustomerType
```

**Description**

This is CustomerType, a member of class OrderData.

## OrderData.DateCompleted Property

**C#**

```
public DateTime DateCompleted;
```

**Visual Basic**

```
Public Property DateCompleted() As DateTime
```

**Description**

This is DateCompleted, a member of class OrderData.

## OrderData.DateCreated Property

**C#**

```
public DateTime DateCreated;
```

**Visual Basic**

```
Public Property DateCreated() As DateTime
```

**Description**

This is DateCreated, a member of class OrderData.

## OrderData.DateRequired Property

**C#**

```
public DateTime DateRequired;
```

**Visual Basic**

```
Public Property DateRequired() As DateTime
```

**Description**

This is DateRequired, a member of class OrderData.

## OrderData.HandlingTotal Property

Gets or sets the total handling charges for the order.

**C#**

```
public decimal HandlingTotal;
```

**Visual Basic**

```
Public Property HandlingTotal() As decimal
```

## OrderData.HttpHost Property

**C#**

```
public string HttpHost;
```

**Visual Basic**

```
Public Property HttpHost() As string
```

**Description**

This is HttpHost, a member of class OrderData.

## OrderData.Id Property

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Description**

This is Id, a member of class OrderData.

## OrderData.OrderTotal Property

Gets or sets the Order Total.

**C#**

```
public decimal OrderTotal;
```

**Visual Basic**

```
Public Property OrderTotal() As decimal
```

## OrderData.Parts Property

**C#**

```
public List<OrderPartData> Parts;
```

**Visual Basic**

```
Public Property Parts() As List<OrderPartData>
```

**Description**

This is Parts, a member of class OrderData.

## OrderData.ShippingTotal Property

Gets or sets the total shipping charged for order.

**C#**

```
public decimal ShippingTotal;
```

**Visual Basic**

```
Public Property ShippingTotal() As decimal
```

## OrderData.SpecialInstructions Property

**C#**

```
public string SpecialInstructions;
```

**Visual Basic**

```
Public Property SpecialInstructions() As string
```

**Description**

This is SpecialInstructions, a member of class OrderData.

## OrderData.StageId Property

**C#**

```
public int StageId;
```

**Visual Basic**

```
Public Property StageId() As Integer
```

**Description**

This is StageId, a member of class OrderData.

## OrderData.StageName Property

**C#**

```
public string StageName;
```

**Visual Basic**

```
Public Property StageName() As string
```

**Description**

This is StageName, a member of class OrderData.

## OrderData.Status Property

**C#**

```
public EkEnumeration.OrderStatus Status;
```

**Visual Basic**

```
Public Property Status() As EkEnumeration.OrderStatus
```

**Description**

This is Status, a member of class OrderData.

## OrderData.Subtotal Property

Gets or sets the Order SubTotal.

**C#**

```
public decimal Subtotal;
```

**Visual Basic**

```
Public Property Subtotal() As decimal
```

## OrderData.TaxTotal Property

Gets or sets the total tax amount for the order.

**C#**

```
public decimal TaxTotal;
```

**Visual Basic**

```
Public Property TaxTotal() As decimal
```

## OrderData.VisitorId Property

**C#**

```
public string VisitorId;
```

**Visual Basic**

```
Public Property VisitorId() As string
```

**Description**

This is VisitorId, a member of class OrderData.

# OrderLineData Class

**Class Hierarchy**

**C#**

```
public class OrderLineData;
```

**Visual Basic**

```
Public Class OrderLineData
```

**File**

OrderLineData.cs

# OrderLineData Properties

The properties of the OrderLineData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Configuration (see page 1304) | This is Configuration, a member of class OrderLineData. |
| | ItemType (see page 1304) | This is ItemType, a member of class OrderLineData. |
| | LineItemId (see page 1304) | This is LineItemId, a member of class OrderLineData. |
| | OrderPartId (see page 1304) | This is OrderPartId, a member of class OrderLineData. |
| | PriceEach (see page 1304) | Gets or sets the unit sale price for each line item. |

| | | |
|---|---|---|
| | PriceSubTotal (see page 1305) | Gets or sets the line item subtotal. This does not reflect and discounts or adjustments. |
| | PriceTotal (see page 1305) | Gets or sets the total line item price. This reflects quantity and any applicable discounts. |
| | ProductId (see page 1305) | This is ProductId, a member of class OrderLineData. |
| | ProductLanguage (see page 1305) | This is ProductLanguage, a member of class OrderLineData. |
| | Quantity (see page 1305) | This is Quantity, a member of class OrderLineData. |

## OrderLineData.Configuration Property

**C#**

```
public OrderKitConfigData Configuration;
```

**Visual Basic**

```
Public Property Configuration() As OrderKitConfigData
```

**Description**

This is Configuration, a member of class OrderLineData.

## OrderLineData.ItemType Property

**C#**

```
public EkEnumeration.CatalogEntryType ItemType;
```

**Visual Basic**

```
Public Property ItemType() As EkEnumeration.CatalogEntryType
```

**Description**

This is ItemType, a member of class OrderLineData.

## OrderLineData.LineItemId Property

**C#**

```
public long LineItemId;
```

**Visual Basic**

```
Public Property LineItemId() As long
```

**Description**

This is LineItemId, a member of class OrderLineData.

## OrderLineData.OrderPartId Property

**C#**

```
public long OrderPartId;
```

**Visual Basic**

```
Public Property OrderPartId() As long
```

**Description**

This is OrderPartId, a member of class OrderLineData.

## OrderLineData.PriceEach Property

Gets or sets the unit sale price for each line item.

**C#**

```
public decimal PriceEach;
```

**Visual Basic**

```
Public Property PriceEach() As decimal
```

## OrderLineData.PriceSubTotal Property

Gets or sets the line item subtotal. This does not reflect and discounts or adjustments.

**C#**

```
public decimal PriceSubTotal;
```

**Visual Basic**

```
Public Property PriceSubTotal() As decimal
```

## OrderLineData.PriceTotal Property

Gets or sets the total line item price. This reflects quantity and any applicable discounts.

**C#**

```
public decimal PriceTotal;
```

**Visual Basic**

```
Public Property PriceTotal() As decimal
```

## OrderLineData.ProductId Property

**C#**

```
public long ProductId;
```

**Visual Basic**

```
Public Property ProductId() As long
```

**Description**

This is ProductId, a member of class OrderLineData.

## OrderLineData.ProductLanguage Property

**C#**

```
public long ProductLanguage;
```

**Visual Basic**

```
Public Property ProductLanguage() As long
```

**Description**

This is ProductLanguage, a member of class OrderLineData.

## OrderLineData.Quantity Property

**C#**

```
public int Quantity;
```

**Visual Basic**

```
Public Property Quantity() As Integer
```

**Description**

This is Quantity, a member of class OrderLineData.

# OrderPartData Class

data class representing order part sin the cms.

**Class Hierarchy**

**C#**

```
public class OrderPartData;
```

**Visual Basic**

```
Public Class OrderPartData
```

**File**

OrderPartData.cs

## OrderPartData Properties

The properties of the OrderPartData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DateShipped (see page 1306) | This is DateShipped, a member of class OrderPartData. |
| | HandlingTotal (see page 1307) | This is HandlingTotal, a member of class OrderPartData. |
| | Lines (see page 1307) | This is Lines, a member of class OrderPartData. |
| | OrderId (see page 1307) | This is OrderId, a member of class OrderPartData. |
| | PartId (see page 1307) | This is PartId, a member of class OrderPartData. |
| | PartTotal (see page 1307) | This is PartTotal, a member of class OrderPartData. |
| | ShippingAddressId (see page 1308) | This is ShippingAddressId, a member of class OrderPartData. |
| | ShippingTotal (see page 1308) | This is ShippingTotal, a member of class OrderPartData. |
| | Status (see page 1308) | This is Status, a member of class OrderPartData. |
| | Subtotal (see page 1308) | This is Subtotal, a member of class OrderPartData. |
| | SupplierId (see page 1308) | This is SupplierId, a member of class OrderPartData. |
| | TaxTotal (see page 1309) | This is TaxTotal, a member of class OrderPartData. |
| | TrackingNumber (see page 1309) | This is TrackingNumber, a member of class OrderPartData. |

## OrderPartData.DateShipped Property

**C#**

```
public DateTime DateShipped;
```

**Visual Basic**

```
Public Property DateShipped() As DateTime
```

**Description**

This is DateShipped, a member of class OrderPartData.

## OrderPartData.HandlingTotal Property

**C#**

```
public decimal HandlingTotal;
```

**Visual Basic**

```
Public Property HandlingTotal() As decimal
```

**Description**

This is HandlingTotal, a member of class OrderPartData.

## OrderPartData.Lines Property

**C#**

```
public List<OrderLineData> Lines;
```

**Visual Basic**

```
Public Property Lines() As List<OrderLineData>
```

**Description**

This is Lines, a member of class OrderPartData.

## OrderPartData.OrderId Property

**C#**

```
public long OrderId;
```

**Visual Basic**

```
Public Property OrderId() As long
```

**Description**

This is OrderId, a member of class OrderPartData.

## OrderPartData.PartId Property

**C#**

```
public long PartId;
```

**Visual Basic**

```
Public Property PartId() As long
```

**Description**

This is PartId, a member of class OrderPartData.

## OrderPartData.PartTotal Property

**C#**

```
public decimal PartTotal;
```

**Visual Basic**

```
Public Property PartTotal() As decimal
```

**Description**

This is PartTotal, a member of class OrderPartData.

## OrderPartData.ShippingAddressId Property

**C#**

```
public long ShippingAddressId;
```

**Visual Basic**

```
Public Property ShippingAddressId() As long
```

**Description**

This is ShippingAddressId, a member of class OrderPartData.

## OrderPartData.ShippingTotal Property

**C#**

```
public decimal ShippingTotal;
```

**Visual Basic**

```
Public Property ShippingTotal() As decimal
```

**Description**

This is ShippingTotal, a member of class OrderPartData.

## OrderPartData.Status Property

**C#**

```
public int Status;
```

**Visual Basic**

```
Public Property Status() As Integer
```

**Description**

This is Status, a member of class OrderPartData.

## OrderPartData.Subtotal Property

**C#**

```
public decimal Subtotal;
```

**Visual Basic**

```
Public Property Subtotal() As decimal
```

**Description**

This is Subtotal, a member of class OrderPartData.

## OrderPartData.SupplierId Property

**C#**

```
public long SupplierId;
```

**Visual Basic**

```
Public Property SupplierId() As long
```

**Description**

This is SupplierId, a member of class OrderPartData.

## OrderPartData.TaxTotal Property

**C#**

```
public decimal TaxTotal;
```

**Visual Basic**

```
Public Property TaxTotal() As decimal
```

**Description**

This is TaxTotal, a member of class OrderPartData.

## OrderPartData.TrackingNumber Property

**C#**

```
public string TrackingNumber;
```

**Visual Basic**

```
Public Property TrackingNumber() As string
```

**Description**

This is TrackingNumber, a member of class OrderPartData.

# PackageData Class

Data class used for representing package sizes used for shipping items.

**Class Hierarchy**

**C#**

```
public class PackageData;
```

**Visual Basic**

```
Public Class PackageData
```

**File**

PackageData.cs

# PackageData Constructor

This is the overview for the PackageData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | PackageData.PackageData () (see page 1310) | This is PackageData, a member of class PackageData. |
| | PackageData.PackageData (string, Dimensions, Weight) (see page 1310) | |

| | | |
|---|---|---|
| | PackageData.PackageData (string, Dimensions, Weight, long) (see page 1310) | |

## PackageData.PackageData Constructor ()

**C#**

```
public PackageData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is PackageData, a member of class PackageData.

## PackageData.PackageData Constructor (string, Dimensions, Weight)

**C#**

```
public PackageData(string name, Dimensions dimensions, Weight maxWeight);
```

**Visual Basic**

```
Public Sub New(name As string, dimensions As Dimensions, maxWeight As Weight)
```

**Parameters**

| Parameters | Description |
|---|---|
| name | the name of the package |
| dimensions | dimensions of the package |
| maxWeight | maximum weight capacity of the package |

## PackageData.PackageData Constructor (string, Dimensions, Weight, long)

**C#**

```
public PackageData(string name, Dimensions dimensions, Weight maxWeight, long id);
```

**Visual Basic**

```
Public Sub New(name As string, dimensions As Dimensions, maxWeight As Weight, id As long)
```

**Parameters**

| Parameters | Description |
|---|---|
| name | the name of the package |
| dimensions | dimensions of the package |
| maxWeight | maximum weight capacity of the package |
| id | the Id (see page 1311) of the package. |

# PackageData Properties

The properties of the PackageData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Dimensions (see page 1311) | gets or sets the dimensions of the package. |
| | Id (see page 1311) | Gets or sets the Id of the package. |

| | | |
|---|---|---|
| | MaxWeight (see page 1311) | gets or sets the maximum weight capacity of the package. |
| | Name (see page 1311) | Gets or sets the name of the package. |

## PackageData.Dimensions Property

gets or sets the dimensions of the package.

**C#**

```
public Dimensions Dimensions;
```

**Visual Basic**

```
Public Property Dimensions() As Dimensions
```

## PackageData.Id Property

Gets or sets the Id of the package.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## PackageData.MaxWeight Property

gets or sets the maximum weight capacity of the package.

**C#**

```
public Weight MaxWeight;
```

**Visual Basic**

```
Public Property MaxWeight() As Weight
```

## PackageData.Name Property

Gets or sets the name of the package.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

# PaymentAuthorization Class

Data class representing payment authorization from gateway.

**Class Hierarchy**

**C#**

```
public class PaymentAuthorization;
```

**Visual Basic**

```
Public Class PaymentAuthorization
```

**File**

PaymentAuthorization.cs

# PaymentAuthorization Properties

The properties of the PaymentAuthorization class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AuthorizedOn (see page 1312) | Gets or sets the authorization date. |
| | CapturedOn (see page 1312) | Gets or sets the Capture Date. |
| | CreditedOn (see page 1312) | Gets or sets the Credited Date. |
| | Gateway (see page 1312) | Gets or sets the name of the gateway used for this authorization. |
| | Message (see page 1313) | Gets or sets a success or error message for the authorization. |
| | RecurringTransactionId (see page 1313) | Gets or sets the Transaction id for a recurring transaction. |
| | TransactionId (see page 1313) | Gets or sets the Transaction Id for this authorization. |
| | VoidedOn (see page 1313) | Gets or sets the Voided Date. |

## PaymentAuthorization.AuthorizedOn Property

Gets or sets the authorization date.

**C#**

```
public DateTime AuthorizedOn;
```

**Visual Basic**

```
Public Property AuthorizedOn() As DateTime
```

## PaymentAuthorization.CapturedOn Property

Gets or sets the Capture Date.

**C#**

```
public DateTime CapturedOn;
```

**Visual Basic**

```
Public Property CapturedOn() As DateTime
```

## PaymentAuthorization.CreditedOn Property

Gets or sets the Credited Date.

**C#**

```
public DateTime CreditedOn;
```

**Visual Basic**

```
Public Property CreditedOn() As DateTime
```

## PaymentAuthorization.Gateway Property

Gets or sets the name of the gateway used for this authorization.

**C#**

```
public string Gateway;
```

**Visual Basic**

```
Public Property Gateway() As string
```

### PaymentAuthorization.Message Property

Gets or sets a success or error message for the authorization.

**C#**

```
public string Message;
```

**Visual Basic**

```
Public Property Message() As string
```

### PaymentAuthorization.RecurringTransactionId Property

Gets or sets the Transaction id for a recurring transaction.

**C#**

```
public string RecurringTransactionId;
```

**Visual Basic**

```
Public Property RecurringTransactionId() As string
```

### PaymentAuthorization.TransactionId Property

Gets or sets the Transaction Id for this authorization.

**C#**

```
public string TransactionId;
```

**Visual Basic**

```
Public Property TransactionId() As string
```

### PaymentAuthorization.VoidedOn Property

Gets or sets the Voided Date.

**C#**

```
public DateTime VoidedOn;
```

**Visual Basic**

```
Public Property VoidedOn() As DateTime
```

## PipelineStageData Class

**Class Hierarchy**

**C#**

```
public class PipelineStageData;
```

**Visual Basic**

```
Public Class PipelineStageData
```

**File**

PipelineStageData.cs

# PipelineStageData Properties

The properties of the PipelineStageData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Name (see page 1314) | This is Name, a member of class PipelineStageData. |
| | Sequence (see page 1314) | This is Sequence, a member of class PipelineStageData. |
| | StatusId (see page 1314) | This is StatusId, a member of class PipelineStageData. |

## PipelineStageData.Name Property

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

**Description**

This is Name, a member of class PipelineStageData.

## PipelineStageData.Sequence Property

**C#**

```
public int Sequence;
```

**Visual Basic**

```
Public Property Sequence() As Integer
```

**Description**

This is Sequence, a member of class PipelineStageData.

## PipelineStageData.StatusId Property

**C#**

```
public long StatusId;
```

**Visual Basic**

```
Public Property StatusId() As long
```

**Description**

This is StatusId, a member of class PipelineStageData.

# PostalCodeTaxRateData Class

**Class Hierarchy**

**C#**

```
public class PostalCodeTaxRateData : TaxRateData;
```

**Visual Basic**

```
Public Class PostalCodeTaxRateData
Inherits TaxRateData
```

**File**

PostalCodeTaxRateData.cs

**Description**

This is class Ektron.Cms.Commerce.PostalCodeTaxRateData.

# PostalCodeTaxRateData Constructor

This is the overview for the PostalCodeTaxRateData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | PostalCodeTaxRateData.PostalCodeTaxRateData () (see page 1315) | This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData. |
| | PostalCodeTaxRateData.PostalCodeTaxRateData (string, long, long, decimal) (see page 1315) | This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData. |
| | PostalCodeTaxRateData.PostalCodeTaxRateData (string, long, long, decimal, long) (see page 1316) | This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData. |

## PostalCodeTaxRateData.PostalCodeTaxRateData Constructor ()

**C#**

```
public PostalCodeTaxRateData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData.

## PostalCodeTaxRateData.PostalCodeTaxRateData Constructor (string, long, long, decimal)

**C#**

```
public PostalCodeTaxRateData(string postalCode, long regionId, long taxClassId, decimal rate);
```

**Visual Basic**

```
Public Sub New(postalCode As string, regionId As long, taxClassId As long, rate As decimal)
```

**Description**

This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData.

### PostalCodeTaxRateData.PostalCodeTaxRateData Constructor (string, long, long, decimal, long)

**C#**

```
public PostalCodeTaxRateData(string postalCode, long regionId, long taxClassId, decimal rate,
long id);
```

**Visual Basic**

```
Public Sub New(postalCode As string, regionId As long, taxClassId As long, rate As decimal, id
As long)
```

**Description**

This is PostalCodeTaxRateData, a member of class PostalCodeTaxRateData.

## PostalCodeTaxRateData Properties

The properties of the PostalCodeTaxRateData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | PostalCode (see page 1316) | Gets or sets the postal code associated with this tax rate. |
| | RegionId (see page 1316) | Gets or sets the regionid asociated with this postal code. |

### PostalCodeTaxRateData.PostalCode Property

Gets or sets the postal code associated with this tax rate.

**C#**

```
public string PostalCode;
```

**Visual Basic**

```
Public Property PostalCode() As string
```

### PostalCodeTaxRateData.RegionId Property

Gets or sets the regionid asociated with this postal code.

**C#**

```
public long RegionId;
```

**Visual Basic**

```
Public Property RegionId() As long
```

# PricingData Class

**Class Hierarchy**

Ektron.Cms.Commerce.PricingData

**C#**

```
public class PricingData;
```

**Visual Basic**

```
Public Class PricingData
```

**File**

PricingData.cs

**Description**

This is class Ektron.Cms.Commerce.PricingData.

# PricingData Constructor

This is the overview for the PricingData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | PricingData.PricingData () (see page 1317) | This is PricingData, a member of class PricingData. |
| | PricingData.PricingData (List<CurrencyPricingData>) (see page 1317) | This is PricingData, a member of class PricingData. |
| | PricingData.PricingData (long, decimal, decimal) (see page 1317) | Creates a PricingData with a single CurrencyPricingData (see page 1252) that contains a single tierprice with the supplied sale price. |

## PricingData.PricingData Constructor ()

**C#**

```
public PricingData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is PricingData, a member of class PricingData.

## PricingData.PricingData Constructor (List<CurrencyPricingData>)

**C#**

```
public PricingData(List<CurrencyPricingData> currencyPricelist);
```

**Visual Basic**

```
Public Sub New(currencyPricelist As List<CurrencyPricingData>)
```

**Description**

This is PricingData, a member of class PricingData.

## PricingData.PricingData Constructor (long, decimal, decimal)

Creates a PricingData with a single CurrencyPricingData (see page 1252) that contains a single tierprice with the supplied sale price.

**C#**

```
public PricingData(long currencyId, decimal salePrice, decimal listPrice);
```

**Visual Basic**

```
Public Sub New(currencyId As long, salePrice As decimal, listPrice As decimal)
```

**Parameters**

| Parameters | Description |
|---|---|
| currencyId | Id of currency to create pricing for. |
| salePrice | sale Price for single tier item. |
| listPrice | List Price for currency. |

# PricingData Methods

The methods of the PricingData class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | GetCurrencyById (see page 1318) | This is GetCurrencyById, a member of class PricingData. |
| | GetCurrencyByIso (see page 1318) | This is GetCurrencyByIso, a member of class PricingData. |
| | GetCurrencyByIsoNumber (see page 1319) | This is GetCurrencyByIsoNumber, a member of class PricingData. |
| | GetListPrice (see page 1319) | This is GetListPrice, a member of class PricingData. |
| | GetListPriceByCurrencyId (see page 1319) | This is GetListPriceByCurrencyId, a member of class PricingData. |
| | GetListPriceByIsoNumber (see page 1319) | This is GetListPriceByIsoNumber, a member of class PricingData. |
| | GetSalePrice (see page 1319) | This is GetSalePrice, a member of class PricingData. |
| | GetSalePriceByCurrencyId (see page 1320) | This is GetSalePriceByCurrencyId, a member of class PricingData. |
| | GetSalePriceByIsoNumber (see page 1320) | This is GetSalePriceByIsoNumber, a member of class PricingData. |
| | Validate (see page 1320) | This is Validate, a member of class PricingData. |

## PricingData.GetCurrencyById Method

**C#**

```
public CurrencyPricingData GetCurrencyById(long currencyId);
```

**Visual Basic**

```
Public Function GetCurrencyById(currencyId As long) As CurrencyPricingData
```

**Description**

This is GetCurrencyById, a member of class PricingData.

## PricingData.GetCurrencyByIso Method

**C#**

```
public CurrencyPricingData GetCurrencyByIso(string alphaIsoCode);
```

**Visual Basic**

```
Public Function GetCurrencyByIso(alphaIsoCode As string) As CurrencyPricingData
```

**Description**

This is GetCurrencyByIso, a member of class PricingData.

## PricingData.GetCurrencyByIsoNumber Method

**C#**

```
public CurrencyPricingData GetCurrencyByIsoNumber(int numericIsoCode);
```

**Visual Basic**

```
Public Function GetCurrencyByIsoNumber(numericIsoCode As Integer) As CurrencyPricingData
```

**Description**

This is GetCurrencyByIsoNumber, a member of class PricingData.

## PricingData.GetListPrice Method

**C#**

```
public decimal GetListPrice(string alphaIsoCode);
```

**Visual Basic**

```
Public Function GetListPrice(alphaIsoCode As string) As decimal
```

**Description**

This is GetListPrice, a member of class PricingData.

## PricingData.GetListPriceByCurrencyId Method

**C#**

```
public decimal GetListPriceByCurrencyId(long currencyId);
```

**Visual Basic**

```
Public Function GetListPriceByCurrencyId(currencyId As long) As decimal
```

**Description**

This is GetListPriceByCurrencyId, a member of class PricingData.

## PricingData.GetListPriceByIsoNumber Method

**C#**

```
public decimal GetListPriceByIsoNumber(int numericIsoCode);
```

**Visual Basic**

```
Public Function GetListPriceByIsoNumber(numericIsoCode As Integer) As decimal
```

**Description**

This is GetListPriceByIsoNumber, a member of class PricingData.

## PricingData.GetSalePrice Method

**C#**

```
public decimal GetSalePrice(int quantity, string alphaIsoCode);
```

**Visual Basic**

```
Public Function GetSalePrice(quantity As Integer, alphaIsoCode As string) As decimal
```

**Description**

This is GetSalePrice, a member of class PricingData.

## PricingData.GetSalePriceByCurrencyId Method

**C#**

```
public decimal GetSalePriceByCurrencyId(int quantity, long currencyId);
```

**Visual Basic**

```
Public Function GetSalePriceByCurrencyId(quantity As Integer, currencyId As long) As decimal
```

**Description**

This is GetSalePriceByCurrencyId, a member of class PricingData.

## PricingData.GetSalePriceByIsoNumber Method

**C#**

```
public decimal GetSalePriceByIsoNumber(int quantity, int numericIsoCode);
```

**Visual Basic**

```
Public Function GetSalePriceByIsoNumber(quantity As Integer, numericIsoCode As Integer) As
decimal
```

**Description**

This is GetSalePriceByIsoNumber, a member of class PricingData.

## PricingData.Validate Method

**C#**

```
public ValidationResults Validate();
```

**Visual Basic**

```
Public Function Validate() As ValidationResults
```

**Description**

This is Validate, a member of class PricingData.

# PricingData Properties

The properties of the PricingData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CurrencyPricelist (see page 1320) | Gets list of CurrencyPriceData objects associated with this PricingData (see page 1316) object. |
| | IsRecurringPrice (see page 1321) | Gets the Is Recurring Price flag. If true, this price is for recurring billing. |
| | Recurrence (see page 1321) | Gets or sets the recurrence pattern for this price if it supports recurring billing. |

## PricingData.CurrencyPricelist Property

Gets list of CurrencyPriceData objects associated with this PricingData (see page 1316) object.

**C#**

```
public List<Ektron.Cms.Commerce.CurrencyPricingData> CurrencyPricelist;
```

**Visual Basic**

```
Public Property CurrencyPricelist() As List<Ektron.Cms.Commerce.CurrencyPricingData>
```

## PricingData.IsRecurringPrice Property

Gets the Is Recurring Price flag. If true, this price is for recurring billing.

**C#**

```
public bool IsRecurringPrice;
```

**Visual Basic**

```
Public Property IsRecurringPrice() As bool
```

## PricingData.Recurrence Property

Gets or sets the recurrence pattern for this price if it supports recurring billing.

**C#**

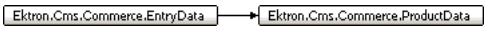
```
public RecurrenceData Recurrence;
```

**Visual Basic**

```
Public Property Recurrence() As RecurrenceData
```

# ProductData Class

**Class Hierarchy**




**C#**

```
public class ProductData : EntryData;
```

**Visual Basic**

```
Public Class ProductData
Inherits EntryData
```

**File**

ProductData.cs

# ProductData Constructor

This is the overview for the ProductData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | ProductData.ProductData () (see page 1321) | This is ProductData, a member of class ProductData. |
| | ProductData.ProductData (int) (see page 1322) | This is ProductData, a member of class ProductData. |

## ProductData.ProductData Constructor ()

**C#**

```
public ProductData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is ProductData, a member of class ProductData.

### ProductData.ProductData Constructor (int)

**C#**

```
public ProductData(int languageId);
```

**Visual Basic**

```
Public Sub New(languageId As Integer)
```

**Description**

This is ProductData, a member of class ProductData.

## ProductData Properties

The properties of the ProductData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Variants (see page 1322) | This is Variants, a member of class ProductData. |

### ProductData.Variants Property

**C#**

```
public List<ProductVariantData> Variants;
```

**Visual Basic**

```
Public Property Variants() As List<ProductVariantData>
```

**Description**

This is Variants, a member of class ProductData.

## ProductTypeAttributeData Class

**Class Hierarchy**

Ektron.Cms.Commerce.ProductTypeAttributeData

**C#**

```
public class ProductTypeAttributeData;
```

**Visual Basic**

```
Public Class ProductTypeAttributeData
```

**File**

ProductAttributeData.cs

**Description**

This is class Ektron.Cms.Commerce.ProductTypeAttributeData.

# ProductTypeAttributeData Constructor

This is the overview for the ProductTypeAttributeData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | ProductTypeAttributeData.ProductTypeAttributeData () (see page 1323) | This is ProductTypeAttributeData, a member of class ProductTypeAttributeData. |
| | ProductTypeAttributeData.ProductTypeAttributeData (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus) (see page 1323) | This is ProductTypeAttributeData, a member of class ProductTypeAttributeData. |

## ProductTypeAttributeData.ProductTypeAttributeData Constructor ()

**C#**

```
public ProductTypeAttributeData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is ProductTypeAttributeData, a member of class ProductTypeAttributeData.

## ProductTypeAttributeData.ProductTypeAttributeData Constructor (string, int, EkEnumeration.ProductTypeAttributeDataType, object, EkEnumeration.ProductTypeAttributeStatus)

**C#**

```
public ProductTypeAttributeData(string name, int displayOrder,
EkEnumeration.ProductTypeAttributeDataType dataType, object defaultValue,
EkEnumeration.ProductTypeAttributeStatus activeStatus);
```

**Visual Basic**

```
Public Sub New(name As string, displayOrder As Integer, dataType As
EkEnumeration.ProductTypeAttributeDataType, defaultValue As object, activeStatus As
EkEnumeration.ProductTypeAttributeStatus)
```

**Description**

This is ProductTypeAttributeData, a member of class ProductTypeAttributeData.

# ProductTypeAttributeData Properties

The properties of the ProductTypeAttributeData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | ActiveStatus (see page 1324) | This is ActiveStatus, a member of class ProductTypeAttributeData. |
| | DataType (see page 1324) | This is DataType, a member of class ProductTypeAttributeData. |

| | | |
|---|---|---|
| | DefaultValue (see page 1324) | This is DefaultValue, a member of class ProductTypeAttributeData. |
| | DisplayOrder (see page 1324) | This is DisplayOrder, a member of class ProductTypeAttributeData. |
| | Id (see page 1324) | This is Id, a member of class ProductTypeAttributeData. |
| | Name (see page 1325) | This is Name, a member of class ProductTypeAttributeData. |

## ProductTypeAttributeData.ActiveStatus Property

**C#**

```
public EkEnumeration.ProductTypeAttributeStatus ActiveStatus;
```

**Visual Basic**

```
Public Property ActiveStatus() As EkEnumeration.ProductTypeAttributeStatus
```

**Description**

This is ActiveStatus, a member of class ProductTypeAttributeData.

## ProductTypeAttributeData.DataType Property

**C#**

```
public EkEnumeration.ProductTypeAttributeDataType DataType;
```

**Visual Basic**

```
Public Property DataType() As EkEnumeration.ProductTypeAttributeDataType
```

**Description**

This is DataType, a member of class ProductTypeAttributeData.

## ProductTypeAttributeData.DefaultValue Property

**C#**

```
public object DefaultValue;
```

**Visual Basic**

```
Public Property DefaultValue() As object
```

**Description**

This is DefaultValue, a member of class ProductTypeAttributeData.

## ProductTypeAttributeData.DisplayOrder Property

**C#**

```
public int DisplayOrder;
```

**Visual Basic**

```
Public Property DisplayOrder() As Integer
```

**Description**

This is DisplayOrder, a member of class ProductTypeAttributeData.

## ProductTypeAttributeData.Id Property

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Description**

This is Id, a member of class ProductTypeAttributeData.

### ProductTypeAttributeData.Name Property

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

**Description**

This is Name, a member of class ProductTypeAttributeData.

## ProductTypeData Class

Data class representing ecommerce product types.

**Class Hierarchy**

**C#**

```
[Serializable]
public class ProductTypeData : XmlConfigData;
```

**Visual Basic**

```
<Serializable>
Public Class ProductTypeData
Inherits XmlConfigData
```

**File**

ProductTypeData.cs

## ProductTypeData Properties

The properties of the ProductTypeData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Attributes (see page 1325) | This is Attributes, a member of class ProductTypeData. |
| | DefaultThumbnails (see page 1326) | This is DefaultThumbnails, a member of class ProductTypeData. |
| | EntryClass (see page 1326) | This is EntryClass, a member of class ProductTypeData. |

### ProductTypeData.Attributes Property

**C#**

```
public List<ProductTypeAttributeData> Attributes;
```

**Visual Basic**

```
Public Property Attributes() As List<ProductTypeAttributeData>
```

**Description**

This is Attributes, a member of class ProductTypeData.

### ProductTypeData.DefaultThumbnails Property

**C#**

```
public List<ThumbnailDefaultData> DefaultThumbnails;
```

**Visual Basic**

```
Public Property DefaultThumbnails() As List<ThumbnailDefaultData>
```

**Description**

This is DefaultThumbnails, a member of class ProductTypeData.

### ProductTypeData.EntryClass Property

**C#**

```
public EkEnumeration.CatalogEntryType EntryClass;
```

**Visual Basic**

```
Public Property EntryClass() As EkEnumeration.CatalogEntryType
```

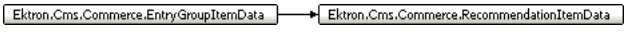
**Description**

This is EntryClass, a member of class ProductTypeData.

## RecommendationItemData Class

Data class representing entry reccomendations in ecommerce system.

**Class Hierarchy**




**C#**

```
public class RecommendationItemData : EntryGroupItemData;
```

**Visual Basic**

```
Public Class RecommendationItemData
Inherits EntryGroupItemData
```

**File**

RecommendationItemData.cs

## RecommendationItemData Constructor

This is the overview for the RecommendationItemData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | RecommendationItemData.RecommendationItemData () (see page 1327) | This is RecommendationItemData, a member of class RecommendationItemData. |
| | RecommendationItemData.RecommendationItemData (long, long, long) (see page 1327) | This is RecommendationItemData, a member of class RecommendationItemData. |
| | RecommendationItemData.RecommendationItemData (long, long, long, EkEnumeration.RecommendationType) (see page 1327) | |

## RecommendationItemData.RecommendationItemData Constructor ()

**C#**

```
public RecommendationItemData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is RecommendationItemData, a member of class RecommendationItemData.

## RecommendationItemData.RecommendationItemData Constructor (long, long, long)

**C#**

```
private RecommendationItemData(long id, long groupId, long entryId);
```

**Visual Basic**

```
Private Sub New(id As long, groupId As long, entryId As long)
```

**Description**

This is RecommendationItemData, a member of class RecommendationItemData.

## RecommendationItemData.RecommendationItemData Constructor (long, long, long, EkEnumeration.RecommendationType)

**C#**

```
public RecommendationItemData(long id, long entryId, long entryLanguage,
EkEnumeration.RecommendationType recommendationType);
```

**Visual Basic**

```
Public Sub New(id As long, entryId As long, entryLanguage As long, recommendationType As
EkEnumeration.RecommendationType)
```

**Parameters**

| Parameters | Description |
|---|---|
| id | id of reccomendation. |
| entryId | entry id associated with recommendation. |
| entryLanguage | language of entryId. |
| recommendationType | type of recommendation. |

# RecommendationItemData Properties

The properties of the RecommendationItemData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | EntryLanguage (see page 1328) | Gets or sets the language of the entry. |
| | EntryType (see page 1328) | Gets or sets the entry type associated with this recommendation. |
| | Html (see page 1328) | Gets or sets the Html of the recommendation. |
| | Image (see page 1328) | Gets or sets the image associated with the recommendation. |
| | ImageThumbnail (see page 1329) | Gets or sets the image thumbnail associated with the recommendation. |
| | Pricing (see page 1329) | The entry pricing, including currencies and quantity. |
| | Quicklink (see page 1329) | Gets or sets the quiclink of the recommendation. |
| | Summary (see page 1329) | Gets or sets the summary of the recommendation. |
| | Title (see page 1329) | Gets or sets the title of the recommendation. |
| | Type (see page 1330) | Gets or sets the type of reccomendation. |

## RecommendationItemData.EntryLanguage Property

Gets or sets the language of the entry.

**C#**

```
public long EntryLanguage;
```

**Visual Basic**

```
Public Property EntryLanguage() As long
```

## RecommendationItemData.EntryType Property

Gets or sets the entry type associated with this recommendation.

**C#**

```
public EkEnumeration.CatalogEntryType EntryType;
```

**Visual Basic**

```
Public Property EntryType() As EkEnumeration.CatalogEntryType
```

## RecommendationItemData.Html Property

Gets or sets the Html of the recommendation.

**C#**

```
public string Html;
```

**Visual Basic**

```
Public Property Html() As string
```

## RecommendationItemData.Image Property

Gets or sets the image associated with the recommendation.

**C#**

```
public string Image;
```

**Visual Basic**

```
Public Property Image() As string
```

## RecommendationItemData.ImageThumbnail Property

Gets or sets the image thumbnail associated with the recommendation.

**C#**

```
public string ImageThumbnail;
```

**Visual Basic**

```
Public Property ImageThumbnail() As string
```

## RecommendationItemData.Pricing Property

The entry pricing, including currencies and quantity.

**C#**

```
public PricingData Pricing;
```

**Visual Basic**

```
Public Property Pricing() As PricingData
```

**Returns**

PricingData (see page 1316) of the catalog entry.

## RecommendationItemData.Quicklink Property

Gets or sets the quiclink of the recommendation.

**C#**

```
public string Quicklink;
```

**Visual Basic**

```
Public Property Quicklink() As string
```

## RecommendationItemData.Summary Property

Gets or sets the summary of the recommendation.

**C#**

```
public string Summary;
```

**Visual Basic**

```
Public Property Summary() As string
```

## RecommendationItemData.Title Property

Gets or sets the title of the recommendation.

**C#**

```
public string Title;
```

**Visual Basic**

```
Public Property Title() As string
```

## RecommendationItemData.Type Property

Gets or sets the type of reccomendation.

**C#**

```
public EkEnumeration.RecommendationType Type;
```

**Visual Basic**

```
Public Property Type() As EkEnumeration.RecommendationType
```

# RegionData Class

Data class for region

**Class Hierarchy**

Ektron.Cms.Commerce.RegionData

**C#**

```
public class RegionData;
```

**Visual Basic**

```
Public Class RegionData
```

**File**

RegionData.cs

# RegionData Constructor

This is the overview for the RegionData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | RegionData.RegionData () (see page 1330) | This is RegionData, a member of class RegionData. |
| | RegionData.RegionData (string, long, string, bool) (see page 1331) | |

## RegionData.RegionData Constructor ()

**C#**

```
public RegionData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is RegionData, a member of class RegionData.

## RegionData.RegionData Constructor (string, long, string, bool)

**C#**

```
public RegionData(string name, long countryId, string code, bool enabled);
```

**Visual Basic**

```
Public Sub New(name As string, countryId As long, code As string, enabled As bool)
```

**Parameters**

| Parameters | Description |
|---|---|
| name | name of the region. |
| countryId | Id (see page 1332) of the country the region belongs to. |
| code | code of the region. |
| enabled | indicates whether or not the region is enabled for use. |

# RegionData Methods

The methods of the RegionData class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | ToString (see page 1331) | Returns Name (see page 1332) of Region. |
| | Validate (see page 1331) | Validates the RegionData (see page 1330) object is valid for saving. |

## RegionData.ToString Method

Returns Name (see page 1332) of Region.

**C#**

```
public override string ToString();
```

**Visual Basic**

```
Public override Function ToString() As string
```

## RegionData.Validate Method

Validates the RegionData (see page 1330) object is valid for saving.

**C#**

```
public ValidationResults Validate();
```

**Visual Basic**

```
Public Function Validate() As ValidationResults
```

**Returns**

Returns a ValidationResult object with any validation errors.

# RegionData Properties

The properties of the RegionData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Code (see page 1332) | Gets or sets the code of the region. |

| | | |
|---|---|---|
| | CountryId (see page 1332) | Gets or sets the Id (see page 1332) of the country the region belongs to. |
| | Enabled (see page 1332) | Gets or sets whether or not the region is enabled for use. |
| | Id (see page 1332) | Gets or sets the Id of the region. |
| | Name (see page 1332) | Gets or sets the name of the region. |

## RegionData.Code Property

Gets or sets the code of the region.

**C#**

```
public string Code;
```

**Visual Basic**

```
Public Property Code() As string
```

## RegionData.CountryId Property

Gets or sets the Id (see page 1332) of the country the region belongs to.

**C#**

```
public long CountryId;
```

**Visual Basic**

```
Public Property CountryId() As long
```

## RegionData.Enabled Property

Gets or sets whether or not the region is enabled for use.

**C#**

```
public bool Enabled;
```

**Visual Basic**

```
Public Property Enabled() As bool
```

## RegionData.Id Property

Gets or sets the Id of the region.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## RegionData.Name Property

Gets or sets the name of the region.

**C#**

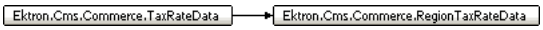
```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

# RegionTaxRateData Class

**Class Hierarchy**




**C#**

```
public class RegionTaxRateData : TaxRateData;
```

**Visual Basic**

```
Public Class RegionTaxRateData
Inherits TaxRateData
```

**File**

RegionTaxRateData.cs

**Description**

This is class Ektron.Cms.Commerce.RegionTaxRateData.

# RegionTaxRateData Constructor

This is the overview for the RegionTaxRateData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | RegionTaxRateData.RegionTaxRateData () (see page 1333) | This is RegionTaxRateData, a member of class RegionTaxRateData. |
| | RegionTaxRateData.RegionTaxRateData (long, long, decimal) (see page 1333) | This is RegionTaxRateData, a member of class RegionTaxRateData. |
| | RegionTaxRateData.RegionTaxRateData (long, long, decimal, long) (see page 1334) | This is RegionTaxRateData, a member of class RegionTaxRateData. |

## RegionTaxRateData.RegionTaxRateData Constructor ()

**C#**

```
public RegionTaxRateData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is RegionTaxRateData, a member of class RegionTaxRateData.

## RegionTaxRateData.RegionTaxRateData Constructor (long, long, decimal)

**C#**

```
public RegionTaxRateData(long regionId, long taxClassId, decimal rate);
```

**Visual Basic**

```
Public Sub New(regionId As long, taxClassId As long, rate As decimal)
```

**Description**

This is RegionTaxRateData, a member of class RegionTaxRateData.

## RegionTaxRateData.RegionTaxRateData Constructor (long, long, decimal, long)

**C#**

```
public RegionTaxRateData(long regionId, long taxClassId, decimal rate, long id);
```

**Visual Basic**

```
Public Sub New(regionId As long, taxClassId As long, rate As decimal, id As long)
```

**Description**

This is RegionTaxRateData, a member of class RegionTaxRateData.

# RegionTaxRateData Properties

The properties of the RegionTaxRateData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | RegionData (see page 1334) | Gets or sets the RegionData object associated with this tax rate. |

## RegionTaxRateData.RegionData Property

Gets or sets the RegionData object associated with this tax rate.

**C#**

```
public RegionData RegionData;
```

**Visual Basic**

```
Public Property RegionData() As RegionData
```

# ShippingBaseData Class

Base data class for representing shipping method information.

**Class Hierarchy**

Ektron.Cms.Commerce.ShippingBaseData

**C#**

```
public class ShippingBaseData;
```

**Visual Basic**

```
Public Class ShippingBaseData
```

**File**

ShippingBaseData.cs

## ShippingBaseData.ShippingBaseData Constructor

**C#**

```
public ShippingBaseData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is ShippingBaseData, a member of class ShippingBaseData.

# ShippingBaseData Properties

The properties of the ShippingBaseData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Id (see page 1335) | gets or sets the id of the the shipping data. |
| | Name (see page 1335) | Gets or sets the name of the the shipping data |
| | ProviderService (see page 1335) | Gets or sets the name of the provider service for the shipping data. |

## ShippingBaseData.Id Property

gets or sets the id of the the shipping data.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## ShippingBaseData.Name Property

Gets or sets the name of the the shipping data

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

## ShippingBaseData.ProviderService Property

Gets or sets the name of the provider service for the shipping data.

**C#**

```
public string ProviderService;
```

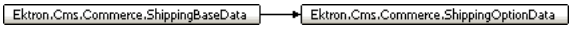
**Visual Basic**

```
Public Property ProviderService() As string
```

# ShippingOptionData Class

Data class representing a quote for a particular shipping method.

**Class Hierarchy**



**C#**

```
public class ShippingOptionData : ShippingBaseData;
```

**Visual Basic**

```
Public Class ShippingOptionData
Inherits ShippingBaseData
```

**File**

ShippingFeeData.cs

## ShippingOptionData.ShippingOptionData Constructor

**C#**

```
public ShippingOptionData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is ShippingOptionData, a member of class ShippingOptionData.

## ShippingOptionData Properties

The properties of the ShippingOptionData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CurrencyId (see page 1336) | Gets or sets the currency associated with this shipping Fee. |
| | ShippingFee (see page 1336) | Gets or sets the fee for this particular shipping option. |

### ShippingOptionData.CurrencyId Property

Gets or sets the currency associated with this shipping Fee.

**C#**

```
public int CurrencyId;
```

**Visual Basic**

```
Public Property CurrencyId() As Integer
```

### ShippingOptionData.ShippingFee Property

Gets or sets the fee for this particular shipping option.

**C#**

```
public decimal ShippingFee;
```

**Visual Basic**

```
Public Property ShippingFee() As decimal
```

# TaxClassData Class

Data object representing tax class information.

**Class Hierarchy**

**C#**

```
public class TaxClassData;
```

**Visual Basic**

```
Public Class TaxClassData
```

**File**

TaxClassData.cs

# TaxClassData Constructor

This is the overview for the TaxClassData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | TaxClassData.TaxClassData () (see page 1337) | This is TaxClassData, a member of class TaxClassData. |
| | TaxClassData.TaxClassData (string) (see page 1337) | |
| | TaxClassData.TaxClassData (string, long) (see page 1338) | |

## TaxClassData.TaxClassData Constructor ()

**C#**

```
public TaxClassData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is TaxClassData, a member of class TaxClassData.

## TaxClassData.TaxClassData Constructor (string)

**C#**

```
public TaxClassData(string name);
```

**Visual Basic**

```
Public Sub New(name As string)
```

**Parameters**

| Parameters | Description |
|---|---|
| name | name of the tax class |

## TaxClassData.TaxClassData Constructor (string, long)

**C#**

```
public TaxClassData(string name, long id);
```

**Visual Basic**

```
Public Sub New(name As string, id As long)
```

**Parameters**

| Parameters | Description |
|---|---|
| name | name of the tax class |
| id | name of the tax class |

# TaxClassData Properties

The properties of the TaxClassData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Id (see page 1338) | Gets or sets the name of the tax class |
| | Name (see page 1338) | Gets or sets the name of the tax class. |

## TaxClassData.Id Property

Gets or sets the name of the tax class

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## TaxClassData.Name Property

Gets or sets the name of the tax class.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

# TaxRateData Class

**Class Hierarchy**

**C#**

```
public class TaxRateData;
```

**Visual Basic**

```
Public Class TaxRateData
```

**File**

TaxRateData.cs

**Description**

This is class Ektron.Cms.Commerce.TaxRateData.

# TaxRateData Constructor

This is the overview for the TaxRateData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | TaxRateData.TaxRateData () (see page 1339) | This is TaxRateData, a member of class TaxRateData. |
| | TaxRateData.TaxRateData (long, long, long, decimal) (see page 1339) | This is TaxRateData, a member of class TaxRateData. |
| | TaxRateData.TaxRateData (long, long, long, decimal, long) (see page 1339) | This is TaxRateData, a member of class TaxRateData. |

## TaxRateData.TaxRateData Constructor ()

**C#**

```
public TaxRateData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is TaxRateData, a member of class TaxRateData.

## TaxRateData.TaxRateData Constructor (long, long, long, decimal)

**C#**

```
public TaxRateData(long typeId, long typeItemId, long taxClassId, decimal rate);
```

**Visual Basic**

```
Public Sub New(typeId As long, typeItemId As long, taxClassId As long, rate As decimal)
```

**Description**

This is TaxRateData, a member of class TaxRateData.

## TaxRateData.TaxRateData Constructor (long, long, long, decimal, long)

**C#**

```
public TaxRateData(long typeId, long typeItemId, long taxClassId, decimal rate, long id);
```

**Visual Basic**

```
Public Sub New(typeId As long, typeItemId As long, taxClassId As long, rate As decimal, id As
long)
```

### Description

This is TaxRateData, a member of class TaxRateData.

# TaxRateData Properties

The properties of the TaxRateData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Id (see page 1340) | Gets or sets the Id of the TaxRate. |
| | Rate (see page 1340) | Gets or sets the actual rate for this TaxRate object. Should be entered in decimal format - i.e. .075 for 7.5% |
| | TaxClassId (see page 1340) | Gets or sets the Id (see page 1340) of the tax class that this rate applies to. |
| | TypeId (see page 1340) | Gets or sets the type of the Tax Rate (see page 1340). |
| | TypeItemId (see page 1341) | Gets or sets the Id (see page 1340) of the tax type object this tax rate applies to. For instance, for a region sales tax, TypeItemId would be the Id (see page 1340) of the corresponding region. |

## TaxRateData.Id Property

Gets or sets the Id of the TaxRate.

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## TaxRateData.Rate Property

Gets or sets the actual rate for this TaxRate object. Should be entered in decimal format - i.e. .075 for 7.5%

**C#**

```
public decimal Rate;
```

**Visual Basic**

```
Public Property Rate() As decimal
```

## TaxRateData.TaxClassId Property

Gets or sets the Id (see page 1340) of the tax class that this rate applies to.

**C#**

```
public long TaxClassId;
```

**Visual Basic**

```
Public Property TaxClassId() As long
```

## TaxRateData.TypeId Property

Gets or sets the type of the Tax Rate (see page 1340).

**C#**

```
public long TypeId;
```

**Visual Basic**

```
Public Property TypeId() As long
```

## TaxRateData.TypeItemId Property

Gets or sets the Id (see page 1340) of the tax type object this tax rate applies to. For instance, for a region sales tax, TypeItemId would be the Id (see page 1340) of the corresponding region.

**C#**

```
public long TypeItemId;
```

**Visual Basic**

```
Public Property TypeItemId() As long
```

# TaxTypeData Class

Data class representing Tax Types.

**Class Hierarchy**

**C#**

```
public class TaxTypeData;
```

**Visual Basic**

```
Public Class TaxTypeData
```

**File**

TaxTypeData.cs

## TaxTypeData Constructor

This is the overview for the TaxTypeData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | TaxTypeData.TaxTypeData () (see page 1341) | This is TaxTypeData, a member of class TaxTypeData. |
| | TaxTypeData.TaxTypeData (string, int) (see page 1342) | |
| | TaxTypeData.TaxTypeData (string, int, long) (see page 1342) | |

## TaxTypeData.TaxTypeData Constructor ()

**C#**

```
public TaxTypeData();
```

**Visual Basic**

```
Public Sub New()
```

### Description

This is TaxTypeData, a member of class TaxTypeData.

## TaxTypeData.TaxTypeData Constructor (string, int)

### C#

```
public TaxTypeData(string name, int precedence);
```

### Visual Basic

```
Public Sub New(name As string, precedence As Integer)
```

### Parameters

| Parameters | Description |
|---|---|
| name | name of the tax class |
| precedence | Gets or sets the precedence of the tax class. This is the priority order in which it will be applied. For example, if type "Postal Code Sales Tax is priority 1" and it exists for an address it will be used over type "State Sales Tax" which is priority 2. |

## TaxTypeData.TaxTypeData Constructor (string, int, long)

### C#

```
public TaxTypeData(string name, int precedence, long id);
```

### Visual Basic

```
Public Sub New(name As string, precedence As Integer, id As long)
```

### Parameters

| Parameters | Description |
|---|---|
| name | name of the tax class |
| precedence | Gets or sets the precedence of the tax class. This is the priority order in which it will be applied. For example, if type "Postal Code Sales Tax is priority 1" and it exists for an address it will be used over type "State Sales Tax" which is priority 2. |
| id | name of the tax class |

# TaxTypeData Properties

The properties of the TaxTypeData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Id (see page 1342) | Gets or sets the name of the tax class |
| | Name (see page 1343) | Gets or sets the name of the tax class. |
| | Precedence (see page 1343) | Gets or sets the precedence of the tax class. This is the priority order in which it will be applied. For example, if type "Postal Code Sales Tax is priority 1" and it exists for an address it will be used over type "State Sales Tax" which is priority 2. |

## TaxTypeData.Id Property

Gets or sets the name of the tax class

### C#

```
public long Id;
```

### Visual Basic

```
Public Property Id() As long
```

## TaxTypeData.Name Property

Gets or sets the name of the tax class.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

## TaxTypeData.Precedence Property

Gets or sets the precedence of the tax class. This is the priority order in which it will be applied. For example, if type "Postal Code Sales Tax is priority 1" and it exists for an address it will be used over type "State Sales Tax" which is priority 2.

**C#**

```
public int Precedence;
```

**Visual Basic**

```
Public Property Precedence() As Integer
```

# ThumbnailData Class

**Class Hierarchy**

**C#**

```
public class ThumbnailData : MediaData, IMediaThumbnailData;
```

**Visual Basic**

```
Public Class ThumbnailData
Inherits MediaData
Inherits IMediaThumbnailData
```

**File**

ThumbnailData.cs

**Description**

This is class Ektron.Cms.ThumbnailData.

# ThumbnailData Properties

The properties of the ThumbnailData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Height (see page 1344) | This is Height, a member of class ThumbnailData. |
| | Width (see page 1344) | This is Width, a member of class ThumbnailData. |

## ThumbnailData.Height Property

**C#**

```
public int Height;
```

**Visual Basic**

```
Public Property Height() As Integer
```

**Description**

This is Height, a member of class ThumbnailData.

## ThumbnailData.Width Property

**C#**

```
public int Width;
```

**Visual Basic**

```
Public Property Width() As Integer
```

**Description**

This is Width, a member of class ThumbnailData.

# ThumbnailDefaultData Class

**Class Hierarchy**

**C#**

```
public class ThumbnailDefaultData : IMediaThumbnailDefault;
```

**Visual Basic**

```
Public Class ThumbnailDefaultData
Inherits IMediaThumbnailDefault
```

**File**

ThumbnailDefaultData.cs

**Description**

This is class Ektron.Cms.ThumbnailDefaultData.

# ThumbnailDefaultData Properties

The properties of the ThumbnailDefaultData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | FileNameFormat (see page 1345) | This is FileNameFormat, a member of class ThumbnailDefaultData. |
| | Height (see page 1345) | This is Height, a member of class ThumbnailDefaultData. |
| | Id (see page 1345) | This is Id, a member of class ThumbnailDefaultData. |
| | MimeType (see page 1345) | This is MimeType, a member of class ThumbnailDefaultData. |
| | Title (see page 1345) | This is Title, a member of class ThumbnailDefaultData. |

| | | |
|---|---|---|
| | Width (see page 1346) | This is Width, a member of class ThumbnailDefaultData. |

## ThumbnailDefaultData.FileNameFormat Property

**C#**

```
public string FileNameFormat;
```

**Visual Basic**

```
Public Property FileNameFormat() As string
```

**Description**

This is FileNameFormat, a member of class ThumbnailDefaultData.

## ThumbnailDefaultData.Height Property

**C#**

```
public int Height;
```

**Visual Basic**

```
Public Property Height() As Integer
```

**Description**

This is Height, a member of class ThumbnailDefaultData.

## ThumbnailDefaultData.Id Property

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Description**

This is Id, a member of class ThumbnailDefaultData.

## ThumbnailDefaultData.MimeType Property

**C#**

```
public string MimeType;
```

**Visual Basic**

```
Public Property MimeType() As string
```

**Description**

This is MimeType, a member of class ThumbnailDefaultData.

## ThumbnailDefaultData.Title Property

**C#**

```
public string Title;
```

**Visual Basic**

```
Public Property Title() As string
```

**Description**

This is Title, a member of class ThumbnailDefaultData.

## ThumbnailDefaultData.Width Property

**C#**

```
public int Width;
```

**Visual Basic**

```
Public Property Width() As Integer
```

**Description**

This is Width, a member of class ThumbnailDefaultData.

# TierPriceData Class

**Class Hierarchy**

**C#**

```
public class TierPriceData : CmsData<TierPriceData>, IComparable<TierPriceData>;
```

**Visual Basic**

```
Public Class TierPriceData
Inherits CmsData
Inherits IComparable
```

**File**

TierPriceData.cs

**Description**

This is class Ektron.Cms.Commerce.TierPriceData.

# TierPriceData Constructor

This is the overview for the TierPriceData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | TierPriceData.TierPriceData () (see page 1347) | This is TierPriceData, a member of class TierPriceData. |
| | TierPriceData.TierPriceData (int, decimal) (see page 1347) | This is TierPriceData, a member of class TierPriceData. |
| | TierPriceData.TierPriceData (int, decimal, long) (see page 1347) | This is TierPriceData, a member of class TierPriceData. |

## TierPriceData.TierPriceData Constructor ()

**C#**

```
public TierPriceData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is TierPriceData, a member of class TierPriceData.

## TierPriceData.TierPriceData Constructor (int, decimal)

**C#**

```
public TierPriceData(int quantity, decimal salePrice);
```

**Visual Basic**

```
Public Sub New(quantity As Integer, salePrice As decimal)
```

**Description**

This is TierPriceData, a member of class TierPriceData.

## TierPriceData.TierPriceData Constructor (int, decimal, long)

**C#**

```
public TierPriceData(int quantity, decimal salePrice, long id);
```

**Visual Basic**

```
Public Sub New(quantity As Integer, salePrice As decimal, id As long)
```

**Description**

This is TierPriceData, a member of class TierPriceData.

# TierPriceData Methods

The methods of the TierPriceData class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Clone (see page 1347) | This is Clone, a member of class TierPriceData. |
| | CompareTo (see page 1348) | This is CompareTo, a member of class TierPriceData. |
| | Validate (see page 1348) | This is Validate, a member of class TierPriceData. |

## TierPriceData.Clone Method

**C#**

```
public TierPriceData Clone();
```

**Visual Basic**

```
Public Function Clone() As TierPriceData
```

**Description**

This is Clone, a member of class TierPriceData.

## TierPriceData.CompareTo Method

**C#**

```
public int CompareTo(TierPriceData other);
```

**Visual Basic**

```
Public Function CompareTo(other As TierPriceData) As Integer
```

**Description**

This is CompareTo, a member of class TierPriceData.

## TierPriceData.Validate Method

**C#**

```
public override ValidationResults Validate();
```

**Visual Basic**

```
Public override Function Validate() As ValidationResults
```

**Description**

This is Validate, a member of class TierPriceData.

# TierPriceData Properties

The properties of the TierPriceData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DataState (see page 1348) | Gets or sets the TierPrices persistence State. Used internally. |
| | Id (see page 1348) | This is Id, a member of class TierPriceData. |
| | Quantity (see page 1349) | This is Quantity, a member of class TierPriceData. |
| | RecurrenceId (see page 1349) | Gets or sets the id of the recurrence pattern for this price if it supports recurring billing. |
| | SalePrice (see page 1349) | This is SalePrice, a member of class TierPriceData. |

## TierPriceData.DataState Property

Gets or sets the TierPrices persistence State. Used internally.

**C#**

```
public EkEnumeration.DataState DataState;
```

**Visual Basic**

```
Public Property DataState() As EkEnumeration.DataState
```

## TierPriceData.Id Property

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

**Description**

This is Id, a member of class TierPriceData.

## TierPriceData.Quantity Property

**C#**

```
[RangeValidator(1, RangeBoundaryType.Inclusive, 0, RangeBoundaryType.Ignore,
MessageTemplateResourceName="TierPricing_QuantityNegative",
MessageTemplateResourceType=typeof(Errors))]
public int Quantity;
```

**Visual Basic**

```
<RangeValidator(1, RangeBoundaryType.Inclusive, 0, RangeBoundaryType.Ignore,
MessageTemplateResourceName="TierPricing_QuantityNegative",
MessageTemplateResourceType=typeof(Errors))>
Public Property Quantity() As Integer
```

**Description**

This is Quantity, a member of class TierPriceData.

## TierPriceData.RecurrenceId Property

Gets or sets the id of the recurrence pattern for this price if it supports recurring billing.

**C#**

```
public long RecurrenceId;
```

**Visual Basic**

```
Public Property RecurrenceId() As long
```

## TierPriceData.SalePrice Property

**C#**

```
public decimal SalePrice;
```

**Visual Basic**

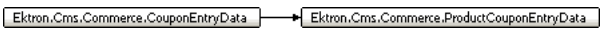
```
Public Property SalePrice() As decimal
```

**Description**

This is SalePrice, a member of class TierPriceData.

# ProductCouponEntryData Class

**Class Hierarchy**




**C#**

```
public class ProductCouponEntryData : CouponEntryData;
```

**Visual Basic**

```
Public Class ProductCouponEntryData
Inherits CouponEntryData
```

**File**

CouponEntryData.cs

**Description**

This is class Ektron.Cms.Commerce.ProductCouponEntryData.

# ProductCouponEntryData Constructor

This is the overview for the ProductCouponEntryData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | ProductCouponEntryData.ProductCouponEntryData () (see page 1350) | This is ProductCouponEntryData, a member of class ProductCouponEntryData. |
| | ProductCouponEntryData.ProductCouponEntryData (long, EkEnumeration.CMSObjectTypes, long, string, bool, EkEnumeration.CatalogEntryType) (see page 1350) | This is ProductCouponEntryData, a member of class ProductCouponEntryData. |

## ProductCouponEntryData.ProductCouponEntryData Constructor ()

**C#**

```
public ProductCouponEntryData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is ProductCouponEntryData, a member of class ProductCouponEntryData.

## ProductCouponEntryData.ProductCouponEntryData Constructor (long, EkEnumeration.CMSObjectTypes, long, string, bool, EkEnumeration.CatalogEntryType)

**C#**

```
public ProductCouponEntryData(long objectId, EkEnumeration.CMSObjectTypes objectType, long
couponId, string title, bool isIncluded, EkEnumeration.CatalogEntryType entryType);
```

**Visual Basic**

```
Public Sub New(objectId As long, objectType As EkEnumeration.CMSObjectTypes, couponId As long,
title As string, isIncluded As bool, entryType As EkEnumeration.CatalogEntryType)
```

**Description**

This is ProductCouponEntryData, a member of class ProductCouponEntryData.

# ProductCouponEntryData Properties

The properties of the ProductCouponEntryData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | EntryType (see page 1351) | Gets or sets the type of catalog entry. |

## ProductCouponEntryData.EntryType Property

Gets or sets the type of catalog entry.

**C#**

```
public Ektron.Cms.Common.EkEnumeration.CatalogEntryType EntryType;
```

**Visual Basic**

```
Public Property EntryType() As Ektron.Cms.Common.EkEnumeration.CatalogEntryType
```

# WarehouseData Class

**Class Hierarchy**

**C#**

```
public class WarehouseData;
```

**Visual Basic**

```
Public Class WarehouseData
```

**File**

WarehouseData.cs

**Description**

This is class Ektron.Cms.Commerce.WarehouseData.

# WarehouseData Constructor

This is the overview for the WarehouseData constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | WarehouseData.WarehouseData () (see page 1351) | This is WarehouseData, a member of class WarehouseData. |
| | WarehouseData.WarehouseData (string, AddressData) (see page 1352) | This is WarehouseData, a member of class WarehouseData. |

## WarehouseData.WarehouseData Constructor ()

**C#**

```
public WarehouseData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is WarehouseData, a member of class WarehouseData.

## WarehouseData.WarehouseData Constructor (string, AddressData)

**C#**

```
public WarehouseData(string warehouseName, AddressData address);
```

**Visual Basic**

```
Public Sub New(warehouseName As string, address As AddressData)
```

**Description**

This is WarehouseData, a member of class WarehouseData.

# WarehouseData Properties

The properties of the WarehouseData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Address (see page 1352) | Gets or sets the address of this Warehouse. |
| | Id (see page 1352) | Gets or sets Id of Address (see page 1352). |
| | IsDefaultWarehouse (see page 1352) | Gets or sets the IsDefaultWarehouse status property for the warehouse. If true, this warehouse is the default warehouse for the system. |
| | Name (see page 1353) | Gets or sets the name of this Warehouse. |

## WarehouseData.Address Property

Gets or sets the address of this Warehouse.

**C#**

```
public AddressData Address;
```

**Visual Basic**

```
Public Property Address() As AddressData
```

## WarehouseData.Id Property

Gets or sets Id of Address (see page 1352).

**C#**

```
public long Id;
```

**Visual Basic**

```
Public Property Id() As long
```

## WarehouseData.IsDefaultWarehouse Property

Gets or sets the IsDefaultWarehouse status property for the warehouse. If true, this warehouse is the default warehouse for the system.

**C#**

```
public bool IsDefaultWarehouse;
```

**Visual Basic**

```
Public Property IsDefaultWarehouse() As bool
```

## WarehouseData.Name Property

Gets or sets the name of this Warehouse.

**C#**

```
public string Name;
```

**Visual Basic**

```
Public Property Name() As string
```

# Weight Class

Class used to define mesurement information.

**Class Hierarchy**

**C#**

```
public class Weight : IComparable<Weight>;
```

**Visual Basic**

```
Public Class Weight
Inherits IComparable
```

**File**

Measure.cs

# Weight Constructor

This is the overview for the Weight constructor overload.

**Overload List**

| | Name | Description |
|---|---|---|
| | Weight.Weight () (see page 1353) | This is Weight, a member of class Weight. |
| | Weight.Weight (WeightUnit, float) (see page 1354) | |

## Weight.Weight Constructor ()

**C#**

```
public Weight();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is Weight, a member of class Weight.

## Weight.Weight Constructor (WeightUnit, float)

**C#**

```
public Weight(WeightUnit units, float totalWeight);
```

**Visual Basic**

```
Public Sub New(units As WeightUnit, totalWeight As float)
```

**Parameters**

| Parameters | Description |
|---|---|
| units | Units (see page 1355) in which the package is weighed. |
| totalWeight | Total weight of the package in Units (see page 1355). |

# Weight Methods

The methods of the Weight class are listed here.

**Public Methods**

| | Name | Description |
|---|---|---|
| | Clone (see page 1354) | This is Clone, a member of class Weight. |
| | CompareTo (see page 1354) | This is CompareTo, a member of class Weight. |
| | ConvertUnits (see page 1354) | Converts the current Weight (see page 1353) object to a new object in the supplied units. |

## Weight.Clone Method

**C#**

```
public Weight Clone();
```

**Visual Basic**

```
Public Function Clone() As Weight
```

**Description**

This is Clone, a member of class Weight.

## Weight.CompareTo Method

**C#**

```
public int CompareTo(Weight other);
```

**Visual Basic**

```
Public Function CompareTo(other As Weight) As Integer
```

**Description**

This is CompareTo, a member of class Weight.

## Weight.ConvertUnits Method

Converts the current Weight (see page 1353) object to a new object in the supplied units.

**C#**

```
public Weight ConvertUnits(WeightUnit toUnit);
```

**Visual Basic**

```
Public Function ConvertUnits(toUnit As WeightUnit) As Weight
```

**Parameters**

| Parameters | Description |
|---|---|
| toUnit | WeightUnit (see page 1364) to convert to |

# Weight Properties

The properties of the Weight class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Amount (see page 1355) | Gets or sets the actual weight value in Units (see page 1355). |
| | Units (see page 1355) | Gets or sets the Units in which the package is weighed. |

## Weight.Amount Property

Gets or sets the actual weight value in Units (see page 1355).

**C#**

```
public float Amount;
```

**Visual Basic**

```
Public Property Amount() As float
```

## Weight.Units Property

Gets or sets the Units in which the package is weighed.

**C#**

```
public WeightUnit Units;
```

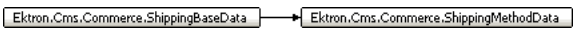
**Visual Basic**

```
Public Property Units() As WeightUnit
```

# ShippingMethodData Class

Data class used to represent shipping methods defined in the cms.

**Class Hierarchy**




**C#**

```
public class ShippingMethodData : ShippingBaseData;
```

**Visual Basic**

```
Public Class ShippingMethodData
Inherits ShippingBaseData
```

**File**

ShippingData.cs

## ShippingMethodData.ShippingMethodData Constructor

**C#**

```
public ShippingMethodData();
```

**Visual Basic**

```
Public Sub New()
```

**Description**

This is ShippingMethodData, a member of class ShippingMethodData.

## ShippingMethodData Properties

The properties of the ShippingMethodData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DisplayOrder (see page 1356) | Gets or sets the display order for this shipping method. |
| | IsActive (see page 1356) | Gets or sets whethe ror not the Shipping Method is Active. |

### ShippingMethodData.DisplayOrder Property

Gets or sets the display order for this shipping method.

**C#**

```
public int DisplayOrder;
```

**Visual Basic**

```
Public Property DisplayOrder() As Integer
```

### ShippingMethodData.IsActive Property

Gets or sets whethe ror not the Shipping Method is Active.

**C#**

```
public bool IsActive;
```

**Visual Basic**

```
Public Property IsActive() As bool
```

# Structs, Records, Enums

The following table lists structs, records, enums in this documentation.

**Enumerations**

| | Name | Description |
|---|---|---|
| | AddressProperty (see page 1357) | Enumeration of Address properties used for sorting and filtering AddressData (see page 1206) objects. |
| | CountryProperty (see page 1357) | Enumeration of CountryData (see page 1239) properties used for sorting and filtering CountryData (see page 1239) objects. |
| | CouponProperty (see page 1358) | Enumeration of Coupon properties used for sorting and filtering CouponData (see page 1243) objects. |

| | | |
|---|---|---|
| | CurrencyProperty (see page 1358) | Enumeration of CurrencyData (see page 1250) properties used for sorting and filtering CurrencyData (see page 1250) objects. |
| | CustomerProperty (see page 1358) | Enumeration of Customer properties used for sorting and filtering CustomerData (see page 1255) objects. |
| | EntryAttributeProperty (see page 1359) | Enumeration of EntryAttributeProperty properties available for sorting and filtering. |
| | EntryGroupItemProperty (see page 1359) | Enumeration of Entry Group Item properties used for sorting and filtering EntryGroupItemData (see page 1279) objects. |
| | EntryGroupProperty (see page 1359) | Enumeration of EntryGroup properties used for sorting and filtering EntryGroupData (see page 1276) objects. |
| | EntryProperty (see page 1360) | Enumeration of EntryData (see page 1264) columns available for sorting and filtering. |
| | ExchangeRateProperty (see page 1360) | Enumeration of ExchangeRate columns available for sorting and filtering. |
| | LinearUnit (see page 1360) | Enumeration of units for measuring length. |
| | OrderProperty (see page 1360) | Enumeration of order properties used for sorting and filtering OrderData (see page 1298) objects. |
| | PackageProperty (see page 1361) | Enumeration representing properties of the packagedata class. |
| | ProductTypeAttributeProperty (see page 1361) | Enumeration of ProductTypeAttributeData (see page 1322) properties available for sorting and filtering. |
| | ProductTypeProperty (see page 1361) | Enumeration of ProductType properties used for sorting and filtering CustomerData (see page 1255) objects. |
| | RegionProperty (see page 1362) | Enumeration of Region properties used for sorting and filtering RegionData (see page 1330) objects. |
| | ShippingMethodProperty (see page 1362) | Enumeration representing properties of the ShippingMethodData (see page 1355) class. |
| | TaxClassProperty (see page 1362) | / Enumeration of TaxClass columns available for sorting and filtering. |
| | TaxRateProperty (see page 1363) | Enumeration of possible values to order Lists of TaxRateData (see page 1338) by. |
| | TaxRateType (see page 1363) | This Represents the default Tax rate Types supported by The CMS. Additonal tax Types may be supported by customers. |
| | TaxTypeProperty (see page 1363) | / Enumeration of TaxType columns available for sorting and filtering. |
| | WarehouseProperty (see page 1364) | Enumeration of Warehouse properties used for sorting and filtering WarehouseData (see page 1351) objects. |
| | WeightUnit (see page 1364) | Enumeration of units for measuring weight. |

# Ektron.Cms.Commerce.AddressProperty Enumeration

Enumeration of Address properties used for sorting and filtering AddressData (see page 1206) objects.

**C#**

```
public enum AddressProperty {
}
```

**Visual Basic**

```
Public Enum AddressProperty
End Enum
```

**File**

AddressData.cs

# Ektron.Cms.Commerce.CountryProperty Enumeration

Enumeration of CountryData (see page 1239) properties used for sorting and filtering CountryData (see page 1239) objects.

**C#**

```
public enum CountryProperty {
}
```

**Visual Basic**

```
Public Enum CountryProperty
End Enum
```

**File**

CountryData.cs

# Ektron.Cms.Commerce.CouponProperty Enumeration

Enumeration of Coupon properties used for sorting and filtering CouponData (see page 1243) objects.

**C#**

```
public enum CouponProperty {
}
```

**Visual Basic**

```
Public Enum CouponProperty
End Enum
```

**File**

CouponData.cs

# Ektron.Cms.Commerce.CurrencyProperty Enumeration

Enumeration of CurrencyData (see page 1250) properties used for sorting and filtering CurrencyData (see page 1250) objects.

**C#**

```
public enum CurrencyProperty : int {
}
```

**Visual Basic**

```
Public Enum CurrencyProperty
Inherits int
End Enum
```

**File**

CurrencyData.cs

# Ektron.Cms.Commerce.CustomerProperty Enumeration

Enumeration of Customer properties used for sorting and filtering CustomerData (see page 1255) objects.

**C#**

```
public enum CustomerProperty {
}
```

**Visual Basic**

```
Public Enum CustomerProperty
```

```
End Enum
```

**File**

CustomerData.cs

# Ektron.Cms.Commerce.EntryAttributeProperty Enumeration

Enumeration of EntryAttributeProperty properties available for sorting and filtering.

**C#**

```
public enum EntryAttributeProperty : int {
}
```

**Visual Basic**

```
Public Enum EntryAttributeProperty
Inherits int
End Enum
```

**File**

ProductAttributeData.cs

# Ektron.Cms.Commerce.EntryGroupItemProperty Enumeration

Enumeration of Entry Group Item properties used for sorting and filtering EntryGroupItemData (see page 1279) objects.

**C#**

```
public enum EntryGroupItemProperty : int {
}
```

**Visual Basic**

```
Public Enum EntryGroupItemProperty
Inherits int
End Enum
```

**File**

EntryGroupItemData.cs

# Ektron.Cms.Commerce.EntryGroupProperty Enumeration

Enumeration of EntryGroup properties used for sorting and filtering EntryGroupData (see page 1276) objects.

**C#**

```
public enum EntryGroupProperty : int {
}
```

**Visual Basic**

```
Public Enum EntryGroupProperty
Inherits int
End Enum
```

**File**

EntryGroupData.cs

# Ektron.Cms.Commerce.EntryProperty Enumeration

Enumeration of EntryData (see page 1264) columns available for sorting and filtering.

**C#**

```
public enum EntryProperty {
}
```

**Visual Basic**

```
Public Enum EntryProperty
End Enum
```

**File**

EntryData.cs

# Ektron.Cms.Commerce.ExchangeRateProperty Enumeration

Enumeration of ExchangeRate columns available for sorting and filtering.

**C#**

```
public enum ExchangeRateProperty {
}
```

**Visual Basic**

```
Public Enum ExchangeRateProperty
End Enum
```

**File**

ExchangeRateData.cs

# Ektron.Cms.Commerce.LinearUnit Enumeration

Enumeration of units for measuring length.

**C#**

```
public enum LinearUnit {
}
```

**Visual Basic**

```
Public Enum LinearUnit
End Enum
```

**File**

Measure.cs

# Ektron.Cms.Commerce.OrderProperty Enumeration

Enumeration of order properties used for sorting and filtering OrderData (see page 1298) objects.

**C#**

```
public enum OrderProperty : int {
```

```
}
```

**Visual Basic**

```
Public Enum OrderProperty
Inherits int
End Enum
```

**File**

OrderData.cs

# Ektron.Cms.Commerce.PackageProperty Enumeration

Enumeration representing properties of the packagedata class.

**C#**

```
public enum PackageProperty {
}
```

**Visual Basic**

```
Public Enum PackageProperty
End Enum
```

**File**

PackageData.cs

# Ektron.Cms.Commerce.ProductTypeAttributeProperty Enumeration

Enumeration of ProductTypeAttributeData (see page 1322) properties available for sorting and filtering.

**C#**

```
public enum ProductTypeAttributeProperty : int {
}
```

**Visual Basic**

```
Public Enum ProductTypeAttributeProperty
Inherits int
End Enum
```

**File**

ProductAttributeData.cs

# Ektron.Cms.Commerce.ProductTypeProperty Enumeration

Enumeration of ProductType properties used for sorting and filtering CustomerData (see page 1255) objects.

**C#**

```
public enum ProductTypeProperty : int {
}
```

**Visual Basic**

```
Public Enum ProductTypeProperty
Inherits int
End Enum
```

**File**

ProductTypeData.cs

# Ektron.Cms.Commerce.RegionProperty Enumeration

Enumeration of Region properties used for sorting and filtering RegionData (see page 1330) objects.

**C#**

```
public enum RegionProperty : int {
}
```

**Visual Basic**

```
Public Enum RegionProperty
Inherits int
End Enum
```

**File**

RegionData.cs

# Ektron.Cms.Commerce.ShippingMethodProperty Enumeration

Enumeration representing properties of the ShippingMethodData (see page 1355) class.

**C#**

```
public enum ShippingMethodProperty {
}
```

**Visual Basic**

```
Public Enum ShippingMethodProperty
End Enum
```

**File**

ShippingData.cs

# Ektron.Cms.Commerce.TaxClassProperty Enumeration

/ Enumeration of TaxClass columns available for sorting and filtering.

**C#**

```
public enum TaxClassProperty {
}
```

**Visual Basic**

```
Public Enum TaxClassProperty
End Enum
```

**File**

TaxClassData.cs

# Ektron.Cms.Commerce.TaxRateProperty Enumeration

Enumeration of possible values to order Lists of TaxRateData (see page 1338) by.

**C#**

```
public enum TaxRateProperty {
}
```

**Visual Basic**

```
Public Enum TaxRateProperty
End Enum
```

**File**

TaxRateData.cs

# Ektron.Cms.Commerce.TaxRateType Enumeration

This Represents the default Tax rate Types supported by The CMS. Additonal tax Types may be supported by customers.

**C#**

```
public enum TaxRateType {
}
```

**Visual Basic**

```
Public Enum TaxRateType
End Enum
```

**File**

TaxRateData.cs

**Remarks**

TaxRateType Id's 1-100 should be reserved for internal use.

**Example**

customer could create specialized VAT TaxRate type.

# Ektron.Cms.Commerce.TaxTypeProperty Enumeration

/ Enumeration of TaxType columns available for sorting and filtering.

**C#**

```
public enum TaxTypeProperty {
}
```

**Visual Basic**

```
Public Enum TaxTypeProperty
End Enum
```

**File**

TaxTypeData.cs

## Ektron.Cms.Commerce.WarehouseProperty Enumeration

Enumeration of Warehouse properties used for sorting and filtering WarehouseData (see page 1351) objects.

**C#**

```
public enum WarehouseProperty : int {
}
```

**Visual Basic**

```
Public Enum WarehouseProperty
Inherits int
End Enum
```

**File**

WarehouseData.cs

## Ektron.Cms.Commerce.WeightUnit Enumeration

Enumeration of units for measuring weight.

**C#**

```
public enum WeightUnit {
}
```

**Visual Basic**

```
Public Enum WeightUnit
End Enum
```

**File**

Measure.cs

# Ektron.Cms.Common Namespace

This is namespace Ektron.Cms.Common.

**Classes**

| | Name | Description |
|---|---|---|
| | UrlAliasAutoData (see page 1365) | A data class used to get or set information about an Automatic URL Alias. |
| | UrlAliasManualData (see page 1370) | A data class used to get or set information about a Manual URL Alias. |
| | UrlAliasRegExData (see page 1374) | A data class used to get or set information about a URL Alias that uses RegEx. |

# Classes

The following table lists classes in this documentation.

### Classes

| | Name | Description |
|---|---|---|
| | UrlAliasAutoData (see page 1365) | A data class used to get or set information about an Automatic URL Alias. |
| | UrlAliasManualData (see page 1370) | A data class used to get or set information about a Manual URL Alias. |
| | UrlAliasRegExData (see page 1374) | A data class used to get or set information about a URL Alias that uses RegEx. |

# UrlAliasAutoData Class

A data class used to get or set information about an Automatic URL Alias.

### Class Hierarchy

Ektron.Cms.Common.UrlAliasAutoData

### C#

```
public class UrlAliasAutoData;
```

### Visual Basic

```
Public Class UrlAliasAutoData
```

### File

UrlAliasAutoData.vb

# UrlAliasAutoData Properties

The properties of the UrlAliasAutoData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AliasName (see page 1366) | Gets or sets the alias name for the URL. |
| | AutoAliasType (see page 1366) | Gets or sets the automatic alias source type. Types are Taxonomy (see page 842) and Folder (see page 652). |
| | AutoId (see page 1366) | Gets the unique ID generated for this record. |
| | Example (see page 1366) | Gets or sets a sample of the selected configuration. |
| | ExcludedPath (see page 1367) | Returns the folder or taxonomy path excluded from the selected path. |
| | FileExtension (see page 1367) | Gets or sets the selected file extension. |
| R | FormattedSelectedPath (see page 1367) | Returns the folder or taxonomy path with modifications |
| | HostName (see page 1367) | Gets the name of the host where the content resides. |
| | IsEnabled (see page 1367) | Gets or sets whether the configuration is active or not. |
| | LanguageId (see page 1368) | Gets the language ID for the automatic alias. |
| | PageNameType (see page 1368) | Gets or sets the name for the configuration. Available types are:<br>• ContentTitle<br>• ContentId<br>• ContentIdAndLanguage |
| | ReplacementCharacter (see page 1368) | Gets or sets the character that will replace special characters in the generated URL. |
| | SelectedPath (see page 1368) | Returns the folder or taxonomy path selected by the user |

| | | |
|---|---|---|
| | SiteId (see page 1369) | Gets or sets the root folder ID of the site. If no site is specified, returns 0 (zero). |
| | SourceId (see page 1369) | Gets or sets either the taxonomy ID or the folder ID depending on the auto alias type selected. |
| | SourceName (see page 1369) | Gets the name of the taxonomy or folder selected depending on the auto alias type selected. |
| | Target (see page 1369) | Gets or sets the destination for the alias. |

## UrlAliasAutoData.AliasName Property

Gets or sets the alias name for the URL.

**C#**

```
public String AliasName;
```

**Visual Basic**

```
Public Property AliasName() As String
```

**Returns**

The alias name for the URL.

## UrlAliasAutoData.AutoAliasType Property

Gets or sets the automatic alias source type. Types are Taxonomy (see page 842) and Folder (see page 652).

**C#**

```
public EkEnumeration.AutoAliasType AutoAliasType;
```

**Visual Basic**

```
Public Property AutoAliasType() As EkEnumeration.AutoAliasType
```

**Returns**

The automatic alias source type.

## UrlAliasAutoData.AutoId Property

Gets the unique ID generated for this record.

**C#**

```
public Long AutoId;
```

**Visual Basic**

```
Public Property AutoId() As Long
```

**Returns**

The unique ID generated for this record.

## UrlAliasAutoData.Example Property

Gets or sets a sample of the selected configuration.

**C#**

```
public String Example;
```

**Visual Basic**

```
Public Property Example() As String
```

**Returns**

A sample of the selected configuration.

## UrlAliasAutoData.ExcludedPath Property

Returns the folder or taxonomy path excluded from the selected path.

**C#**

```
public String ExcludedPath;
```

**Visual Basic**

```
Public Property ExcludedPath() As String
```

## UrlAliasAutoData.FileExtension Property

Gets or sets the selected file extension.

**C#**

```
public String FileExtension;
```

**Visual Basic**

```
Public Property FileExtension() As String
```

**Returns**

The selected file extension.

## UrlAliasAutoData.FormattedSelectedPath Property

Returns the folder or taxonomy path with modifications

**C#**

```
public ReadOnly String FormattedSelectedPath;
```

**Visual Basic**

```
Public ReadOnly Property FormattedSelectedPath() As String
```

## UrlAliasAutoData.HostName Property

Gets the name of the host where the content resides.

**C#**

```
public String HostName;
```

**Visual Basic**

```
Public Property HostName() As String
```

**Returns**

The name of the host where the content resides.

## UrlAliasAutoData.IsEnabled Property

Gets or sets whether the configuration is active or not.

**C#**

```
public Boolean IsEnabled;
```

**Visual Basic**

```
Public Property IsEnabled() As Boolean
```

**Returns**

True if the configuration is active or not. Otherwise, it returns false.

## UrlAliasAutoData.LanguageId Property

Gets the language ID for the automatic alias.

**C#**

```
public int LanguageId;
```

**Visual Basic**

```
Public Property LanguageId() As Integer
```

**Returns**

The language ID for the automatic alias.

## UrlAliasAutoData.PageNameType Property

Gets or sets the name for the configuration. Available types are:

- ContentTitle
- ContentId
- ContentIdAndLanguage

**C#**

```
public EkEnumeration.AutoAliasNameType PageNameType;
```

**Visual Basic**

```
Public Property PageNameType() As EkEnumeration.AutoAliasNameType
```

**Returns**

The name for the configuration.

## UrlAliasAutoData.ReplacementCharacter Property

Gets or sets the character that will replace special characters in the generated URL.

**C#**

```
public String ReplacementCharacter;
```

**Visual Basic**

```
Public Property ReplacementCharacter() As String
```

**Returns**

The character that will replace special characters in the generated URL.

## UrlAliasAutoData.SelectedPath Property

Returns the folder or taxonomy path selected by the user

**C#**

```
public String SelectedPath;
```

**Visual Basic**

```
Public Property SelectedPath() As String
```

## UrlAliasAutoData.SiteId Property

Gets or sets the root folder ID of the site. If no site is specified, returns 0 (zero).

**C#**

```
public Long SiteId;
```

**Visual Basic**

```
Public Property SiteId() As Long
```

**Returns**

The root folder ID of the site. If no site is specified, returns 0 (zero).

## UrlAliasAutoData.SourceId Property

Gets or sets either the taxonomy ID or the folder ID depending on the auto alias type selected.

**C#**

```
public Long SourceId;
```

**Visual Basic**

```
Public Property SourceId() As Long
```

**Returns**

Either the taxonomy ID or the folder ID depending on the auto alias type selected.

## UrlAliasAutoData.SourceName Property

Gets the name of the taxonomy or folder selected depending on the auto alias type selected.

**C#**

```
public String SourceName;
```

**Visual Basic**

```
Public Property SourceName() As String
```

**Returns**

The name of the taxonomy or folder selected depending on the auto alias type selected.

## UrlAliasAutoData.Target Property

Gets or sets the destination for the alias.

**C#**

```
public String Target;
```

**Visual Basic**

```
Public Property Target() As String
```

**Returns**

The destination for the alias.

# UrlAliasManualData Class

A data class used to get or set information about a Manual URL Alias.

### Class Hierarchy

### C#

```
public class UrlAliasManualData;
```

### Visual Basic

```
Public Class UrlAliasManualData
```

### File

UrlAliasManualData.vb

# UrlAliasManualData Properties

The properties of the UrlAliasManualData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AliasId (see page 1370) | Gets the unique ID generated for this record. |
| | AliasName (see page 1371) | Gets or sets the alias name for the URL without the extension. |
| | AliasPageName (see page 1371) | Gets the name of the alias without the query string. |
| | ContentId (see page 1371) | Gets or sets the Content (see page 788) ID. |
| | ContentLanguage (see page 1371) | Gets the content language. |
| | ContentTitle (see page 1372) | Gets the title of the content. |
| | DisplayAlias (see page 1372) | Gets the alias URL with the extension and query parameters. |
| | FileExtension (see page 1372) | Gets or sets the file extension for the alias. |
| | HostName (see page 1372) | Gets the name of the host where the content resides. |
| | IsDefault (see page 1373) | Gets or sets which alias name should show when the content has multiple aliases. |
| | IsEnabled (see page 1373) | Gets or sets whether the alias name is active or not. |
| | LibraryId (see page 1373) | Gets or sets the Library (see page 664) ID for the content. |
| | QueryString (see page 1373) | Gets or sets the querystring parameters to append to the aliased URL. |
| | QueryStringAction (see page 1373) | Gets or sets the querystringaction parameters. |
| | SiteId (see page 1374) | Gets the root folder ID of the site. If no site is specified, returns 0 (zero). |
| | Target (see page 1374) | Gets or sets the destination URL for the alias. |

## UrlAliasManualData.AliasId Property

Gets the unique ID generated for this record.

### C#

```
public Long AliasId;
```

### Visual Basic

```
Public Property AliasId() As Long
```

**Returns**

The unique ID generated for this record.

## UrlAliasManualData.AliasName Property

Gets or sets the alias name for the URL without the extension.

**C#**

```
public String AliasName;
```

**Visual Basic**

```
Public Property AliasName() As String
```

**Returns**

The alias name for the URL without the extension.

## UrlAliasManualData.AliasPageName Property

Gets the name of the alias without the query string.

**C#**

```
public String AliasPageName;
```

**Visual Basic**

```
Public Property AliasPageName() As String
```

**Returns**

The name of the alias without the query string.

## UrlAliasManualData.ContentId Property

Gets or sets the Content (see page 788) ID.

**C#**

```
public Long ContentId;
```

**Visual Basic**

```
Public Property ContentId() As Long
```

**Returns**

The content ID.

## UrlAliasManualData.ContentLanguage Property

Gets the content language.

**C#**

```
public int ContentLanguage;
```

**Visual Basic**

```
Public Property ContentLanguage() As Integer
```

**Returns**

The content language.

## UrlAliasManualData.ContentTitle Property

Gets the title of the content.

**C#**

```
public String ContentTitle;
```

**Visual Basic**

```
Public Property ContentTitle() As String
```

**Returns**

The title of the content.

## UrlAliasManualData.DisplayAlias Property

Gets the alias URL with the extension and query parameters.

**C#**

```
public String DisplayAlias;
```

**Visual Basic**

```
Public Property DisplayAlias() As String
```

**Returns**

The alias URL with the extension and query parameters.

**Remarks**

This is a computed field.

## UrlAliasManualData.FileExtension Property

Gets or sets the file extension for the alias.

**C#**

```
public String FileExtension;
```

**Visual Basic**

```
Public Property FileExtension() As String
```

**Returns**

The file extension for the alias.

## UrlAliasManualData.HostName Property

Gets the name of the host where the content resides.

**C#**

```
public String HostName;
```

**Visual Basic**

```
Public Property HostName() As String
```

**Returns**

The name of the host where the content resides.

## UrlAliasManualData.IsDefault Property

Gets or sets which alias name should show when the content has multiple aliases.

**C#**

```
public Boolean IsDefault;
```

**Visual Basic**

```
Public Property IsDefault() As Boolean
```

**Returns**

Which alias name should show when the content has multiple aliases.

## UrlAliasManualData.IsEnabled Property

Gets or sets whether the alias name is active or not.

**C#**

```
public Boolean IsEnabled;
```

**Visual Basic**

```
Public Property IsEnabled() As Boolean
```

**Returns**

Returns true if the alias name is active. Otherwise, it returns false.

## UrlAliasManualData.LibraryId Property

Gets or sets the Library (see page 664) ID for the content.

**C#**

```
public Long LibraryId;
```

**Visual Basic**

```
Public Property LibraryId() As Long
```

**Returns**

The Library (see page 664) ID for the content.

## UrlAliasManualData.QueryString Property

Gets or sets the querystring parameters to append to the aliased URL.

**C#**

```
public String QueryString;
```

**Visual Basic**

```
Public Property QueryString() As String
```

**Returns**

The querystring parameters to append to the aliased URL.

## UrlAliasManualData.QueryStringAction Property

Gets or sets the querystringaction parameters.

**C#**

```
public EkEnumeration.QueryStringActionType QueryStringAction;
```

**Visual Basic**

```
Public Property QueryStringAction() As EkEnumeration.QueryStringActionType
```

**Returns**

The querystring parameters.

## UrlAliasManualData.SiteId Property

Gets the root folder ID of the site. If no site is specified, returns 0 (zero).

**C#**

```
public Long SiteId;
```

**Visual Basic**

```
Public Property SiteId() As Long
```

**Returns**

The root folder ID of the site. If no site is specified, returns 0 (zero).

## UrlAliasManualData.Target Property

Gets or sets the destination URL for the alias.

**C#**

```
public String Target;
```

**Visual Basic**

```
Public Property Target() As String
```

**Returns**

The destination URL for the alias.

# UrlAliasRegExData Class

A data class used to get or set information about a URL Alias that uses RegEx.

**Class Hierarchy**

**C#**

```
public class UrlAliasRegExData;
```

**Visual Basic**

```
Public Class UrlAliasRegExData
```

**File**

UrlAliasRegExData.vb

# UrlAliasRegExData Properties

The properties of the UrlAliasRegExData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Expression (see page 1375) | Gets or sets the regular expression to be used. |
| | ExpressionMap (see page 1375) | Gets or sets the destination for the expression. |
| | ExpressionName (see page 1375) | Gets or sets a descriptive name for the expression. |
| | IsEnabled (see page 1376) | Gets or sets whether the expression is active or not. |
| | Priority (see page 1376) | Gets or sets the look up order for the expression list. Priorities are:<br>• None<br>• High<br>• Medium<br>• Low |
| | RegExId (see page 1376) | Gets the unique ID generated for each record. |
| | SiteId (see page 1377) | Gets or sets the root folder ID of the site. Returns zero if no site is specified. |
| | TransformedUrl (see page 1377) | Gets or sets a sample of how the ExpressionMap (see page 1375) URL will look. |

## UrlAliasRegExData.Expression Property

Gets or sets the regular expression to be used.

**C#**

```
public String Expression;
```

**Visual Basic**

```
Public Property Expression() As String
```

**Returns**

The regular expression to be used.

## UrlAliasRegExData.ExpressionMap Property

Gets or sets the destination for the expression.

**C#**

```
public String ExpressionMap;
```

**Visual Basic**

```
Public Property ExpressionMap() As String
```

**Returns**

The destination for the expression.

## UrlAliasRegExData.ExpressionName Property

Gets or sets a descriptive name for the expression.

**C#**

```
public String ExpressionName;
```

**Visual Basic**

```
Public Property ExpressionName() As String
```

**Returns**

A descriptive name for the expression.

## UrlAliasRegExData.IsEnabled Property

Gets or sets whether the expression is active or not.

**C#**

```
public Boolean IsEnabled;
```

**Visual Basic**

```
Public Property IsEnabled() As Boolean
```

**Returns**

Returns true if the expression is active. Otherwise, it returns false.

## UrlAliasRegExData.Priority Property

Gets or sets the look up order for the expression list. Priorities are:

- None
- High
- Medium
- Low

**C#**

```
public EkEnumeration.RegExPriority Priority;
```

**Visual Basic**

```
Public Property Priority() As EkEnumeration.RegExPriority
```

**Returns**

The look up order for the expression list.

## UrlAliasRegExData.RegExId Property

Gets the unique ID generated for each record.

**C#**

```
public Long RegExId;
```

**Visual Basic**

```
Public Property RegExId() As Long
```

**Returns**

The unique ID generated for each record.

### UrlAliasRegExData.SiteId Property

Gets or sets the root folder ID of the site. Returns zero if no site is specified.

**C#**

```
public Long SiteId;
```

**Visual Basic**

```
Public Property SiteId() As Long
```

**Returns**

The root folder ID of the site. Returns zero if no site is specified.

### UrlAliasRegExData.TransformedUrl Property

Gets or sets a sample of how the ExpressionMap (see page 1375) URL will look.

**C#**

```
public String TransformedUrl;
```

**Visual Basic**

```
Public Property TransformedUrl() As String
```

**Returns**

A sample of how the ExpressionMap (see page 1375) URL will look.

# CommunityUserData Class

UserData (see page 1419) object plus Community (see page 718) related properties.

**Class Hierarchy**

**C#**

```
public class CommunityUserData : UserData;
```

**Visual Basic**

```
Public Class CommunityUserData
Inherits UserData
```

**File**

UserData.vb

# CommunityUserData Properties

The properties of the CommunityUserData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Tags (see page 1378) | Gets a Comma delimitted list of tags applied to the user. Editing this list does not affect tags on user. Tags must be added to User (see page 913) though the Tag API (see page 639). |

## CommunityUserData.Tags Property

Gets a Comma delimitted list of tags applied to the user. Editing this list does not affect tags on user. Tags must be added to User (see page 913) though the Tag API (see page 639).

**C#**

```
public String Tags;
```

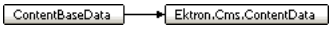
**Visual Basic**

```
Public Property Tags() As String
```

# ContentData Class

A data object that contains the properties that make up a content item.

**Class Hierarchy**




**C#**

```
public class ContentData : ContentBaseData;
```

**Visual Basic**

```
Public Class ContentData
Inherits ContentBaseData
```

**File**

ContentData.vb

# ContentData Properties

The properties of the ContentData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | ApprovalMethod (see page 1380) | Gets or sets the approval method of a content item for the ContentData (see page 1378) object. Choices are:<br>• **1** = Force All Approvers - content checked-out during the approval process will force the approval chain to reset to the beginning.<br>• **0** = Do Not Force All Approvers - content checked-out during the approval process will continue through the approval chain. |
| | Approver (see page 1381) | Gets or sets the approvers of a content item for the ContentData (see page 1378) object. |
| | Comment (see page 1381) | Gets or sets comment information associated with a content item for the ContentData (see page 1378) object. |

| | | |
|---|---|---|
| | ContType (see page 1381) | Gets or sets the content type as an integer for the ContentData (see page 1378) object. Choices are:<br>• **-1** = AllTypes<br>• **1** = Content (see page 788)<br>• **2** = Forms<br>• **3** = Archive_Content<br>• **4** = Archive_Forms<br>• **7** = LibraryItem<br>• **8** = Assets<br>• **9** = Archive_Assets<br>• **12** = Archive_Media<br>• **99** = NonLibraryContent<br>• **104** = Multimedia<br>• **106** = ImageContent<br>• **1111** = DiscussionTopic<br>• **3333** = CatalogEntry |
| | DateCreated (see page 1382) | Gets or sets the creation date of a content item for the ContentData (see page 1378) object. |
| | DisplayDateCreated (see page 1382) | Gets or sets the creation date of a content item formatted for display for the ContentData (see page 1378) object. For example, `June 11, 2007`. |
| | DisplayEndDate (see page 1382) | Gets or sets the end date of a content item formatted for display. This is for the ContentData (see page 1378) object. For example, `June 11, 2007`. |
| | DisplayGoLive (see page 1383) | Gets or sets the start date a content item goes live on the site formatted for display. This is for the ContentData (see page 1378) object. For example, `June 11, 2007`. |
| | DisplayLastEditDate (see page 1383) | Gets or sets the content's last edit date formatted for display for the ContentData (see page 1378) object. For example, `June 11, 2007`. |
| | EditorFirstName (see page 1383) | Gets or sets the content editor's first name for the ContentData (see page 1378) object. |
| | EditorLastName (see page 1383) | Gets or sets the content editor's last name for the ContentData (see page 1378) object. |
| | EditorUserNames (see page 1384) | Gets or sets the username of the last editor of the content item for the ContentData (see page 1378) object. |
| | EndDate (see page 1384) | Gets or sets the end date of a content item. This is for the ContentData (see page 1378) object. |
| | EndDateAction (see page 1384) | Gets or sets the action to perform on a content item once the End Date has been reached. This is for the ContentData (see page 1378) object. Choices are:<br>• **1** = Archive and remove from site (expire)<br>• **2** = Archive and remain on site<br>• **3** = Add to the CMS Refresh Report |
| | FolderId (see page 1385) | Gets or Sets the content item's folder ID for the ContentData (see page 1378) object. |
| | FolderName (see page 1385) | Gets or sets the content item's folder name for the ContentData (see page 1378) object. |
| | GoLive (see page 1385) | Gets or sets the start date a content item goes live on the site. This is for the ContentData (see page 1378) object. |
| | HistoryId (see page 1385) | Gets or sets the content item's history ID for the ContentData (see page 1378) object. |
| | HyperLink (see page 1386) | Gets or sets the content item's Hyperlink in the CMS400.NET for the ContentData (see page 1378) object. |
| | InheritedFrom (see page 1386) | Gets or sets the folder ID from which to inherit permissions for the ContentData (see page 1378) object. |
| | IsInherited (see page 1386) | Gets or sets whether the content item inherits permissions from the folder settings for the ContentData (see page 1378) object. |
| | IsPublished (see page 1386) | Gets or sets whether the content is in a published state for the ContentData (see page 1378) object. |
| | IsSearchable (see page 1387) | Gets or sets whether the content is searchable for the ContentData (see page 1378) object. True = the content searchable. |

| | | |
|---|---|---|
| | IsXmlInherited (see page 1387) | Gets or sets whether the XML Smart Form (see page 836) configuration is inherited from the folder. True = XML Smart Form (see page 836) configuration is inherited from the folder. This is for the ContentData (see page 1378) object. |
| | LastEditDate (see page 1387) | Gets or sets the content's last edit date for the ContentData (see page 1378) object. |
| | ManualAlias (see page 1387) | Gets or sets the manual alias by its name for the ContentData (see page 1378) object. For example, `/~YourSite~/text.aspx`. |
| | ManualAliasId (see page 1388) | Gets or sets the ID of the of a manual alias for the ContentData (see page 1378) object. |
| | MetaData (see page 1388) | Gets or sets the Metadata of a content item as an array of ContentMetadata. This is for the ContentData (see page 1378) object. |
| | Path (see page 1388) | Gets or sets the CMS400.NET path to content item for the ContentData (see page 1378) object. |
| | Permissions (see page 1389) | Gets or sets permission data for the ContentData (see page 1378) object. |
| | Status (see page 1389) | Gets or sets the letter value representing the status of a content item for the ContentData (see page 1378) object.<br>• **A** = Approved<br>• **O** = Checked Out<br>• **I** = Checked In<br>• **S** = Submitted for Approval<br>• **M** = Marked for Deletion<br>• **P** = Pending Go Live Date<br>• **T** = Awaiting Completion of Associated Tasks<br>• **D** = Pending Deletion |
| | StyleSheet (see page 1389) | Gets or sets the style sheet to use with a content item for the ContentData (see page 1378) object. |
| | TemplateConfiguration (see page 1390) | Gets or sets the content item's template configuration for the ContentData (see page 1378) object. |
| | UserId (see page 1390) | Gets or sets the ID of the user for the ContentData (see page 1378) object. |
| | UserName (see page 1390) | Important: This property is obsolete. To retrieve the user name, call EkUser GetActiveUserbyID with the correct User (see page 913) ID.<br>Gets or sets the content's username for the ContentData (see page 1378) object. |
| | XmlConfiguration (see page 1390) | Gets or sets the content item's XML configuration for the ContentData (see page 1378) object. |
| | XmlInheritedFrom (see page 1391) | Gets or sets the ID of the folder from which the XML Smart Form (see page 836) configuration is inherited. This is for the ContentData (see page 1378) object. |

# ContentData.ApprovalMethod Property

Gets or sets the approval method of a content item for the ContentData (see page 1378) object. Choices are:

- **1** = Force All Approvers - content checked-out during the approval process will force the approval chain to reset to the beginning.
- **0** = Do Not Force All Approvers - content checked-out during the approval process will continue through the approval chain.

**C#**

```
public int ApprovalMethod;
```

**Visual Basic**

```
Public Property ApprovalMethod() As Integer
```

**Returns**

An integer value representing the type of approval method.

**Remarks**

If the content is inheriting the approval method from the folder, the IsInherited (see page 1386) property must be false to change this property.

# ContentData.Approver Property

Gets or sets the approvers of a content item for the ContentData (see page 1378) object.

**C#**

```
public String Approver;
```

**Visual Basic**

```
Public Property Approver() As String
```

**Returns**

A string value representing the approver(s) of a content item.

# ContentData.Comment Property

Gets or sets comment information associated with a content item for the ContentData (see page 1378) object.

**C#**

```
public String Comment;
```

**Visual Basic**

```
Public Property Comment() As String
```

**Returns**

Comment information associated with a content item.

# ContentData.ContType Property

Gets or sets the content type as an integer for the ContentData (see page 1378) object. Choices are:

- **-1** = AllTypes
- **1** = Content (see page 788)
- **2** = Forms
- **3** = Archive_Content
- **4** = Archive_Forms
- **7** = LibraryItem
- **8** = Assets
- **9** = Archive_Assets
- **12** = Archive_Media
- **99** = NonLibraryContent
- **104** = Multimedia
- **106** = ImageContent

- **1111** = DiscussionTopic
- **3333** = CatalogEntry

**C#**

```
public int ContType;
```

**Visual Basic**

```
Public Property ContType() As Integer
```

**Returns**

An integer representing the content type.

# ContentData.DateCreated Property

Gets or sets the creation date of a content item for the ContentData (see page 1378) object.

**C#**

```
public DateTime DateCreated;
```

**Visual Basic**

```
Public Property DateCreated() As DateTime
```

**Returns**

The creation date of a content item as DateTime.

# ContentData.DisplayDateCreated Property

Gets or sets the creation date of a content item formatted for display for the ContentData (see page 1378) object. For example, `June 11, 2007`.

**C#**

```
public String DisplayDateCreated;
```

**Visual Basic**

```
Public Property DisplayDateCreated() As String
```

**Returns**

Creation date of a content item formatted for display.

# ContentData.DisplayEndDate Property

Gets or sets the end date of a content item formatted for display. This is for the ContentData (see page 1378) object. For example, `June 11, 2007`.

**C#**

```
public String DisplayEndDate;
```

**Visual Basic**

```
Public Property DisplayEndDate() As String
```

**Returns**

End date of a content item formatted for display.

# ContentData.DisplayGoLive Property

Gets or sets the start date a content item goes live on the site formatted for display. This is for the ContentData (see page 1378) object. For example, `June 11, 2007`.

**C#**

```
public String DisplayGoLive;
```

**Visual Basic**

```
Public Property DisplayGoLive() As String
```

**Returns**

Start date a content item goes live on the site formatted for display.

# ContentData.DisplayLastEditDate Property

Gets or sets the content's last edit date formatted for display for the ContentData (see page 1378) object. For example, `June 11, 2007`.

**C#**

```
public String DisplayLastEditDate;
```

**Visual Basic**

```
Public Property DisplayLastEditDate() As String
```

**Returns**

Content (see page 788)'s last edit date formatted for display.

# ContentData.EditorFirstName Property

Gets or sets the content editor's first name for the ContentData (see page 1378) object.

**C#**

```
public String EditorFirstName;
```

**Visual Basic**

```
Public Property EditorFirstName() As String
```

**Returns**

The editor's first name as a string.

# ContentData.EditorLastName Property

Gets or sets the content editor's last name for the ContentData (see page 1378) object.

**C#**

```
public String EditorLastName;
```

**Visual Basic**

```
Public Property EditorLastName() As String
```

**Returns**

The editor's last name as a string.

# ContentData.EditorUserNames Property

Gets or sets the username of the last editor of the content item for the ContentData (see page 1378) object.

**C#**

```
[Obsolete("This property is no longer valid.  To retrieve the user name, call EkUser
GetActiveUserbyID with the correct User ID")]
public String EditorUserNames;
```

**Visual Basic**

```
<Obsolete("This property is no longer valid.  To retrieve the user name, call EkUser
GetActiveUserbyID with the correct User ID")>
Public Property EditorUserNames() As String
```

**Returns**

Username of the last editor of the content item.

# ContentData.EndDate Property

Gets or sets the end date of a content item. This is for the ContentData (see page 1378) object.

**C#**

```
public String EndDate;
```

**Visual Basic**

```
Public Property EndDate() As String
```

**Returns**

A string value that represents the end date of a content item.

# ContentData.EndDateAction Property

Gets or sets the action to perform on a content item once the End Date has been reached. This is for the ContentData (see page 1378) object. Choices are:

- **1** = Archive and remove from site (expire)
- **2** = Archive and remain on site
- **3** = Add to the CMS Refresh Report

**C#**

```
public int EndDateAction;
```

**Visual Basic**

```
Public Property EndDateAction() As Integer
```

**Returns**

An integer value representing one of the above end date action choices.

# ContentData.FolderId Property

Gets or Sets the content item's folder ID for the ContentData (see page 1378) object.

**C#**

```
public Long FolderId;
```

**Visual Basic**

```
Public Property FolderId() As Long
```

**Returns**

Folder (see page 652) ID as an Long.

# ContentData.FolderName Property

Gets or sets the content item's folder name for the ContentData (see page 1378) object.

**C#**

```
public String FolderName;
```

**Visual Basic**

```
Public Property FolderName() As String
```

**Returns**

The content item's folder name.

# ContentData.GoLive Property

Gets or sets the start date a content item goes live on the site. This is for the ContentData (see page 1378) object.

**C#**

```
public String GoLive;
```

**Visual Basic**

```
Public Property GoLive() As String
```

**Returns**

A string value that represents the date a content item goes live.

# ContentData.HistoryId Property

Gets or sets the content item's history ID for the ContentData (see page 1378) object.

**C#**

```
public Long HistoryId;
```

**Visual Basic**

```
Public Property HistoryId() As Long
```

**Returns**

The numeric ID (Long) of the content item's history.

# ContentData.HyperLink Property

Gets or sets the content item's Hyperlink in the CMS400.NET for the ContentData (see page 1378) object.

**C#**

```
public String HyperLink;
```

**Visual Basic**

```
Public Property HyperLink() As String
```

**Returns**

A content item's Hyperlink information.

# ContentData.InheritedFrom Property

Gets or sets the folder ID from which to inherit permissions for the ContentData (see page 1378) object.

**C#**

```
public Long InheritedFrom;
```

**Visual Basic**

```
Public Property InheritedFrom() As Long
```

**Returns**

An numeric ID representing the folder from which permissions are inherited.

# ContentData.IsInherited Property

Gets or sets whether the content item inherits permissions from the folder settings for the ContentData (see page 1378) object.

**C#**

```
public Boolean IsInherited;
```

**Visual Basic**

```
Public Property IsInherited() As Boolean
```

**Returns**

A boolean that returns true when the content item inherits permissions from the folder settings.

# ContentData.IsPublished Property

Gets or sets whether the content is in a published state for the ContentData (see page 1378) object.

**C#**

```
public Boolean IsPublished;
```

**Visual Basic**

```
Public Property IsPublished() As Boolean
```

**Returns**

Returns a boolean based on whether the content is in a published state. True = Published

# ContentData.IsSearchable Property

Gets or sets whether the content is searchable for the ContentData (see page 1378) object. True = the content searchable.

**C#**

```
public Boolean IsSearchable;
```

**Visual Basic**

```
Public Property IsSearchable() As Boolean
```

**Returns**

A boolean based on whether the content is searchable.

- **True** = content is searchable.
- **False** = content is not searchable.

**Description**

# ContentData.IsXmlInherited Property

Gets or sets whether the XML Smart Form (see page 836) configuration is inherited from the folder. True = XML Smart Form (see page 836) configuration is inherited from the folder. This is for the ContentData (see page 1378) object.

**C#**

```
public Boolean IsXmlInherited;
```

**Visual Basic**

```
Public Property IsXmlInherited() As Boolean
```

**Returns**

A boolean based on whether the XML Smart Form (see page 836) configuration is inherited from the folder.

- **True** = content XML Smart Form (see page 836) configuration is inherited from the folder.
- **False** = content XML Smart Form (see page 836) configuration is not inherited from the folder.

# ContentData.LastEditDate Property

Gets or sets the content's last edit date for the ContentData (see page 1378) object.

**C#**

```
public String LastEditDate;
```

**Visual Basic**

```
Public Property LastEditDate() As String
```

**Returns**

Content (see page 788)'s last edit date as a string.

# ContentData.ManualAlias Property

Gets or sets the manual alias by its name for the ContentData (see page 1378) object. For example,

/~YourSite~/text.aspx.

**C#**

```
public String ManualAlias;
```

**Visual Basic**

```
Public Property ManualAlias() As String
```

**Returns**

A string value representing the name of a manual alias.

# ContentData.ManualAliasId Property

Gets or sets the ID of the of a manual alias for the ContentData (see page 1378) object.

**C#**

```
public Long ManualAliasId;
```

**Visual Basic**

```
Public Property ManualAliasId() As Long
```

**Returns**

A numeric value (Long) representing a manual alias ID.

# ContentData.MetaData Property

Gets or sets the Metadata of a content item as an array of ContentMetadata. This is for the ContentData (see page 1378) object.

**C#**

```
public ContentMetaData() MetaData;
```

**Visual Basic**

```
Public Property MetaData() As ContentMetaData()
```

**Returns**

Metadata as an array of ContentMetaData.

# ContentData.Path Property

Gets or sets the CMS400.NET path to content item for the ContentData (see page 1378) object.

**C#**

```
public String Path;
```

**Visual Basic**

```
Public Property Path() As String
```

**Returns**

A string value representing the CMS400.NET path to the content item.

# ContentData.Permissions Property

Gets or sets permission data for the ContentData (see page 1378) object.

**C#**

```
public PermissionData Permissions;
```

**Visual Basic**

```
Public Property Permissions() As PermissionData
```

**Returns**

Permissions as a PermissionData object.

# ContentData.Status Property

Gets or sets the letter value representing the status of a content item for the ContentData (see page 1378) object.

- **A** = Approved
- **O** = Checked Out
- **I** = Checked In
- **S** = Submitted for Approval
- **M** = Marked for Deletion
- **P** = Pending Go Live Date
- **T** = Awaiting Completion of Associated Tasks
- **D** = Pending Deletion

**C#**

```
public String Status;
```

**Visual Basic**

```
Public Property Status() As String
```

**Returns**

A letter value representing the status of a content item.

# ContentData.StyleSheet Property

Gets or sets the style sheet to use with a content item for the ContentData (see page 1378) object.

**C#**

```
public String StyleSheet;
```

**Visual Basic**

```
Public Property StyleSheet() As String
```

**Returns**

A string value representing the style sheet to use with a content item.

# ContentData.TemplateConfiguration Property

Gets or sets the content item's template configuration for the ContentData (see page 1378) object.

**C#**

```
public TemplateData TemplateConfiguration;
```

**Visual Basic**

```
Public Property TemplateConfiguration() As TemplateData
```

**Returns**

A content item's template configuration as TemplateData.

# ContentData.UserId Property

Gets or sets the ID of the user for the ContentData (see page 1378) object.

**C#**

```
public Long UserId;
```

**Visual Basic**

```
Public Property UserId() As Long
```

**Returns**

User (see page 913) ID as a Long.

# ContentData.UserName Property

Important: This property is obsolete. To retrieve the user name, call EkUser GetActiveUserbyID with the correct User (see page 913) ID.

Gets or sets the content's username for the ContentData (see page 1378) object.

**C#**

```
[Obsolete("This property is no longer valid.  To retrieve the user name, call EkUser
GetActiveUserbyID with the correct User ID")]
public String UserName;
```

**Visual Basic**

```
<Obsolete("This property is no longer valid.  To retrieve the user name, call EkUser
GetActiveUserbyID with the correct User ID")>
Public Property UserName() As String
```

**Returns**

The content's username.

# ContentData.XmlConfiguration Property

Gets or sets the content item's XML configuration for the ContentData (see page 1378) object.

**C#**

```
public XmlConfigData XmlConfiguration;
```

**Visual Basic**

```
Public Property XmlConfiguration() As XmlConfigData
```

**Returns**

A content item's XML configuration as XmlConfigData.

## ContentData.XmlInheritedFrom Property

Gets or sets the ID of the folder from which the XML Smart Form (see page 836) configuration is inherited. This is for the ContentData (see page 1378) object.

**C#**

```
public Long XmlInheritedFrom;
```

**Visual Basic**

```
Public Property XmlInheritedFrom() As Long
```

**Returns**

A numeric ID (Long) of the folder from which the XML Smart Form (see page 836) configuration is inherited.

# ContentRatingLabels Class

Provides a convinient value class

**Class Hierarchy**

Ektron.Cms.ContentRatingLabels

**C#**

```
public class ContentRatingLabels;
```

**Visual Basic**

```
Public Class ContentRatingLabels
```

**File**

ContentRatingLabels.vb

# ContentRatingLabels Properties

The properties of the ContentRatingLabels class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AlreadyRatedMessage (see page 1392) | The message displayed to a user when they rate content, but have already rated it. Defaults to "You have already rated this content." |

| | | |
|---|---|---|
| | BadLabel (see page 1392) | The label corresponding to a "Bad" content rating. This defaults to "Poor". |
| | CharsRemainingLabel (see page 1392) | The label indicating to the user how many more characters they may enter into the 'comments' section of the rating form. Defaults to "Characters Remaining: ". |
| | CookiesRequiredMessage (see page 1392) | The message displayed to a user when they rate content, but they have cookies disabled. Defaults to "We're sorry, but you must enable cookies in your browser to rate this content." |
| | FeedbackHeader (see page 1393) | The label corresponding to how the user would rate the content block. Defaults to "How would you rate the usefullness of this content?". |
| | GoodLabel (see page 1393) | The label corresponding to a "Good" content rating. This defaults to "Outstanding". |
| | RatingLevelLabel (see page 1393) | The label used to indicate the meaning of the rating displayed to the user, e.g. 7.8/9. Defaults to "Rating Level: ". |
| | ThankYouMessage (see page 1393) | The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback.". |
| | TotalRatingsLabel (see page 1393) | The label used to indicate the meaning of the total ratings displayed to the user, e.g. "Total Ratings: 986". Defaults to "Total Ratings: " |
| | UserCommentsHeader (see page 1394) | The label corresponding to why the user rated content the way they did. Defaults to "Tell us why you rated the content this way". |
| | ValidationMessage (see page 1394) | The label displayed to a user when they try to submit a content rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.". |

# ContentRatingLabels.AlreadyRatedMessage Property

The message displayed to a user when they rate content, but have already rated it. Defaults to "You have already rated this content."

**C#**

```
public String AlreadyRatedMessage;
```

**Visual Basic**

```
Public Property AlreadyRatedMessage() As String
```

# ContentRatingLabels.BadLabel Property

The label corresponding to a "Bad" content rating. This defaults to "Poor".

**C#**

```
public String BadLabel;
```

**Visual Basic**

```
Public Property BadLabel() As String
```

# ContentRatingLabels.CharsRemainingLabel Property

The label indicating to the user how many more characters they may enter into the 'comments' section of the rating form. Defaults to "Characters Remaining: ".

**C#**

```
public String CharsRemainingLabel;
```

**Visual Basic**

```
Public Property CharsRemainingLabel() As String
```

# ContentRatingLabels.CookiesRequiredMessage Property

The message displayed to a user when they rate content, but they have cookies disabled. Defaults to "We're sorry, but you must

enable cookies in your browser to rate this content."

**C#**

```
public String CookiesRequiredMessage;
```

**Visual Basic**

```
Public Property CookiesRequiredMessage() As String
```

# ContentRatingLabels.FeedbackHeader Property

The label corresponding to how the user would rate the content block. Defaults to "How would you rate the usefullness of this content?".

**C#**

```
public String FeedbackHeader;
```

**Visual Basic**

```
Public Property FeedbackHeader() As String
```

# ContentRatingLabels.GoodLabel Property

The label corresponding to a "Good" content rating. This defaults to "Outstanding".

**C#**

```
public String GoodLabel;
```

**Visual Basic**

```
Public Property GoodLabel() As String
```

# ContentRatingLabels.RatingLevelLabel Property

The label used to indicate the meaning of the rating displayed to the user, e.g. 7.8/9. Defaults to "Rating Level: ".

**C#**

```
public String RatingLevelLabel;
```

**Visual Basic**

```
Public Property RatingLevelLabel() As String
```

# ContentRatingLabels.ThankYouMessage Property

The label displayed to a user thanking them for their content rating. Defaults to "Thank you for your feedback.".

**C#**

```
public String ThankYouMessage;
```

**Visual Basic**

```
Public Property ThankYouMessage() As String
```

# ContentRatingLabels.TotalRatingsLabel Property

The label used to indicate the meaning of the total ratings displayed to the user, e.g. "Total Ratings: 986". Defaults to "Total

Ratings: "

**C#**

```
public String TotalRatingsLabel;
```

**Visual Basic**

```
Public Property TotalRatingsLabel() As String
```

## ContentRatingLabels.UserCommentsHeader Property

The label corresponding to why the user rated content the way they did. Defaults to "Tell us why you rated the content this way".

**C#**

```
public String UserCommentsHeader;
```

**Visual Basic**

```
Public Property UserCommentsHeader() As String
```

## ContentRatingLabels.ValidationMessage Property

The label displayed to a user when they try to submit a content rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.".

**C#**

```
public String ValidationMessage;
```

**Visual Basic**

```
Public Property ValidationMessage() As String
```

# FormData Class

Describes a form design.

Declaration Example:

```
Dim formItem As Ektron.Cms.FormData
```

**Class Hierarchy**

**C#**

```
public class FormData : ContentData;
```

**Visual Basic**

```
Public Class FormData
Inherits ContentData
```

**File**

FormData.vb

**Example**

```
Dim formApi As New Ektron.Cms.Content.Form
```

```
Dim formItem As Ektron.Cms.FormData
formItem = formApi.GetForm(CInt(2))
```

# FormData Properties

The properties of the FormData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | Autofill (see page 1395) | Specifies whether the form data should be automatically filled if the user is known and has previously entered data for this form. |
| | Description (see page 1396) | Gets or sets a description of the form. The description must be 255 characters or less. |
| | MailCc (see page 1396) | Gets or sets the email address(es) to copy an email to when form data is submitted. Multiple email addresses should be separated by semicolons (";"). |
| | MailFrom (see page 1396) | Gets or sets the email address assigned as the sender when email is sent upon submittal of form data. |
| | MailPreamble (see page 1396) | Gets or sets the text that precedes the submitted data in the body of the email sent when form data is submitted. Maximum length is 128 characters. |
| | MailSubject (see page 1397) | Gets or sets the subject of the email sent when form data is submitted. Maximum length is 128 characters. |
| | MailTo (see page 1397) | Gets or sets the email address(es) to send an email to when form data is submitted. Multiple email addresses should be separated by semicolons (";"). |
| | PollRevision (see page 1397) | Gets or sets the revision number of the poll as a string. |
| | SendXmlPacket (see page 1397) | Gets or sets whether the submitted form data should be emailed as well-formed XML data.<br>• **True** - formats the data by making each field name a tag around each value.<br>• **False** - sends the data without consideration whether it is well-formed XML. |
| | StoreDataTo (see page 1398) | Specifies how the data should be processed. Valid values include:<br>• **"db"** submitted data should be stored in the database.<br>• **"email"** submitted data should be included in an email sent to the configured address(es).<br>• **"both"** submitted data should stored in the database AND sent in an email. |
| | Validation (see page 1398) | Gets or sets the form's validation information as an array of FormValidationData. |

# FormData.Autofill Property

Specifies whether the form data should be automatically filled if the user is known and has previously entered data for this form.

**C#**

```
public Boolean Autofill;
```

**Visual Basic**

```
Public Property Autofill() As Boolean
```

**Returns**

True if form data should be automatically filled. False if the form should always be blank.

**Remarks**

Requires that data be stored in the database.

# FormData.Description Property

Gets or sets a description of the form. The description must be 255 characters or less.

**C#**

```
public String Description;
```

**Visual Basic**

```
Public Property Description() As String
```

**Remarks**

The string should be plain text. HTML tags should not be used.

# FormData.MailCc Property

Gets or sets the email address(es) to copy an email to when form data is submitted. Multiple email addresses should be separated by semicolons (";").

**C#**

```
public String MailCc;
```

**Visual Basic**

```
Public Property MailCc() As String
```

# FormData.MailFrom Property

Gets or sets the email address assigned as the sender when email is sent upon submittal of form data.

**C#**

```
public String MailFrom;
```

**Visual Basic**

```
Public Property MailFrom() As String
```

# FormData.MailPreamble Property

Gets or sets the text that precedes the submitted data in the body of the email sent when form data is submitted. Maximum length is 128 characters.

**C#**

```
public String MailPreamble;
```

**Visual Basic**

```
Public Property MailPreamble() As String
```

**Remarks**

The string should be plain text. HTML tags should not be used.

# FormData.MailSubject Property

Gets or sets the subject of the email sent when form data is submitted. Maximum length is 128 characters.

**C#**

```
public String MailSubject;
```

**Visual Basic**

```
Public Property MailSubject() As String
```

# FormData.MailTo Property

Gets or sets the email address(es) to send an email to when form data is submitted. Multiple email addresses should be separated by semicolons (";").

**C#**

```
public String MailTo;
```

**Visual Basic**

```
Public Property MailTo() As String
```

**Remarks**

Applies only when StoreDataTo (see page 1398) is "email" or "both".

# FormData.PollRevision Property

Gets or sets the revision number of the poll as a string.

**C#**

```
public String PollRevision;
```

**Visual Basic**

```
Public Property PollRevision() As String
```

**Returns**

string

**Remarks**

It is used in the browser cookie when users load the poll on a web page.

# FormData.SendXmlPacket Property

Gets or sets whether the submitted form data should be emailed as well-formed XML data.

- **True** - formats the data by making each field name a tag around each value.
- **False** - sends the data without consideration whether it is well-formed XML.

**C#**

```
public Boolean SendXmlPacket;
```

**Visual Basic**

```
Public Property SendXmlPacket() As Boolean
```

**Remarks**

The XML structure of the form data submitted is:

```
<cmsMail>
    <FormTitle>This is the Title</FormTitle>
    <FormDescription>This is the description.</FormDescription>
    <date>7/11/2005 5:24:58 PM</date>
    <MailTo></MailTo>
    <MailFrom></MailFrom>
    <MailCc></MailCc>
    <MailSubject></MailSubject>
    <MailPreamble></MailPreamble>
    <Data>
      <Field1>Value of Field 1</Field1>
      <Field2>Value of Field 2</Field2>
      <Field3>Value of Field 3</Field3>
    </Data>
</cmsMail>
```

**Note:** The <date> is NOT ISO 8601. It's format is dependent on the web server.

# FormData.StoreDataTo Property

Specifies how the data should be processed. Valid values include:

- **"db"** submitted data should be stored in the database.
- **"email"** submitted data should be included in an email sent to the configured address(es).
- **"both"** submitted data should stored in the database AND sent in an email.

**C#**

```
public String StoreDataTo;
```

**Visual Basic**

```
Public Property StoreDataTo() As String
```

# FormData.Validation Property

Gets or sets the form's validation information as an array of FormValidationData.

**C#**

```
public FormValidationData() Validation;
```

**Visual Basic**

```
Public Property Validation() As FormValidationData()
```

**Returns**

Ektron.Cms.FormValidationData()

# FormFieldDataItem Class

Describes an instance of a data value from an individual form field. The data is a name/value pair.

**Class Hierarchy**

**C#**

```
public class FormFieldDataItem;
```

**Visual Basic**

```
Public Class FormFieldDataItem
```

**File**

FormFieldDataItem.vb

**Remarks**

Typically, this class is used in an array.

# FormFieldDataItem Properties

The properties of the FormFieldDataItem class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DataValue (see page 1399) | Gets or sets the value of the form field. |
| | FieldName (see page 1399) | Gets or sets the name of the form field. |

# FormFieldDataItem.DataValue Property

Gets or sets the value of the form field.

**C#**

```
public Object DataValue;
```

**Visual Basic**

```
Public Property DataValue() As Object
```

**Remarks**

The data type of the value is variant (i.e., an Object).

# FormFieldDataItem.FieldName Property

Gets or sets the name of the form field.

**C#**

```
public String FieldName;
```

**Visual Basic**

```
Public Property FieldName() As String
```

# FormFieldDatalist Class

Describes an individual datalist suitable as an enumeration for a select list.

**Class Hierarchy**

**C#**

```
public class FormFieldDatalist;
```

**Visual Basic**

```
Public Class FormFieldDatalist
```

**File**

FormFieldDatalist.vb

**Remarks**

Typically used as an array in class FormFieldList (see page 1407).

Typically referenced by name in class FormFieldDefinition (see page 1404).

# FormFieldDatalist Properties

The properties of the FormFieldDatalist class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | CaptionXPath (see page 1400) | Gets or sets the CaptionXPath of the pre-defined datalist. |
| | DatalistItems (see page 1401) | Gets or sets the array of items defined in the datalist. |
| | DataNamespaces (see page 1401) | Gets or sets the Data Namespace of the pre-defined datalist. |
| | DataSelect (see page 1401) | Gets or sets the Data Select of the pre-defined datalist. |
| | DataSrc (see page 1401) | Gets or sets the Data Source of the pre-defined datalist. |
| | Language (see page 1402) | Gets or sets the language code of the datalist. For example, en or en-US. |
| | Name (see page 1402) | Gets or sets the name of the datalist. |
| | ValueXPath (see page 1402) | Gets or sets the Value XPath of the pre-defined datalist. |

# FormFieldDatalist.CaptionXPath Property

Gets or sets the CaptionXPath of the pre-defined datalist.

**C#**

```
[XmlAttributeAttribute("ektdesignns_captionxpath")]
public String CaptionXPath;
```

**Visual Basic**

```
<XmlAttributeAttribute("ektdesignns_captionxpath")>
Public Property CaptionXPath() As String
```

# FormFieldDatalist.DatalistItems Property

Gets or sets the array of items defined in the datalist.

**C#**

```
[XmlElement("item")]
public FormFieldDatalistItem() DatalistItems;
```

**Visual Basic**

```
<XmlElement("item")>
Public Property DatalistItems() As FormFieldDatalistItem()
```

# FormFieldDatalist.DataNamespaces Property

Gets or sets the Data Namespace of the pre-defined datalist.

**C#**

```
[XmlAttributeAttribute("ektdesignns_datanamespaces")]
public String DataNamespaces;
```

**Visual Basic**

```
<XmlAttributeAttribute("ektdesignns_datanamespaces")>
Public Property DataNamespaces() As String
```

# FormFieldDatalist.DataSelect Property

Gets or sets the Data Select of the pre-defined datalist.

**C#**

```
[XmlAttributeAttribute("ektdesignns_dataselect")]
public String DataSelect;
```

**Visual Basic**

```
<XmlAttributeAttribute("ektdesignns_dataselect")>
Public Property DataSelect() As String
```

# FormFieldDatalist.DataSrc Property

Gets or sets the Data Source of the pre-defined datalist.

**C#**

```
[XmlAttributeAttribute("ektdesignns_datasrc")]
public String DataSrc;
```

**Visual Basic**

```
<XmlAttributeAttribute("ektdesignns_datasrc")>
```

```
Public Property DataSrc() As String
```

## FormFieldDatalist.Language Property

Gets or sets the language code of the datalist. For example, en or en-US.

**C#**

```
[XmlAttributeAttribute("xml:lang")]
public String Language;
```

**Visual Basic**

```
<XmlAttributeAttribute("xml:lang")>
Public Property Language() As String
```

**Returns**

Language code.

**Remarks**

The first two letters of the language code conform to ISO 639. The last two letters are an optional country code that conforms to ISO 3166.

## FormFieldDatalist.Name Property

Gets or sets the name of the datalist.

**C#**

```
[XmlAttributeAttribute("name")]
public String Name;
```

**Visual Basic**

```
<XmlAttributeAttribute("name")>
Public Property Name() As String
```

## FormFieldDatalist.ValueXPath Property

Gets or sets the Value XPath of the pre-defined datalist.

**C#**

```
[XmlAttributeAttribute("ektdesignns_valuexpath")]
public String ValueXPath;
```

**Visual Basic**

```
<XmlAttributeAttribute("ektdesignns_valuexpath")>
Public Property ValueXPath() As String
```

# FormFieldDatalistItem Class

Describes an individual datalist item (most importantly its value and display value).

**Class Hierarchy**

Ektron.Cms.FormFieldDatalistItem

**C#**

```
public class FormFieldDatalistItem;
```

**Visual Basic**

```
Public Class FormFieldDatalistItem
```

**File**

FormFieldDatalistItem.vb

**Remarks**

Typically used as an array in class FormFieldDatalist (see page 1400) Cms.FormFieldDatalist (see page 1400)

# FormFieldDatalistItem Properties

The properties of the FormFieldDatalistItem class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DataValue (see page 1403) | Gets or sets the value for this item in the data list. |
| | DisplayValue (see page 1403) | Gets or sets the value to be displayed. |
| | IsDefault (see page 1404) | Gets or sets whether this item is the default value in the datalist. Set to **True** if this item is a default value for the list. |
| | IsEnabled (see page 1404) | Gets or sets whether this item is enabled for selection. Set to **True** if this item is enabled. |

# FormFieldDatalistItem.DataValue Property

Gets or sets the value for this item in the data list.

**C#**

```
[XmlAttributeAttribute("value")]
public String DataValue;
```

**Visual Basic**

```
<XmlAttributeAttribute("value")>
Public Property DataValue() As String
```

**Remarks**

This value is typically language independent and suitable for machine processing.

# FormFieldDatalistItem.DisplayValue Property

Gets or sets the value to be displayed.

**C#**

```
[XmlText()]
public String DisplayValue;
```

**Visual Basic**

```
<XmlText()>
Public Property DisplayValue() As String
```

**Remarks**

This value is typically dependent on the language and suitable for human reading.

## FormFieldDatalistItem.IsDefault Property

Gets or sets whether this item is the default value in the datalist. Set to **True** if this item is a default value for the list.

**C#**

```
[XmlAttributeAttribute("default")]
public Boolean IsDefault;
```

**Visual Basic**

```
<XmlAttributeAttribute("default")>
Public Property IsDefault() As Boolean
```

**Remarks**

If a list allows only one item to be selected at a time, then only one item should be marked as the default. If more than one may be selected, then it is reasonable to have more than one default.

## FormFieldDatalistItem.IsEnabled Property

Gets or sets whether this item is enabled for selection. Set to **True** if this item is enabled.

**C#**

```
[XmlAttributeAttribute("enabled")]
public Boolean IsEnabled;
```

**Visual Basic**

```
<XmlAttributeAttribute("enabled")>
Public Property IsEnabled() As Boolean
```

**Remarks**

An item may be disabled to prevent it from being selected. Typically an item, once defined and used, would not be deleted from the list, but simply disabled. This preserves the display value of the item for backward compatibility.

# FormFieldDefinition Class

Describes the definition of an individual form field.

**Class Hierarchy**

**C#**

```
public class FormFieldDefinition;
```

**Visual Basic**

```
Public Class FormFieldDefinition
```

**File**

FormFieldDefinition.vb

**Remarks**

Typically used as an array in class FormFieldList (see page 1407).

# FormFieldDefinition Properties

The properties of the FormFieldDefinition class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Basetype (see page 1405) | Gets or sets the basic datatype of the field. |
| | ContentType (see page 1405) | Gets or sets the type of content allowed within the field. This is loosely associated with the datatype. |
| | DatalistName (see page 1406) | Gets or sets the name of the datalist, if one is defined for this field. |
| | Datatype (see page 1406) | Gets or sets the datatype of the field. |
| | DisplayName (see page 1406) | Gets or sets the name of the field to be displayed. |
| | FieldName (see page 1407) | Gets or sets the name of the field. |
| | Language (see page 1407) | Gets or sets the language code of the datalist. For example, en or en-US. |
| | XPath (see page 1407) | Gets or sets the XPath expression that locates the field. |

# FormFieldDefinition.Basetype Property

Gets or sets the basic datatype of the field.

**C#**

```
[XmlAttributeAttribute("basetype")]
public String Basetype;
```

**Visual Basic**

```
<XmlAttributeAttribute("basetype")>
Public Property Basetype() As String
```

**Remarks**

The possible values of the basic datatype vary, including, text, textbox, calendar and checkbox.

# FormFieldDefinition.ContentType Property

Gets or sets the type of content allowed within the field. This is loosely associated with the datatype.

**C#**

```
[XmlAttributeAttribute("content")]
public String ContentType;
```

**Visual Basic**

```
<XmlAttributeAttribute("content")>
Public Property ContentType() As String
```

**Remarks**

The possible values for content vary.

# FormFieldDefinition.DatalistName Property

Gets or sets the name of the datalist, if one is defined for this field.

**C#**

```
[XmlAttributeAttribute("datalist")]
public String DatalistName;
```

**Visual Basic**

```
<XmlAttributeAttribute("datalist")>
Public Property DatalistName() As String
```

**Description**

Name of the associated datalist.

**Remarks**

Applies only to fields whose values are selected from a list. The allowed values are defined in the datalist.

# FormFieldDefinition.Datatype Property

Gets or sets the datatype of the field.

**C#**

```
[XmlAttributeAttribute("datatype")]
public String Datatype;
```

**Visual Basic**

```
<XmlAttributeAttribute("datatype")>
Public Property Datatype() As String
```

**Remarks**

The possible values of the datatype vary. Generally they conform to W3C XML Schema types such as 'string' and 'date'. Other values include 'selection' and 'choice'.

# FormFieldDefinition.DisplayName Property

Gets or sets the name of the field to be displayed.

**C#**

```
[XmlText()]
public String DisplayName;
```

**Visual Basic**

```
<XmlText()>
Public Property DisplayName() As String
```

**Remarks**

The name is suitable for human reading.

## FormFieldDefinition.FieldName Property

Gets or sets the name of the field.

**C#**

```
[XmlAttributeAttribute("name")]
public String FieldName;
```

**Visual Basic**

```
<XmlAttributeAttribute("name")>
Public Property FieldName() As String
```

**Remarks**

The name is suitable for machine processing, such as with an XML parser.

## FormFieldDefinition.Language Property

Gets or sets the language code of the datalist. For example, en or en-US.

**C#**

```
[XmlAttributeAttribute("xml:lang")]
public String Language;
```

**Visual Basic**

```
<XmlAttributeAttribute("xml:lang")>
Public Property Language() As String
```

**Remarks**

The first two letters of the language code conform to ISO 639. The last two letters are an optional country code that conforms to ISO 3166.

## FormFieldDefinition.XPath Property

Gets or sets the XPath expression that locates the field.

**C#**

```
[XmlAttributeAttribute("xpath")]
public String XPath;
```

**Visual Basic**

```
<XmlAttributeAttribute("xpath")>
Public Property XPath() As String
```

# FormFieldList Class

Describes the definition of all the fields in a specific form.

**Class Hierarchy**

**C#**

```
[XmlRoot("fieldlist")]
public class FormFieldList;
```

**Visual Basic**

```
<XmlRoot("fieldlist")>
Public Class FormFieldList
```

**File**

FormFieldList.vb

# FormFieldList Properties

The properties of the FormFieldList class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Datalists (see page 1408) | Gets or sets an array of defined datalists. |
| | Fields (see page 1408) | Gets or sets an array of field definitions. |

# FormFieldList.Datalists Property

Gets or sets an array of defined datalists.

**C#**

```
[XmlElement("datalist")]
public FormFieldDatalist() Datalists;
```

**Visual Basic**

```
<XmlElement("datalist")>
Public Property Datalists() As FormFieldDatalist()
```

**Returns**

Ektron.Cms.FormFieldDatalist (see page 1400)

# FormFieldList.Fields Property

Gets or sets an array of field definitions.

**C#**

```
[XmlElement("field")]
public FormFieldDefinition() Fields;
```

**Visual Basic**

```
<XmlElement("field")>
Public Property Fields() As FormFieldDefinition()
```

**Returns**

Ektron.Cms.FormFieldDefinition (see page 1404)

# FormRequestData Class

A data object that contains information about a form.

### Class Hierarchy

### C#

```
public class FormRequestData;
```

### Visual Basic

```
Public Class FormRequestData
```

### File

FormRequestData.vb

# FormRequestData Properties

The properties of the FormRequestData class are listed here.

### Public Properties

| | Name | Description |
|---|---|---|
| | AutoFill (see page 1410) | Indicates whether a form's fields are pre-filled when the form is displayed.<br>• **True** - a form's fields are pre-filled when the form is displayed.<br>• **False** - a form's fields are empty when the form is displayed. |
| | FieldList (see page 1410) | Gets or sets a list of the fields defined in the form. |
| | Id (see page 1410) | The numeric ID of the form. |
| | IsPoll (see page 1410) | Gets whether the form is used as poll.<br>• **True** - this form is a poll.<br>• **False** - this form is not a poll. |
| | Language (see page 1411) | Gets or sets the form's language ID. For example, 1033 for English, 1034 for Spanish or 1036 for French. When set to 0 (zero), the language is dynamic. |
| | PostBackData (see page 1411) | Gets or sets the data submitted from the form as a collection. |
| | PreFillFielddata (see page 1411) | Gets or sets the data values that are used to pre-fill a form's fields. When you display a form, this is the information pre-loaded into each field.<br>For example, in a Country: field you might have "United States" set to pre-fill the field. This way, people from the United States don't have to type it in each time. Another example is when a user is logged in and fills out a form more that once, his name and and email address always appear in their respective fields after completing the form the first time. |
| | TagRequired (see page 1412) | Gets or sets whether to wrap form tags around the form element.<br>• **True** - wrap form tags around the form element.<br>• **False** - Do not wrap form tags around the form element. |

# FormRequestData.AutoFill Property

Indicates whether a form's fields are pre-filled when the form is displayed.

- **True** - a form's fields are pre-filled when the form is displayed.
- **False** - a form's fields are empty when the form is displayed.

**C#**

```
public Boolean AutoFill;
```

**Visual Basic**

```
Public Property AutoFill() As Boolean
```

# FormRequestData.FieldList Property

Gets or sets a list of the fields defined in the form.

**C#**

```
public FormFieldList FieldList;
```

**Visual Basic**

```
Public Property FieldList() As FormFieldList
```

**Returns**

Ektron.Cms.FormFieldList (see page 1407)

# FormRequestData.Id Property

The numeric ID of the form.

**C#**

```
public Long Id;
```

**Visual Basic**

```
Public Property Id() As Long
```

**Returns**

The numeric ID (Long) of the form.

# FormRequestData.IsPoll Property

Gets whether the form is used as poll.

- **True** - this form is a poll.
- **False** - this form is not a poll.

**C#**

```
public Boolean IsPoll;
```

**Visual Basic**

```
Public Property IsPoll() As Boolean
```

**Remarks**

This should only be read not set.

# FormRequestData.Language Property

Gets or sets the form's language ID. For example, 1033 for English, 1034 for Spanish or 1036 for French. When set to 0 (zero), the language is dynamic.

**C#**

```
public int Language;
```

**Visual Basic**

```
Public Property Language() As Integer
```

**Returns**

The form's language ID as an integer.

# FormRequestData.PostBackData Property

Gets or sets the data submitted from the form as a collection.

**C#**

```
public System.Collections.Specialized.NameValueCollection PostBackData;
```

**Visual Basic**

```
Public Property PostBackData() As System.Collections.Specialized.NameValueCollection
```

**Returns**

System.Collections.Specialized.NameValueCollection

# FormRequestData.PreFillFielddata Property

Gets or sets the data values that are used to pre-fill a form's fields. When you display a form, this is the information pre-loaded into each field.

For example, in a Country: field you might have "United States" set to pre-fill the field. This way, people from the United States don't have to type it in each time. Another example is when a user is logged in and fills out a form more that once, his name and and email address always appear in their respective fields after completing the form the first time.

**C#**

```
public FormFieldDataItem() PreFillFielddata;
```

**Visual Basic**

```
Public Property PreFillFielddata() As FormFieldDataItem()
```

**Remarks**

If the AutoFill (see page 1410) property is set to false, this property appears as empty.

## FormRequestData.TagRequired Property

Gets or sets whether to wrap form tags around the form element.

- **True** - wrap form tags around the form element.
- **False** - Do not wrap form tags around the form element.

**C#**

```
public Boolean TagRequired;
```

**Visual Basic**

```
Public Property TagRequired() As Boolean
```

**Remarks**

Typically set to false.

# FormSubmittedData Class

Describes the data of all the fields submitted from a form at a specific time.

**Class Hierarchy**

Ektron.Cms.FormSubmittedData

**C#**

```
public class FormSubmittedData;
```

**Visual Basic**

```
Public Class FormSubmittedData
```

**File**

FormSubmittedData.vb

**Description**

**Remarks**

This class may be used in an array to describe the total of all the data submitted from a form.

# FormSubmittedData Properties

The properties of the FormSubmittedData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | DataItems (see page 1413) | Gets or sets the array of the data associated with each field defined in the form. |
| | DateSubmitted (see page 1413) | Gets or sets the date the data in the DataItems (see page 1413)() array was submitted. |

| | | |
|---|---|---|
| | FormDataID (see page 1413) | Gets or sets the numeric ID for the form data. |
| | UserId (see page 1413) | Gets or sets the user ID of the person who submitted the data. A value of 0 means anonymous. |
| | UserInfo (see page 1414) | Gets or sets the user information as an Ektron.Cms.UserData (see page 1419) object. |

# FormSubmittedData.DataItems Property

Gets or sets the array of the data associated with each field defined in the form.

**C#**

```
public FormFieldDataItem() DataItems;
```

**Visual Basic**

```
Public Property DataItems() As FormFieldDataItem()
```

# FormSubmittedData.DateSubmitted Property

Gets or sets the date the data in the DataItems (see page 1413)() array was submitted.

**C#**

```
public Date DateSubmitted;
```

**Visual Basic**

```
Public Property DateSubmitted() As Date
```

# FormSubmittedData.FormDataID Property

Gets or sets the numeric ID for the form data.

**C#**

```
public Long FormDataID;
```

**Visual Basic**

```
Public Property FormDataID() As Long
```

**Remarks**

A value of -1 means the ID is undefined.

# FormSubmittedData.UserId Property

Gets or sets the user ID of the person who submitted the data. A value of 0 means anonymous.

**C#**

```
public Long UserId;
```

**Visual Basic**

```
Public Property UserId() As Long
```

**Remarks**

UserId and UserInfo (see page 1414) are not kept synchronized in this class.

## FormSubmittedData.UserInfo Property

Gets or sets the user information as an Ektron.Cms.UserData (see page 1419) object.

**C#**

```
public UserData UserInfo;
```

**Visual Basic**

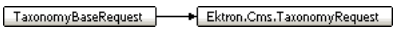
```
Public Property UserInfo() As UserData
```

**Remarks**

UserId (see page 1413) and UserInfo are not kept synchronized in this class.

# TaxonomyRequest Class

**Class Hierarchy**




**C#**

```
[Serializable()]
public class TaxonomyRequest : TaxonomyBaseRequest;
```

**Visual Basic**

```
<Serializable()>
Public Class TaxonomyRequest
Inherits TaxonomyBaseRequest
```

**File**

TaxonomyData.vb

**Description**

This is class Ektron.Cms.TaxonomyRequest.

# TaxonomyRequest Properties

The properties of the TaxonomyRequest class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Depth (see page 1415) | Gets or sets number of taxonomy levels to retrieve below each taxonomy/category. For example, if the taxonomy is Businesses > Restaurants > Pizza , and you set this property to 2, only Business and Restaurants are available in code behind.<br>To retrieve the all of the categories for a taxonomy recursively, enter -1.<br>The default value is 1.<br>**Important!** - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth. |
| | FolderId (see page 1415) | Deprecated. Not in use. |

| | | |
|---|---|---|
| | IncludeItems (see page 1415) | When True, include all items from the taxonomy's structure. When false, use only the taxonomy category's structure. |
| | Preview (see page 1416) | Gets or sets whether preview content should be included in the taxonomy category. |
| | ReadCount (see page 1416) | Gets or sets the quantity of items in each taxonomy category. |
| | TaxonomyIdList (see page 1416) | Gets or sets a list of taxonomy items IDs associated with content. |
| | TaxonomyItemType (see page 1417) | Gets or sets the item type being assigned to the taxonomy.<br>• Content (see page 788) = 0<br>• User (see page 913) = 1<br>• Group = 2<br>• Blogs = 3<br>• Forum (see page 256) = 4 |
| | TaxonomyPath (see page 1417) | Gets or set the path of the taxonomy category. For example, "\New Hampshire\Restaurants\Pizza" |

# TaxonomyRequest.Depth Property

Gets or sets number of taxonomy levels to retrieve below each taxonomy/category. For example, if the taxonomy is Businesses > Restaurants > Pizza , and you set this property to 2, only Business and Restaurants are available in code behind.

To retrieve the all of the categories for a taxonomy recursively, enter -1.

The default value is 1.

**Important!** - For a live site, Ektron strongly recommends leaving this value at 1. Increasing this value can slow down your live Web server. However, for testing on a staging server, you can increase the depth.

**C#**

```
public int Depth;
```

**Visual Basic**

```
Public Property Depth() As Integer
```

**Returns**

An integer representing the depth level.

**Remarks**

This property should not be left empty. Enter a value at least a value of 1.

# TaxonomyRequest.FolderId Property

Deprecated. Not in use.

**C#**

```
public Long FolderId;
```

**Visual Basic**

```
Public Property FolderId() As Long
```

# TaxonomyRequest.IncludeItems Property

When True, include all items from the taxonomy's structure. When false, use only the taxonomy category's structure.

**C#**

```
public Boolean IncludeItems;
```

**Visual Basic**

```
Public Property IncludeItems() As Boolean
```

**Returns**

All items in the taxonomy's structure.

# TaxonomyRequest.Preview Property

Gets or sets whether preview content should be included in the taxonomy category.

**C#**

```
public Boolean Preview;
```

**Visual Basic**

```
Public Property Preview() As Boolean
```

**Returns**

A boolean.

# TaxonomyRequest.ReadCount Property

Gets or sets the quantity of items in each taxonomy category.

**C#**

```
public Boolean ReadCount;
```

**Visual Basic**

```
Public Property ReadCount() As Boolean
```

**Returns**

An integer representing the number of items.

# TaxonomyRequest.TaxonomyIdList Property

Gets or sets a list of taxonomy items IDs associated with content.

**C#**

```
public String TaxonomyIdList;
```

**Visual Basic**

```
Public Property TaxonomyIdList() As String
```

**Returns**

A list of taxonomy items.

## TaxonomyRequest.TaxonomyItemType Property

Gets or sets the item type being assigned to the taxonomy.

- Content (see page 788) = 0
- User (see page 913) = 1
- Group = 2
- Blogs = 3
- Forum (see page 256) = 4

**C#**

```
public TaxonomyItemType TaxonomyItemType;
```

**Visual Basic**

```
Public Property TaxonomyItemType() As TaxonomyItemType
```

**Returns**

The item's type.

## TaxonomyRequest.TaxonomyPath Property

Gets or set the path of the taxonomy category. For example, "\New Hampshire\Restaurants\Pizza"

**C#**

```
public String TaxonomyPath;
```

**Visual Basic**

```
Public Property TaxonomyPath() As String
```

**Returns**

The path of the taxonomy category.

# UriFavorite Class

A list of properties that make up a URI favorite data.

**Class Hierarchy**

**C#**

```
[Serializable()]
public class UriFavorite;
```

**Visual Basic**

```
<Serializable()>
Public Class UriFavorite
```

**File**

TaxonomyData.vb

# UriFavorite Properties

The properties of the UriFavorite class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | FavoriteDesc (see page 1418) | Gets or sets a description for a URI favorite. |
| | FavoriteId (see page 1418) | Gets or sets the numeric ID of the favorite. |
| | FavoriteName (see page 1418) | Gets or sets the name of the URI for the favorite. |
| | FavoriteUri (see page 1419) | Gets or sets the actual URI value of the favorite. |
| | UserId (see page 1419) | Gets or set the ID of the user with which the favorite is associated. |

# UriFavorite.FavoriteDesc Property

Gets or sets a description for a URI favorite.

**C#**

```
public String FavoriteDesc;
```

**Visual Basic**

```
Public Property FavoriteDesc() As String
```

**Returns**

A description of the URI favorite.

# UriFavorite.FavoriteId Property

Gets or sets the numeric ID of the favorite.

**C#**

```
public Long FavoriteId;
```

**Visual Basic**

```
Public Property FavoriteId() As Long
```

**Returns**

The numeric ID of a favorite.

# UriFavorite.FavoriteName Property

Gets or sets the name of the URI for the favorite.

**C#**

```
public String FavoriteName;
```

**Visual Basic**

```
Public Property FavoriteName() As String
```

**Returns**

The name of the URI.

## UriFavorite.FavoriteUri Property

Gets or sets the actual URI value of the favorite.

**C#**

```
public String FavoriteUri;
```

**Visual Basic**

```
Public Property FavoriteUri() As String
```

**Returns**

The URI value.

## UriFavorite.UserId Property

Gets or set the ID of the user with which the favorite is associated.

**C#**

```
public Long UserId;
```

**Visual Basic**

```
Public Property UserId() As Long
```

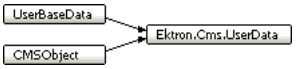
**Returns**

The ID of the user.

# UserData Class

A data object that contains the properties that make up a user.

**Class Hierarchy**




**C#**

```
public class UserData : UserBaseData, CMSObject;
```

**Visual Basic**

```
Public Class UserData
Inherits UserBaseData
Implements CMSObject
```

**File**

UserData.vb

# UserData Properties

The properties of the UserData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | AcceptedTerms (see page 1420) | Gets or set whether the user has accepted the Terms and Conditions associated with a Threaded Discussion. This is for the UserData (see page 1419) object. |
| | AdUserName (see page 1421) | Gets or sets the Active Directory (see page 189) user name for the UserData (see page 1419) object. |
| | DateCreated (see page 1421) | Gets or sets the creation date of a user. This is for the UserData (see page 1419) object. |
| | GroupId (see page 1421) | **The GroupId property is no longer valid. To retrieve a list of groups this User (see page 913) belongs to use Ektron.Cms.API.User.GetAllUserGroups() function.**<br>Gets or sets the numeruc ID (Long) of the group to which the user belongs. This is for the UserData (see page 1419) object. |
| | GroupName (see page 1421) | **The GroupName property is no longer valid. To retrieve a list of groups this User (see page 913) belongs to use Ektron.Cms.API.User.GetAllUserGroups() function.**<br>Gets or sets the name of the group to which the user belongs. This is for the UserData (see page 1419) object. |
| | IsMemberShip (see page 1422) | Gets or sets a boolean indicating whether the user is a Membership (see page 350) user or not. This is for UserData (see page 1419) object. |
| | LanguageId (see page 1422) | Gets or sets the integer ID of the language for the UserData (see page 1419) object. |
| | LanguageName (see page 1422) | Gets or set the name of the user's language for the UserData (see page 1419) object. |
| | LastLoginDate (see page 1423) | Gets or sets the last date a user logged into CMS400.NET. This is for the UserData (see page 1419) object. |
| | LoginAttempts (see page 1423) | Gets an Int16 representing the number of times a user has attempted to login for the UserData (see page 1419) object. This number resets to 0 (zero) once a user successfully logs in. |
| | Rank (see page 1423) | Gets or sets the user's rank for a Threaded Discussion for the UserData (see page 1419) object. |
| | Signature (see page 1423) | Gets or sets the user's signature for the UserData (see page 1419) object, used in the forums and other locations. |
| | UserPreference (see page 1424) | Gets or sets a user's preferences as UserPreferenceData for the UserData (see page 1419) object. |

# UserData.AcceptedTerms Property

Gets or set whether the user has accepted the Terms and Conditions associated with a Threaded Discussion. This is for the UserData (see page 1419) object.

**C#**

```
public Boolean AcceptedTerms;
```

**Visual Basic**

```
Public Property AcceptedTerms() As Boolean
```

**Returns**

A boolean indicating whether the user has accepted the Terms and Conditions associated with a Threaded Discussion.

- **True** = user has accepted Terms and Conditions.
- **False** = user has not accepted Terms and Conditions.

**Description**

# UserData.AdUserName Property

Gets or sets the Active Directory (see page 189) user name for the UserData (see page 1419) object.

**C#**

```
public String AdUserName;
```

**Visual Basic**

```
Public Property AdUserName() As String
```

**Returns**

The Active Directory (see page 189) user name.

# UserData.DateCreated Property

Gets or sets the creation date of a user. This is for the UserData (see page 1419) object.

**C#**

```
public String DateCreated;
```

**Visual Basic**

```
Public Property DateCreated() As String
```

**Returns**

The date the user was created.

# UserData.GroupId Property

**The GroupId property is no longer valid. To retrieve a list of groups this User (see page 913) belongs to use Ektron.Cms.API.User.GetAllUserGroups() function.**

Gets or sets the numeruc ID (Long) of the group to which the user belongs. This is for the UserData (see page 1419) object.

**C#**

```
[ObsoleteAttribute("The GroupId property is no longer valid.  To retrieve a list of groups
this User belongs to use Ektron.Cms.API.User.User.GetGroupsUserIsIn() function.")]
public Long GroupId;
```

**Visual Basic**

```
<ObsoleteAttribute("The GroupId property is no longer valid.  To retrieve a list of groups
this User belongs to use Ektron.Cms.API.User.User.GetGroupsUserIsIn() function.")>
Public Property GroupId() As Long
```

**Returns**

The group ID to which the user belongs.

# UserData.GroupName Property

**The GroupName property is no longer valid. To retrieve a list of groups this User (see page 913) belongs to use Ektron.Cms.API.User.GetAllUserGroups() function.**

Gets or sets the name of the group to which the user belongs. This is for the UserData (see page 1419) object.

**C#**

```
[ObsoleteAttribute("The GroupName property is no longer valid.  To retrieve a list of groups
this User belongs to use Ektron.Cms.API.User.User.GetGroupsUserIsIn() function.")]
public String GroupName;
```

**Visual Basic**

```
<ObsoleteAttribute("The GroupName property is no longer valid.  To retrieve a list of groups
this User belongs to use Ektron.Cms.API.User.User.GetGroupsUserIsIn() function.")>
Public Property GroupName() As String
```

**Returns**

The name of the group to which the user belongs.

# UserData.IsMemberShip Property

Gets or sets a boolean indicating whether the user is a Membership (see page 350) user or not. This is for UserData (see page 1419) object.

**C#**

```
public Boolean IsMemberShip;
```

**Visual Basic**

```
Public Property IsMemberShip() As Boolean
```

**Returns**

A boolean indicating whether the user is a Membership (see page 350) user.

- **True** = user is a Membership (see page 350) user.
- **False** = user is a CMS400.NET user.

**Description**

# UserData.LanguageId Property

Gets or sets the integer ID of the language for the UserData (see page 1419) object.

**C#**

```
public int LanguageId;
```

**Visual Basic**

```
Public Property LanguageId() As Integer
```

**Returns**

The integer ID of the language.

# UserData.LanguageName Property

Gets or set the name of the user's language for the UserData (see page 1419) object.

**C#**

```
public String LanguageName;
```

**Visual Basic**

```
Public Property LanguageName() As String
```

**Returns**

Name of the user's language.

# UserData.LastLoginDate Property

Gets or sets the last date a user logged into CMS400.NET. This is for the UserData (see page 1419) object.

**C#**

```
public String LastLoginDate;
```

**Visual Basic**

```
Public Property LastLoginDate() As String
```

**Returns**

The last login date for a user.

# UserData.LoginAttempts Property

Gets an Int16 representing the number of times a user has attempted to login for the UserData (see page 1419) object. This number resets to 0 (zero) once a user successfully logs in.

**C#**

```
public Int16 LoginAttempts;
```

**Visual Basic**

```
Public Property LoginAttempts() As Int16
```

**Returns**

Int16 representing the number of times a user has attempted to login. This number resets to 0 (zero) once a user successfully logs in.

# UserData.Rank Property

Gets or sets the user's rank for a Threaded Discussion for the UserData (see page 1419) object.

**C#**

```
public UserRank Rank;
```

**Visual Basic**

```
Public Property Rank() As UserRank
```

**Returns**

The user's rank for a Threaded Discussion.

# UserData.Signature Property

Gets or sets the user's signature for the UserData (see page 1419) object, used in the forums and other locations.

**C#**

```
public String Signature;
```

**Visual Basic**

```
Public Property Signature() As String
```

**Returns**

The user's signature used in the forums and other locations.

## UserData.UserPreference Property

Gets or sets a user's preferences as UserPreferenceData for the UserData (see page 1419) object.

**C#**

```
public UserPreferenceData UserPreference;
```

**Visual Basic**

```
Public Property UserPreference() As UserPreferenceData
```

**Returns**

A user's preferences as UserPreferenceData.

# UserRequestData Class

A data object that contains request information for users.

**Class Hierarchy**

**C#**

```
public class UserRequestData : PageRequestData;
```

**Visual Basic**

```
Public Class UserRequestData
Inherits PageRequestData
```

**File**

UserData.vb

# UserRequestData Properties

The properties of the UserRequestData class are listed here.

**Public Properties**

| | Name | Description |
|---|---|---|
| | Group (see page 1425) | ID of User (see page 913) Group to retrieve users from. For example, set Group = 1 to retrieve only users that are members of group ID 1. Leave as -1 to retrieve users from all groups. |

| | | |
|---|---|---|
| | OrderBy (see page 1425) | Sets the column by which the resultset is ordered. Possible values include:<br>• **user_id**<br>• **user_name**<br>• **first_name**<br>• **last_name**<br>• **language**<br>• **email_address1**<br>The default sort is **last_name**. |
| | OrderDirection (see page 1426) | Sets the order by which to sort the OrderBy (see page 1425) column. Possible values are:<br>• **ASC**<br>• **DESC**<br>**ASC** is the default value. |
| | RequiredFlag (see page 1426) | Determines whether or not to retrieve deleted users.<br>• **0** = to retrieve active users.<br>• **1** = to retrieve deleted users. |
| | SearchText (see page 1426) | Possible search rules to apply to results using SQL syntax. For example, last_name = 'Jones'. Possible columns include:<br>• **user_id**<br>• **user_name**<br>• **first_name**<br>• **last_name**<br>• **language**<br>• **email_address1** |
| | Type (see page 1427) | Differentiates request between Cms (see page 7) Users and Membership (see page 350) users.<br>• **0** = to retrieve CMS400.NET users.<br>• **1** = to retrieve Membership (see page 350) users.<br>• **-1** = to retrieve all users. |

# UserRequestData.Group Property

ID of User (see page 913) Group to retrieve users from. For example, set Group = 1 to retrieve only users that are members of group ID 1. Leave as -1 to retrieve users from all groups.

**C#**

```
public Long Group;
```

**Visual Basic**

```
Public Property Group() As Long
```

# UserRequestData.OrderBy Property

Sets the column by which the resultset is ordered. Possible values include:

- **user_id**
- **user_name**
- **first_name**
- **last_name**
- **language**
- **email_address1**

The default sort is **last_name**.

**C#**

```
public String OrderBy;
```

**Visual Basic**

```
Public Property OrderBy() As String
```

# UserRequestData.OrderDirection Property

Sets the order by which to sort the OrderBy (see page 1425) column. Possible values are:

- **ASC**
- **DESC**

**ASC** is the default value.

**C#**

```
public String OrderDirection;
```

**Visual Basic**

```
Public Property OrderDirection() As String
```

# UserRequestData.RequiredFlag Property

Determines whether or not to retrieve deleted users.

- **0** = to retrieve active users.
- **1** = to retrieve deleted users.

**C#**

```
public int RequiredFlag;
```

**Visual Basic**

```
Public Property RequiredFlag() As Integer
```

# UserRequestData.SearchText Property

Possible search rules to apply to results using SQL syntax. For example, last_name = 'Jones'. Possible columns include:

- **user_id**
- **user_name**
- **first_name**
- **last_name**
- **language**

- **email_address1**

**C#**

```
public String SearchText;
```

**Visual Basic**

```
Public Property SearchText() As String
```

# UserRequestData.Type Property

Differentiates request between Cms (see page 7) Users and Membership (see page 350) users.

- **0** = to retrieve CMS400.NET users.
- **1** = to retrieve Membership (see page 350) users.
- **-1** = to retrieve all users.

**C#**

```
public int Type;
```

**Visual Basic**

```
Public Property Type() As Integer
```

# Index


.

.NET Assemblies 572

## A

ActionRecipientInfo class 526
- about ActionRecipientInfo class 526
- ActionRecipientInfo 526
- ActionRecipientInfo properties 526
- DisplayName 526
- Email 527
- FirstName 527
- IsUser 527
- LastName 527
- UserId 527

ActionRequestInfo class 527
- about ActionRequestInfo class 527
- ActionRequestInfo 528
- ActionRequestInfo enumerations 528
- ActionRequestInfo properties 528
- ErrorMessage 529
- Id 529
- Recipients 529
- SenderId 529
- Type 529

ActiveDirectoryUser class 974
- about ActiveDirectoryUser class 974
- ActiveDirectoryUser methods 974
- AddADGroupToCMS 975
- DeSynchUserGroups 975
- DeSynchUsers 975
- GetADMapping 976
- GetADStatus 976
- GetAvailableADGroups 977
- GetAvailableADUsers 977
- GetCMSGroupsToSync 978
- GetCMSRelationshipsToSync 978
- GetCMSUsersToSync 979
- GetDomains 979
- MapCMSUserGroupToAD 980
- MapCMSUserToAD 980
- UpdateADMapping 981
- UpdateADUsersGroups 981
- UpdateUsersInfoFromAD 981

ActiveTopics class 14
- about ActiveTopics class 14
- ActiveTopics methods 14
- ActiveTopics properties 15
- BoardID 15
- CacheInterval 16
- DisplayMode 16
- DynamicForumParameter 17
- DynamicGroupParameter 17
- DynamicThreadParameter 17
- Fill 14
- MaxNumber 17
- ToString 15
- URLPath 18
- Wrap 15

AddressApi class 574
- about AddressApi class 574
- Add 575
- Add method 575
- AddressApi methods 574
- Delete 576
- GetItem 576
- GetList 576
- Update 576
- UpdateOrderAddress 577

AddressData class 1206
- about AddressData class 1206
- AddressData properties 1206
- AddressLine1 1207
- AddressLine2 1207
- City 1207
- Company 1207

Country 1207
Id 1208
Name 1208
Phone 1208
PostalCode 1208
Region 1208
AdjustedBasketItem class 1208
about AdjustedBasketItem class 1208
AdjustedBasketItem 1209
AdjustedBasketItem constructor 1209
AdjustedBasketItem properties 1210
AdjustedTotal 1210
ApplicableTaxRate 1210
AppliedCoupons 1210
HasFreeShipping 1210
TaxesApplied 1211
TotalCouponDiscount 1211
ADUser class 913
about ADUser class 913
AddADGroupToCMS 914
AddADUsersToCMSByUsername 914
ADUser methods 913
DeSynchUserGroups 915
DeSynchUsers 915
GetADMapping 915
GetADStatus 916
GetAvailableADGroups 916
GetAvailableADUsers 916
GetCMSGroupsToSync 917
GetCMSRelationshipsToSync 917
GetCMSUsersToSync 917
GetDomains 918
MapCMSUserGroupToAD 918
MapCMSUserToAD 918
SynchCMSGroupsToAD 919
SynchCMSRelationshipsToAD 919
SynchCMSUsersToAD 919
UpdateADMapping 920
UpdateADUsersGroups 920
UpdateUsersInfoFromAD 920
Analytics class 714
about Analytics class 714
Analytics methods 715
Analytics properties 717
AnalyticsEnabled 717
AppendToIISLog 717
GetSessionRank 715
HardDumpAnalyticsBuffer 715
InMemoryContentHitsData 718
QueryAnalytics 715
RecordsBeforeWrite 718
SoftDumpAnalyticsBuffer 716
TimeBeforeWrite 718
TrackVisit 716, 717
TrackVisit method 716
AnalyticsTracker class 18
about AnalyticsTracker class 18
AnalyticsTracker methods 18
AnalyticsTracker properties 19
DefaultContentID 19
DynamicParameter 20
EnableAnalytics 20
Wrap 19
AppliedCouponData class 1211
about AppliedCouponData class 1211
AppliedCouponData 1212
AppliedCouponData constructor 1211
AppliedCouponData properties 1212
Coupon 1212
CouponDiscount 1212
AppliedCouponDataCollection class 1213
about AppliedCouponDataCollection class 1213
AppliedCouponDataCollection fields 1213
AppliedCouponDataCollection methods 1213
FindByCouponId 1213
ItemAdded 1213
AppliedCouponEventArgs class 1214
about AppliedCouponEventArgs class 1214

AppliedCoupon 1215
AppliedCouponEventArgs 1214
AppliedCouponEventArgs constructor 1214
AppliedCouponEventArgs properties 1215
Asset class 789, 982
about Asset class 789, 982
Asset methods 789, 982
GetAssetMgtConfigInfo 790, 983
GetAssetMgtInfo 790, 983
GetAssetSupertypes 790, 983
GetEditSnippet 790, 984
GetMyAssetMgtID 791, 984
GetViewUrl 791
MakeCopyDMSAsset 791, 984
SetAssetMgtConfigInfo 792, 985
AssetControl class 20
about AssetControl class 20
AssetControl methods 21
AssetControl properties 21
DefaultAssetID 22
DefaultFolderID 22
DynamicParameter 22
Fill 21
IsImage 23
OverrideExtension 23
TaxonomyID 24
ToString 21
UploadType 24

## B

Basket class 1215
about Basket class 1215
AddProduct 1216, 1217
AddProduct method 1216
AddProductVariant 1217
Basket methods 1215
Basket properties 1219
Decrement 1217
GetItemIndex 1217
GetKitConfiguration 1218
GetKitConfigurationAsAdmin 1218
IncrementItemQuantity 1218
MakeDefaultBasket 1218
RemoveItem 1218
RequestInformation 1220
SaveBasket 1219
UpdateItemQuantity 1219
UpdateKitConfiguration 1219
BasketApi class 577
about BasketApi class 577
BasketApi methods 577
CalculateBasket 578, 579
CalculateBasket method 578
CreateNewBasket 579
Delete 580
Empty 580
GetDefaultBasket 581
GetDefaultBasket method 580
GetItem 581
GetList 582
SetDefaultBasket 582, 583
SetDefaultBasket method 582
BasketCalculatorData class 1220
about BasketCalculatorData class 1220
AdjustedItems 1221
Basket 1221
BasketCalculatorData 1220, 1221
BasketCalculatorData constructor 1220
BasketCalculatorData properties 1221
BasketCoupons 1222
BasketTotal 1222
HasFreeShipping 1222
ShipFrom 1222
ShippingRates 1222
ShipTo 1222
TaxClassSubtotals 1223
TotalCouponDiscount 1223
TotalShippingCost 1223

TotalTaxes 1223
ValidationResults 1223
BasketData class 1224
about BasketData class 1224
BasketData properties 1224
DateCreated 1224
DateModified 1224
Id 1225
IsDefault 1225
Items 1225
Name 1225
ShippingAddressId 1225
ShippingMethodId 1226
ShopperId 1226
Subtotal 1226
UserId 1226
BasketItem class 1226
about BasketItem class 1226
AdjustedTotal 1228
BasketItem 1227
BasketItem constructor 1227
BasketItem properties 1227
Configuration 1228
Dimensions 1228
Id 1228
IsTangible 1228
ListPrice 1229
Pricing 1229
ProductId 1229
ProductLanguage 1229
ProductSku 1229
ProductTitle 1230
ProductType 1230
Quantity 1230
Quicklink 1230
SalePrice 1230
TaxClassId 1231
Variant 1231
Weight 1231
BasketItemCollection class 1231
about BasketItemCollection class 1231
AddRange 1232
BasketItemCollection 1231
BasketItemCollection methods 1232
Find 1232
FindAll 1232
FindIndex 1232
BasketVariantData class 1233
about BasketVariantData class 1233
BasketVariantData 1233
BasketVariantData properties 1233
Dimensions 1233
Id 1234
Language 1234
ListPrice 1234
Pricing 1234
SalePrice 1235
Sku 1235
TaxClassId 1235
Title 1235
Weight 1235
Blog class 24, 792, 985
about Blog class 24, 792, 985
AddBlog 793, 986
AddCommentForPost 794, 987
AddPost 794, 987
ArchiveMode 26
Blog methods 25, 792, 986
Blog properties 26
BlogID 27
CacheInterval 27
DateToStart 28
DefaultUserID 28
DeleteBlog 796, 988
DeleteBlogPostComment 796, 988
DeletePost 796, 988
DynamicParameter 28
DynamicUserParameter 29

EditorHeight 29
EditorWidth 29
Fill 25
GetBlankBlogPostData 796, 989
GetBlog 797, 800, 990, 991
GetBlog method 797, 989
GetBlogData 801, 802, 803, 992, 993, 994
GetBlogData method 800, 991
GetBlogDataForPost 803, 995
GetBlogPostData 803, 995
GetBlogRoll 804, 805, 996
GetBlogRoll method 804, 995
GetPost 805, 997
GetPostbyID 805, 997
GetUserBlog 806, 998
JavascriptEditorHTMLMode 30
MaxResults 30
PostParameter 30
RecentPosts 31
ShowHeader 31
ShowRSS 31
ToString 25
UpdateBlogProperties 806, 998
Wrap 25
BlogArchive class 32
about BlogArchive class 32
ArchiveMode 33
BlogArchive methods 32
BlogArchive properties 33
BlogID 33
CacheInterval 33
DynamicParameter 34
Fill 32
Text 34
Wrap 32
WrapTag 34
BlogCalendar class 35
about BlogCalendar class 35
BlogCalendar methods 35
BlogCalendar properties 36
BlogID 36
CacheInterval 36
Fill 35
Text 36
Wrap 35
WrapTag 37
BlogCategories class 37
about BlogCategories class 37
BlogCategories methods 37
BlogCategories properties 38
BlogID 38
CacheInterval 39
Fill 37
Text 39
Wrap 38
WrapTag 39
BlogEntries class 39
about BlogEntries class 39
BlogEntries methods 40
BlogEntries properties 42
BlogID 42
BlogPostParameter 43
CacheInterval 43
DateToStart 43
DefaultUserID 44
DisplayXslt 44
DynamicParameter 44
DynamicUserParameter 45
EditorHeight 45
EditorWidth 45
Fill 40
GetRssFeed 40
GetView 41
GetViewNames 41
JavascriptEditorHTMLMode 46
MaxResults 46
ShowHeader 46
ToString 41

Wrap 41
XmlDoc 47
BlogPost class 47
about BlogPost class 47
BlogPost methods 47
BlogPost properties 48
DefaultContentID 48
ShowType 49
Wrap 48
BlogRecentPosts class 49
about BlogRecentPosts class 49
BlogID 50
BlogRecentPosts methods 49
BlogRecentPosts properties 50
CacheInterval 51
Fill 50
NumberofPosts 51
Text 51
Wrap 50
WrapTag 51
BlogRoll class 52
about BlogRoll class 52
BlogID 53
BlogRoll methods 52
BlogRoll properties 53
CacheInterval 53
Fill 52
Text 54
Wrap 52
WrapTag 54
BlogRSS class 54
about BlogRSS class 54
BlogID 55
BlogRSS methods 54
BlogRSS properties 55
CacheInterval 55
Fill 55
Text 56
WrapTag 56
BreadCrumb class 56
about BreadCrumb class 56
BreadCrumb methods 57
BreadCrumb properties 58
CurrentPageIndicator 59
DefaultContentID 59
DisplayStyle 60
DisplayTitle 60
DynamicParameter 60
Fill 57
GenerateTrail_HTML 57
IconAlt 60
IconPath 61
LinkLastItem 61
LinkTarget 61
MaxItems 61
MetadataName 62
Mode 62
ResetTrail 58
ResetTrail method 57
separator 62
BreadCrumbItem class 63
about BreadCrumbItem class 63
BreadCrumbItem 63
BreadCrumbItem constructor 63
BreadCrumbItem methods 64
BreadCrumbItem properties 65
CaseSensitiveUrlComparison 65
DisplayTitle 65
Equals 64
GetHashCode 64
IconAlt 65
IconPath 65
RawUrl 66
Target 66
ToString 64
BundleData class 1236
about BundleData class 1236
BundleData 1236

BundleData properties 1236
BundledItems 1236
BusinessRule class 640
about BusinessRule class 640
BusinessRule methods 641
GetBusinessRules 641, 642
GetBusinessRules method 641
BusinessRules class 66, 999
about BusinessRules class 66, 999
BusinessRules methods 67, 999
BusinessRules properties 67
Fill 67
GetBusinessRules 1000
GetBusinessRules method 999
Parameters 67
Result 68
RuleID 68
Text 68
Wrap 67
WrapTag 69


## C


Calendar class 69
about Calendar class 69
Calendar methods 69
Calendar properties 70
DefaultCalendarID 70
DisplayType 71
DynamicParameter 71
EndDate 72
Fill 69
HighlightEventID 72
HighlightEventTypeID 72
ShowXML 72
StartDate 73
Text 73
Wrap 70
Cart class 73
about Cart class 73
Cart properties 74
DisplayXslt 74
DynamicProductParameter 75
EnableCoupons 75
LoadingImage 76
Stylesheet 76
TemplateCheckout 76
TemplateProduct 77
TemplateShopping 77
UserControlUrl 77
CatalogData class 1236
about CatalogData class 1236
CatalogData properties 1237
Id 1237
Name 1237
CatalogEntryApi class 583
about CatalogEntryApi class 583
Add 584
AddAndCheckIn 584
AddAndPublish 584
AddAndSave 584
CatalogEntryApi methods 583
Checkout 585
Delete 585
GetAttributeEntryList 585
GetItem 586
GetItem method 585
GetItemEdit 586
GetList 587
GetTaxonomyEntryList 587
Restore 587
UndoCheckOut 587
Update 588
UpdateAndCheckIn 588
UpdateAndPublish 588
Checkout class 78
about Checkout class 78
Cart 80
Checkout delegates 87

Checkout enumerations 78
Checkout events 87
Checkout methods 79
Checkout properties 79
CurrentPhase 81
Customer 81
DefaultCountryId 81
DisplayXslt 82
DynamicPhaseParameter 82
Fill 79
FriendlyErrorMessage 82
IsSSLRequired 83
LoadingImage 83
PaymentMethod 83
PhaseChange 88
PhaseChangeEventHandler 87
PostLogin 88
PostLoginEventHandler 87
ShowErrorDetails 84
Stylesheet 84
TemplateCart 84
TemplateOrderHistory 85
TemplateRecoverPassword 85
TemplateShopping 86
ToString 79
UseMsAjax 86
ValidatePhase 86
Wrap 79
Collection class 88
about Collection class 88
AddText 92
CacheInterval 93
Collection methods 89
Collection properties 91
CollectionDescription 93
CollectionTitle 93
ContentParameter 93
CopyTo 89
Count 94
Current 94
DefaultCollectionID 94
DynamicParameter 94
EkItems 95
EnablePaging 95
Fill 89
GetEnumerator 89
GetHtml 95
GetRssFeed 90
GetView 90
GetViewNames 90
IncludeIcons 96
IsSynchronized 96
LinkTarget 96
MarkupLanguage 97
MaxResults 97
MoveNext 90
Random 97
Reset 91
SuppressWrapperTags 97
SyncRoot 98
this 91
ToString 91
Wrap 91
XmlDoc 98
CommunityDocuments class 98
about CommunityDocuments class 98
AddItemFolderId 99
CommunityDocuments methods 99
CommunityDocuments properties 99
DefaultObjectID 100
DynamicContentBoxHeight 100
DynamicContentBoxWidth 100
DynamicContentTemplate 101
EnableAjax 101
EnableSearch 101
Fill 99
MemberMenuActive 101
ObjectType 102

Recursive 102
ShowAllChecked 102
Stylesheet 102
SuppressWrapperTags 103
TaxonomyDepth 103
TaxonomyId 103
CommunityGroup class 719, 1001
about CommunityGroup class 719, 1001
AcceptGroupInvite 721
AddCommunityGroup 721, 1002
AddUserToCommunityGroup 722, 1003
AddUserToCommunityGroup method 721, 1002
ApprovePendingGroupUser 722, 1004
CancelJoinRequestForCommunityGroup 722, 1004
CommunityGroup methods 720, 1001
DeleteCommunityGroupByID 723, 1004
DeletePendingGroupUser 723, 1005
GetAllCommunityGroups 723, 1005
GetAllCommunityGroupsInDirectory 724, 1006
GetAllUnassignedCommunityGroupUsers 724, 1006
GetCommunityGroupBlog 724
GetCommunityGroupBlogIds 724, 1006
GetCommunityGroupByID 725, 1007
GetCommunityGroupDiscussionBoardIds 725, 1007
GetCommunityGroups 725, 1008
GetCommunityGroupUserIDs 726, 1008
GetCommunityGroupUsers 726, 727, 1009
GetCommunityGroupUsers method 726, 1009
GetFolderIdForCommunityGroup 727, 1010
GetGroupMemberStatus 727, 1010
GetInvitedCommunityGroups 728, 1011
GetMostPopularCommunityGroups 728, 729, 1012
GetMostPopularCommunityGroups method 728, 1011
GetPendingCommunityGroups 729, 1013
GetPendingCommunityGroupUsers 729, 1013
InviteUsers 730, 1014
IsCommunityGroupAdmin 730, 1014
IsUserInCommunityGroup 730, 1014
RemoveUserFromCommunityGroup 731, 1015
UpdateCommunityGroup 731, 1015
CommunityGroupBrowser class 104
about CommunityGroupBrowser class 104
AvatarHeight 105
AvatarWidth 105
CacheInterval 106
CommunityGroupBrowser methods 104
CommunityGroupBrowser properties 105
DisplayXslt 106
Fill 104
Link 107
LinkTarget 107
MarkupLanguage 107
MaxResults 107
Stylesheet 108
TagTemplate 108
TagTemplateTarget 108
TaxonomyId 108
ToString 104
Wrap 104
XmlDoc 109
CommunityGroupList class 109
about CommunityGroupList class 109
AllowGroupCreation 111
AvatarHeight 111
AvatarWidth 111
CacheInterval 111
CommunityGroupList methods 109
CommunityGroupList properties 110
DefaultUserID 112
DisplayMode 112
DisplayXslt 112
DynamicUserParameter 113
Fill 110
LanguageId 113
Link 114
LinkTarget 114
MarkupLanguage 114
MaxResults 114

ShowMySortingOptions 115
ShowSortingOptions 115
Stylesheet 115
TagTemplate 115
TagTemplateTarget 116
ToString 110
Wrap 110
XmlDoc 116
CommunityGroupMembers class 116
about CommunityGroupMembers class 116
AvatarHeight 118
AvatarWidth 118
CacheInterval 118
CommunityGroupMembers methods 117
CommunityGroupMembers properties 117
DefaultCommunityGroupID 119
DisplayMode 119
DisplayXslt 119
DynamicParameter 120
Fill 117
Link 120
LinkTarget 120
MarkupLanguage 121
MaxResults 121
Stylesheet 121
ToString 117
Wrap 117
XmlDoc 122
CommunityGroupProfile class 122
about CommunityGroupProfile class 122
CommunityGroupProfile methods 122
CommunityGroupProfile properties 123
DefaultGroupID 123
DisplayXslt 124
DynamicParameter 124
EkItem 124
Fill 122
GroupAdminLink 125
GroupAdminLinkTarget 125
Stylesheet 125
TagTemplate 125
TagTemplateTarget 126
Text 126
ToString 122
Wrap 123
XmlDoc 126
CommunitySearch class 127
about CommunitySearch class 127
CacheInterval 129
CommunitySearch methods 127
CommunitySearch properties 128
CustomPropertyID 130
DisplayXslt 130
EnableFilteringResults 131
EnableGroupResults 131
EnableMap 131
EnableMouseOutHidesInfo 132
EnableMouseOver 132
EnableUnauthorizedSearching 132
EnableUserResults 133
Fill 127
FriendsOnly 133
GroupTaxonomyID 133
InfoBoxShowsEmail 134
LoadingImage 134
MapControl 134
MarkupLanguage 135
MaxTagCount 135
MembersOnly 135
PageSize 136
ResultProfile 136
SearchPrivate 136
ShowNameEmail 137
StartingTab 137
Stylesheet 138
TemplateGroupProfile 138
TemplateParamName 138
TemplateTarget 139

TemplateUserProfile 139
ToString 127
UseMsAjax 139
UserTaxonomyID 140
Wrap 128
XmlDoc 140
XsltDoc 140
CommunityUserData class 1377
about CommunityUserData class 1377
CommunityUserData properties 1377
Tags 1378
Content class 806, 1016
about Content class 806, 1016
AddAsset 808
AddContent 809, 811, 812, 1017, 1018, 1019
AddContent method 808, 1017
AddXmlConfiguration 813
CheckOutContent 813, 1019
Content methods 806, 1016
CopyContentToFolder 814, 1020
DeleteContentItem 814, 1020
DeleteXmlConfiguration 815
EditAsset 816
GetAddViewLanguage 816, 1021
GetAllComments 816, 1021
GetAllTemplates 817, 1021
GetAllUnassignedItemApprovals 817, 1022
GetAllXmlConfigurations 817
GetChildContent 818, 819, 1022, 1023
GetChildContent method 818, 1022
GetCommunityTemplate 820
GetContent 821, 822, 1024
GetContent method 820
GetContentByHistory 823, 1024
GetContentDisplayLink 823
GetContentForEditing 823, 1025
GetContentReport 824
GetContentReport method 824
GetContentState 825, 1025
GetContentStatus 825, 1025
GetDomain 825, 1026
GetExpireContent 826
GetList 826
GetList method 826
MoveContentToFolder 827, 1026
PublishContent 827, 1026
RelicateXmlConfiguration 828
RenameContent 828, 1027
ReplicateXmlConfiguration 828
RestoreHistoryContent 829
SaveContent 829, 1027
SubmitForDelete 830, 1028
UndoCheckout 830, 1028
UpdateContentMetaData 830, 1029
UpdatexmlCollectionPackage 831
UpdateXmlConfiguration 831
ContentBlock class 140
about ContentBlock class 140
CacheInterval 143
ContentBlock methods 141
ContentBlock properties 143
CopyTo 141
Count 144
Current 144
DefaultContentID 144
DisplayXslt 145
DynamicParameter 145
EkItem 145
Fill 142
GetEnumerator 142
IsSynchronized 146
MoveNext 142
OverrideXslt 146
Reset 142
StartingFolderID 146
SyncRoot 147
this 142
ToString 142

Wrap 143
ContentData class 1378
about ContentData class 1378
ApprovalMethod 1380
Approver 1381
Comment 1381
ContentData properties 1378
ContType 1381
DateCreated 1382
DisplayDateCreated 1382
DisplayEndDate 1382
DisplayGoLive 1383
DisplayLastEditDate 1383
EditorFirstName 1383
EditorLastName 1383
EditorUserNames 1384
EndDate 1384
EndDateAction 1384
FolderId 1385
FolderName 1385
GoLive 1385
HistoryId 1385
HyperLink 1386
InheritedFrom 1386
IsInherited 1386
IsPublished 1386
IsSearchable 1387
IsXmlInherited 1387
LastEditDate 1387
ManualAlias 1387
ManualAliasId 1388
MetaData 1388
Path 1388
Permissions 1389
Status 1389
StyleSheet 1389
TemplateConfiguration 1390
UserId 1390
UserName 1390
XmlConfiguration 1390
XmlInheritedFrom 1391
ContentFlagging class 147
about ContentFlagging class 147
ContentFlagging methods 147
ContentFlagging properties 148
DefaultContentID 148
DisplayXslt 149
DynamicParameter 149
Fill 147
FlagImage 149
FlagText 150
Stylesheet 150
ToString 147
Wrap 148
XmlDoc 151
ContentInfo class 530
about ContentInfo class 530
AssetID 530
Comments 531
ContentInfo properties 530
ErrorMessage 531
FileAndPath 531
FileExtension 531
FileName 531
FolderID 531
Html 532
ID 532
Language 532
MimeType 532
Status 532
Title 532
Type 533
UserID 533
ContentList class 151
about ContentList class 151
CacheInterval 155
ContentIds 155
ContentList methods 151

ContentList properties 154
CopyTo 152
Count 156
Current 156
DefaultContentID 156
Direction 156
DynamicParameter 157
EkItems 157
Fill 152
GetEnumerator 152
GetHtml 157
GetRssFeed 152
IncludeIcons 158
IsSynchronized 158
LinkTarget 158
MarkupLanguage 159
MetaTag 159
MoveNext 152
OrderByDirection 159
OrderKey 160
Random 160
Reset 153
SuppressWrapperTags 160
SyncRoot 161
this 153
ToString 153
Wrap 153
XmlDoc 161
ContentRating class 161, 831, 1029
about ContentRating class 161, 831, 1029
AddContentRating 1030
AlreadyRatedMessage 163
BadLabel 163
CharsRemainingLabel 164
CompleteGraphDisplay 164
ContentRating methods 161, 831, 1029
ContentRating properties 162
CookiesRequiredMessage 164
DefaultContentID 165
DynamicParameter 165
FeedbackHeader 165
Fill 162
GetContentRating 832
GetContentRatingAddRating 834
GetContentRatingResults 834, 1030
GetContentRatingStatistics 835
GetContentRatingStatistics method 834
GoodLabel 166
GraphBarColor 166
InitialGraphDisplay 166
Padding 167
PurgeContentRatings 836, 1030
RatingLevelLabel 167
ThankYouMessage 168
ToString 162
TotalRatingsLabel 168
UserCommentsHeader 168
ValidationMessage 169
Wrap 162
ContentRatingLabels class 1391
about ContentRatingLabels class 1391
AlreadyRatedMessage 1392
BadLabel 1392
CharsRemainingLabel 1392
ContentRatingLabels properties 1391
CookiesRequiredMessage 1392
FeedbackHeader 1393
GoodLabel 1393
RatingLevelLabel 1393
ThankYouMessage 1393
TotalRatingsLabel 1393
UserCommentsHeader 1394
ValidationMessage 1394
ContentReview class 169
about ContentReview class 169
ContentReview methods 169
ContentReview properties 171
DefaultContentID 171

DisplayXslt 172
DynamicParameter 172
DynamicUserParameter 173
Fill 170
GetReviews 173
MaxReviews 173
Moderate 174
RaiseCallbackEvent 170
RatingsMinimum 174
ToString 170
UserID 174
Wrap 170
XmlDoc 175
ContractApi class 588
about ContractApi class 588
Add 589
ContractApi methods 589
Delete 589
GetItem 589
GetList 590
Update 590
ContractData class 1237
about ContractData class 1237
ContractData 1238
ContractData constructor 1238
ContractData properties 1238
Id 1238
Name 1239
CountryApi class 590
about CountryApi class 590
Add 591
CanDelete 591
CountryApi methods 590
Delete 591
GetItem 592
GetList 592
Update 592
CountryData class 1239
about CountryData class 1239
CountryData 1239, 1240
CountryData constructor 1239
CountryData methods 1240
CountryData properties 1241
Enabled 1241
Id 1241
LongIsoCode 1241
Name 1241
ShortIsoCode 1241
ToString 1240
Validate 1240
CountryTaxRateData class 1242
about CountryTaxRateData class 1242
CountryData 1243
CountryTaxRateData 1242, 1243
CountryTaxRateData constructor 1242
CountryTaxRateData properties 1243
CouponApi class 592
about CouponApi class 592
Add 593
AddCouponToCatalog 594
AddCouponToProduct 594
AddCouponToTaxonomy 594
CalculateBasketCoupons 595
CouponApi methods 593
Deactivate 595
Delete 595
DeleteCouponProducts 595
GetCatalogList 596
GetCouponsAppliedToBasket 596
GetItem 596
GetList 596
GetProductList 597
GetProductList method 597
GetTaxonomyList 597
IsCouponAppliedToProduct 598
IsCouponUsedForBasket 598
IsCouponUsedForOrder 598
SaveCouponApplications 598

Update 599
ValidateCoupon 599
ValidateCoupon method 599
CouponData class 1243
about CouponData class 1243
Code 1244
CouponData 1244
CouponData properties 1244
CouponType 1245
CurrencyId 1245
Description 1245
DiscountType 1245
DiscountValue 1245
ExpirationDate 1245
Id 1246
IsActive 1246
IsCombinable 1246
MaximumAmount 1246
MaximumUses 1246
MinimumAmount 1247
OnePerCustomer 1247
StartDate 1247
CouponEntryData class 1247
about CouponEntryData class 1247
CouponEntryData 1248
CouponEntryData constructor 1247
CouponEntryData properties 1248
CouponId 1248
DataState 1249
IsIncluded 1249
ObjectId 1249
ObjectType 1249
Path 1249
Title 1250
CurrencyApi class 600
about CurrencyApi class 600
Add 600
CanDelete 601
CurrencyApi methods 600
Delete 601
GetItem 601
GetList 601
Update 602
CurrencyData class 1250
about CurrencyData class 1250
AlphaIsoCode 1250
CultureCode 1251
CurrencyData properties 1250
CurrencySymbol 1251
Enabled 1251
Id 1251
ISOCurrencySymbol 1251
Name 1252
CurrencyPricingData class 1252
about CurrencyPricingData class 1252
AlphaIsoCode 1254
Clone 1253
CurrencyId 1254
CurrencyPricingData 1252, 1253
CurrencyPricingData constructor 1252
CurrencyPricingData methods 1253
CurrencyPricingData properties 1254
GetSalePrice 1254
ListPrice 1255
NumericIsoCode 1255
PricingType 1255
SalePrice 1255
TierPrices 1255
Validate 1254
CurrencySelect class 175
about CurrencySelect class 175
CacheInterval 176
CurrencySelect methods 175
CurrencySelect properties 176
DisplayXslt 177
EkItem 178
Fill 176
Stylesheet 178

ToString 176
Wrap 176
CustomerApi class 602
about CustomerApi class 602
Add 603
Add method 603
CustomerApi methods 602
Delete 603
GetItem 604
GetList 604
GetOrderList 604
GetOrderTotal 605
Update 605
CustomerData class 1255
about CustomerData class 1255
AverageOrderValue 1256
BillingAddressId 1256
CustomerData properties 1256
DisplayName 1257
EmailAddress 1257
FirstName 1257
Id 1257
IsDeleted 1257
IsMembershipUser 1257
LastName 1258
Password 1258
ShippingAddressId 1258
TotalOrders 1258
TotalOrderValue 1258
UserId 1258
UserName 1259
CustomFields class 642, 1031
about CustomFields class 642, 1031
AddNewEditableCustomFieldAssignments 644, 1032
BoolToYesNo 644, 1033
CustomFields methods 643, 1031
GetEditableCustomFieldAssignments 644, 1033
GetEnhancedMetadataArea 645, 1034
GetEnhancedMetadataScript 645, 1034
GetFieldsByFolder 645, 1035
GetSearchProperties 646, 647, 1035
GetSearchProperties method 646
Populate_AssetRequestObjectFromForm 647
ProcessCustomFields 648, 1035
SearchAllAssets 648, 1036
WriteFilteredMetadataForEdit 649, 1036
WriteFilteredMetadataForView 649, 1037
WriteLibrarySearchExtended 649, 1037
WriteMetadataDefaultForEdit 650, 1037
WriteMetadataTypeForView 650, 1038
WritePublicSearch 650, 1038
WritePublicSearchEx 651

## D

Data Classes 1201
DefaultCatalogIDUITypeEditor class 178
about DefaultCatalogIDUITypeEditor class 178
DesignTimeDiagnostic class 178
about DesignTimeDiagnostic class 178
Authenticated 180
DesignTimeDiagnostic methods 179
DesignTimeDiagnostic properties 179
Fill 179
Hide 180
Language 180
SuppressWrapperTags 180
WrapTag 180
DhtmlMenu class 181
about DhtmlMenu class 181
AccessKey 183
Align 183
BackColor 183
BorderColor 183
BorderStyle 184
BorderWidth 184
CssClass 184
DhtmlMenu methods 181
DhtmlMenu properties 182

DisplayXslt 184
Enabled 184
EnableIE6FormsFix 185
Fill 181
Height 185
HorizontalDisplay 185
MenuHoverOverColor 186
MenuStyle 186
Stylesheet 186
SubMenuBackColor 186
SubMenuHoverOverColor 187
SubMenuItemHeight 187
SubMenuItemWidth 187
SubMenuTopOffset 188
TabIndex 188
ToolTip 188
Width 188
Wrap 181
Dimensions class 1259
about Dimensions class 1259
Clone 1260
CompareTo 1260
ConvertUnits 1261
CubicUnits 1261
Dimensions 1259, 1260
Dimensions constructor 1259
Dimensions methods 1260
Dimensions properties 1261
Height 1261
Length 1261
Units 1262
Width 1262
Directory class 189
about Directory class 189
AddItemFolderId 193
CacheInterval 194
CopyTo 190
Count 194
Current 194
Directory methods 189
Directory properties 192
DisplayXslt 194
DynamicParameter 195
EkItems 195
EnableAjax 195
EnablePaging 196
EnableSearch 196
Fill 190
GetEnumerator 190
GetView 190
GetViewNames 190
IncludeIcons 196
IsSynchronized 197
ItemSortOrder 197
LinkTarget 197
MarkupLanguage 198
MaxResults 198
MoveNext 191
Recursive 198
Reset 191
ShowAllChecked 198
ShowCount 199
SortDirection 199
SuppressWrapperTags 199
SyncRoot 200
TaxonomyCols 200
TaxonomyDepth 200
TaxonomyDescription 200
TaxonomyId 201
TaxonomyItemCols 201
TaxonomyPath 201
TaxonomyTitle 201
TaxonomyTreeData 202
TaxonomyXml 202
this 191
ToString 191
Wrap 192
XmlDoc 202

## E

eCommerce 573

Ektron namespace

Classes 573

Ektron.Cms 7

Ektron.Cms namespace 7

Ektron.Cms.API 639

Ektron.Cms.API namespace 639

Classes 639

Ektron.Cms.API.Analytics 714

Ektron.Cms.API.Analytics namespace 714

Classes 714

Ektron.Cms.API.Analytics.Analytics 714

Ektron.Cms.API.Analytics.Analytics.AnalyticsEnabled 717

Ektron.Cms.API.Analytics.Analytics.AppendToIISLog 717

Ektron.Cms.API.Analytics.Analytics.GetSessionRank 715

Ektron.Cms.API.Analytics.Analytics.HardDumpAnalyticsBuffer 715

Ektron.Cms.API.Analytics.Analytics.InMemoryContentHitsData 718

Ektron.Cms.API.Analytics.Analytics.QueryAnalytics 715

Ektron.Cms.API.Analytics.Analytics.RecordsBeforeWrite 718

Ektron.Cms.API.Analytics.Analytics.SoftDumpAnalyticsBuffer 716

Ektron.Cms.API.Analytics.Analytics.TimeBeforeWrite 718

Ektron.Cms.API.Analytics.Analytics.TrackVisit 716, 717

Ektron.Cms.API.BusinessRule 640

Ektron.Cms.API.BusinessRule.GetBusinessRules 641, 642

Ektron.Cms.API.Community 718

Ektron.Cms.API.Community namespace 718

Classes 719

Ektron.Cms.API.Community.CommunityGroup 719

Ektron.Cms.API.Community.CommunityGroup.AcceptGroupInvite 721

Ektron.Cms.API.Community.CommunityGroup.AddCommunityGroup 721

Ektron.Cms.API.Community.CommunityGroup.AddUserToCommunityGroup 722

Ektron.Cms.API.Community.CommunityGroup.ApprovePendingGroupUser 722

Ektron.Cms.API.Community.CommunityGroup.CancelJoinRequestForCommunityGroup 722

Ektron.Cms.API.Community.CommunityGroup.DeleteCommunityGroupByID 723

Ektron.Cms.API.Community.CommunityGroup.DeletePendingGroupUser 723

Ektron.Cms.API.Community.CommunityGroup.GetAllCommunityGroups 723

Ektron.Cms.API.Community.CommunityGroup.GetAllCommunityGroupsInDirectory 724

Ektron.Cms.API.Community.CommunityGroup.GetAllUnassignedCommunityGroupUsers 724

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupBlog 724

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupBlogIds 724

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupByID 725

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupDiscussionBoardIds 725

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroups 725

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupUserIDs 726

Ektron.Cms.API.Community.CommunityGroup.GetCommunityGroupUsers 726, 727

Ektron.Cms.API.Community.CommunityGroup.GetFolderIdForCommunityGroup 727

Ektron.Cms.API.Community.CommunityGroup.GetGroupMemberStatus 727

Ektron.Cms.API.Community.CommunityGroup.GetInvitedCommunityGroups 728

Ektron.Cms.API.Community.CommunityGroup.GetMostPopularCommunityGroups 728, 729

Ektron.Cms.API.Community.CommunityGroup.GetPendingCommunityGroups 729

Ektron.Cms.API.Community.CommunityGroup.GetPendingCommunityGroupUsers 729

Ektron.Cms.API.Community.CommunityGroup.InviteUsers 730

Ektron.Cms.API.Community.CommunityGroup.IsCommunityGroupAdmin 730

Ektron.Cms.API.Community.CommunityGroup.IsUserInCommunityGroup 730

Ektron.Cms.API.Community.CommunityGroup.RemoveUserFromCommunityGroup 731

Ektron.Cms.API.Community.CommunityGroup.UpdateCommunityGroup 731

Ektron.Cms.API.Community.Favorites 731

Ektron.Cms.API.Community.Favorites.AddContentFavorite 732, 733

Ektron.Cms.API.Community.Favorites.AddFavoriteLink 733

Ektron.Cms.API.Community.Favorites.DeleteContentFavorite 733

Ektron.Cms.API.Community.Favorites.DeleteFavoriteLink 734

Ektron.Cms.API.Community.Favorites.GetAllFavoriteLink 734, 735

Ektron.Cms.API.Community.Favorites.GetContentFavorites 736, 737

Ektron.Cms.API.Community.Favorites.GetFavoriteId 737

Ektron.Cms.API.Community.Favorites.GetFavoriteLink 738

Ektron.Cms.API.Community.Favorites.GetFavoritesDirectory 738

Ektron.Cms.API.Community.Favorites.IsContentFavorite 738

Ektron.Cms.API.Community.Favorites.MoveContentFavorite 739

Ektron.Cms.API.Community.Favorites.MoveFavoriteLink 739

Ektron.Cms.API.Community.Favorites.UpdateFavoriteLink 739

Ektron.Cms.API.Community.Flagging 739

Ektron.Cms.API.Community.Flagging.AddFlagObject 740

Ektron.Cms.API.Community.Flagging.AssignFlagToContent 741

Ektron.Cms.API.Community.Flagging.GetAllFlagEntries 741, 742

Ektron.Cms.API.Community.Flagging.GetContentFlagData 743

Ektron.Cms.API.Community.Flagging.GetDefaultFolderFlagDef 743

Ektron.Cms.API.Community.Flagging.GetFlagDataById 744

Ektron.Cms.API.Community.Flagging.GetFlaggedRank 744

Ektron.Cms.API.Community.Flagging.PurgeFlagEntries 744

Ektron.Cms.API.Community.Flagging.RemoveObjectFlag 744

Ektron.Cms.API.Community.Flagging.UpdateFlagObject 745

Ektron.Cms.API.Community.Friends 745

Ektron.Cms.API.Community.Friends.AcceptFriendInvite 746

Ektron.Cms.API.Community.Friends.AcceptPendingFriend 746

Ektron.Cms.API.Community.Friends.AddFriend 747

Ektron.Cms.API.Community.Friends.AddPendingFriend 747

Ektron.Cms.API.Community.Friends.AddSelectedFriend 748

Ektron.Cms.API.Community.Friends.DeleteFriend 748

Ektron.Cms.API.Community.Friends.DeletePendingFriendRequest 749

Ektron.Cms.API.Community.Friends.DeleteSelectedFriend 749

Ektron.Cms.API.Community.Friends.GetFriends 750, 751

Ektron.Cms.API.Community.Friends.GetFriendsDirectory 751

Ektron.Cms.API.Community.Friends.GetFriendStatus 752

Ektron.Cms.API.Community.Friends.GetInvitedUsers 752

Ektron.Cms.API.Community.Friends.GetPendingFriends 752

Ektron.Cms.API.Community.Friends.GetSelectedFriends 753

Ektron.Cms.API.Community.Friends.GetSentFriendInvitations 753

Ektron.Cms.API.Community.Friends.InviteUsers 753

Ektron.Cms.API.Community.Friends.IsFriend 754

Ektron.Cms.API.Community.Friends.IsSelectedFriend 754

Ektron.Cms.API.Community.Friends.MoveFriend 754

Ektron.Cms.API.Community.MessageBoard 755

Ektron.Cms.API.Community.MessageBoard.AddMessageBoardEntry 755

Ektron.Cms.API.Community.MessageBoard.ApproveMessage 756, 757

Ektron.Cms.API.Community.MessageBoard.DeleteMessageBoardEntries 758, 759

Ektron.Cms.API.Community.MessageBoard.DeleteMessageBoardEntriesByUser 760, 761

Ektron.Cms.API.Community.MessageBoard.DeleteMessageBoardEntry 762
Ektron.Cms.API.Community.MessageBoard.GetAllMessageBoardEntries 764
Ektron.Cms.API.Community.MessageBoard.GetAllMessageBoardEntriesByUser 765, 766
Ektron.Cms.API.Community.MessageBoard.GetMessageBoardEntry 767, 768
Ektron.Cms.API.Community.MessageBoard.IsUserMessageBoardAdmin 769
Ektron.Cms.API.Community.MessageBoard.UpdateMessageText 770
Ektron.Cms.API.Community.Tags 771
Ektron.Cms.API.Community.Tags.AddTag 772
Ektron.Cms.API.Community.Tags.AddTagToCommunityGroup 773, 774
Ektron.Cms.API.Community.Tags.AddTagToContent 775
Ektron.Cms.API.Community.Tags.AddTagToObject 776, 777
Ektron.Cms.API.Community.Tags.AddTagToUser 778
Ektron.Cms.API.Community.Tags.DeleteTagById 779
Ektron.Cms.API.Community.Tags.DeleteTagOnObject 779
Ektron.Cms.API.Community.Tags.DeleteTagsByUser 779
Ektron.Cms.API.Community.Tags.DeleteTagsOnCommunityGroup 780
Ektron.Cms.API.Community.Tags.DeleteTagsOnContent 780
Ektron.Cms.API.Community.Tags.DeleteTagsOnObject 780
Ektron.Cms.API.Community.Tags.DeleteTagsOnUser 781
Ektron.Cms.API.Community.Tags.EditTag 781
Ektron.Cms.API.Community.Tags.GetAllTagRelatedContent 781
Ektron.Cms.API.Community.Tags.GetAllTagRelatedUsers 781
Ektron.Cms.API.Community.Tags.GetAllTags 782, 783
Ektron.Cms.API.Community.Tags.GetDefaultTags 783, 784
Ektron.Cms.API.Community.Tags.GetTagByID 784
Ektron.Cms.API.Community.Tags.GetTagByName 784
Ektron.Cms.API.Community.Tags.GetTagCloud 785
Ektron.Cms.API.Community.Tags.GetTagsForContent 785
Ektron.Cms.API.Community.Tags.GetTagsForObject 786
Ektron.Cms.API.Community.Tags.GetTagsForUser 787
Ektron.Cms.API.Community.Tags.RemoveTagAsDefault 788
Ektron.Cms.API.Community.Tags.SaveTagAsDefault 788
Ektron.Cms.API.Content 788
Ektron.Cms.API.Content namespace 788
Classes 789
Ektron.Cms.API.Content.Asset 789
Ektron.Cms.API.Content.Asset.GetAssetMgtConfigInfo 790
Ektron.Cms.API.Content.Asset.GetAssetMgtInfo 790
Ektron.Cms.API.Content.Asset.GetAssetSupertypes 790
Ektron.Cms.API.Content.Asset.GetEditSnippet 790
Ektron.Cms.API.Content.Asset.GetMyAssetMgtID 791
Ektron.Cms.API.Content.Asset.GetViewUrl 791
Ektron.Cms.API.Content.Asset.MakeCopyDMSAsset 791
Ektron.Cms.API.Content.Asset.SetAssetMgtConfigInfo 792
Ektron.Cms.API.Content.Blog 792
Ektron.Cms.API.Content.Blog.AddBlog 793
Ektron.Cms.API.Content.Blog.AddCommentForPost 794
Ektron.Cms.API.Content.Blog.AddPost 794
Ektron.Cms.API.Content.Blog.DeleteBlog 796
Ektron.Cms.API.Content.Blog.DeleteBlogPostComment 796
Ektron.Cms.API.Content.Blog.DeletePost 796
Ektron.Cms.API.Content.Blog.GetBlankBlogPostData 796
Ektron.Cms.API.Content.Blog.GetBlog 797, 800
Ektron.Cms.API.Content.Blog.GetBlogData 801, 802, 803
Ektron.Cms.API.Content.Blog.GetBlogDataForPost 803
Ektron.Cms.API.Content.Blog.GetBlogPostData 803
Ektron.Cms.API.Content.Blog.GetBlogRoll 804, 805
Ektron.Cms.API.Content.Blog.GetPost 805
Ektron.Cms.API.Content.Blog.GetPostbyID 805
Ektron.Cms.API.Content.Blog.GetUserBlog 806
Ektron.Cms.API.Content.Blog.UpdateBlogProperties 806
Ektron.Cms.API.Content.Content 806
Ektron.Cms.API.Content.Content.AddAsset 808
Ektron.Cms.API.Content.Content.AddContent 809, 811, 812
Ektron.Cms.API.Content.Content.AddXmlConfiguration 813
Ektron.Cms.API.Content.Content.CheckOutContent 813
Ektron.Cms.API.Content.Content.CopyContentToFolder 814
Ektron.Cms.API.Content.Content.DeleteContentItem 814
Ektron.Cms.API.Content.Content.DeleteXmlConfiguration 815

Ektron.Cms.API.Content.Content.EditAsset 816
Ektron.Cms.API.Content.Content.GetAddViewLanguage 816
Ektron.Cms.API.Content.Content.GetAllComments 816
Ektron.Cms.API.Content.Content.GetAllTemplates 817
Ektron.Cms.API.Content.Content.GetAllUnassignedItemApprovals 817
Ektron.Cms.API.Content.Content.GetAllXmlConfigurations 817
Ektron.Cms.API.Content.Content.GetChildContent 818, 819
Ektron.Cms.API.Content.Content.GetCommunityTemplate 820
Ektron.Cms.API.Content.Content.GetContent 821, 822
Ektron.Cms.API.Content.Content.GetContentByHistory 823
Ektron.Cms.API.Content.Content.GetContentDisplayLink 823
Ektron.Cms.API.Content.Content.GetContentForEditing 823
Ektron.Cms.API.Content.Content.GetContentReport 824
Ektron.Cms.API.Content.Content.GetContentState 825
Ektron.Cms.API.Content.Content.GetContentStatus 825
Ektron.Cms.API.Content.Content.GetDomain 825
Ektron.Cms.API.Content.Content.GetExpireContent 826
Ektron.Cms.API.Content.Content.GetList 826
Ektron.Cms.API.Content.Content.MoveContentToFolder 827
Ektron.Cms.API.Content.Content.PublishContent 827
Ektron.Cms.API.Content.Content.RelicateXmlConfiguration 828
Ektron.Cms.API.Content.Content.RenameContent 828
Ektron.Cms.API.Content.Content.ReplicateXmlConfiguration 828
Ektron.Cms.API.Content.Content.RestoreHistoryContent 829
Ektron.Cms.API.Content.Content.SaveContent 829
Ektron.Cms.API.Content.Content.SubmitForDelete 830
Ektron.Cms.API.Content.Content.UndoCheckout 830
Ektron.Cms.API.Content.Content.UpdateContentMetaData 830
Ektron.Cms.API.Content.Content.UpdatexmlCollectionPackage 831
Ektron.Cms.API.Content.Content.UpdateXmlConfiguration 831
Ektron.Cms.API.Content.ContentRating 831
Ektron.Cms.API.Content.ContentRating.GetContentRating 832
Ektron.Cms.API.Content.ContentRating.GetContentRatingAddRating 834
Ektron.Cms.API.Content.ContentRating.GetContentRatingResults 834
Ektron.Cms.API.Content.ContentRating.GetContentRatingStatistics 835
Ektron.Cms.API.Content.ContentRating.PurgeContentRatings 836
Ektron.Cms.API.Content.Form 836
Ektron.Cms.API.Content.Form.AssignFormTask 837
Ektron.Cms.API.Content.Form.EmailFormFieldData 837
Ektron.Cms.API.Content.Form.GetAllForms 838
Ektron.Cms.API.Content.Form.GetForm 839
Ektron.Cms.API.Content.Form.GetFormFieldData 839, 840
Ektron.Cms.API.Content.Form.GetFormFieldList 840
Ektron.Cms.API.Content.Form.GetFormFieldQuestions 840
Ektron.Cms.API.Content.Form.GetFormTitle 841
Ektron.Cms.API.Content.Form.SubmitFormFieldData 841
Ektron.Cms.API.Content.Form.UpdateFormFieldData 841
Ektron.Cms.API.Content.Taxonomy 842
Ektron.Cms.API.Content.Taxonomy.AddTaxonomyItem 843
Ektron.Cms.API.Content.Taxonomy.AddTaxonomySyncFolder 843
Ektron.Cms.API.Content.Taxonomy.CloneTaxonomy 844
Ektron.Cms.API.Content.Taxonomy.CreateTaxonomy 844
Ektron.Cms.API.Content.Taxonomy.DeleteTaxonomy 844
Ektron.Cms.API.Content.Taxonomy.GetTaxonomyIdByPath 845
Ektron.Cms.API.Content.Taxonomy.ImportTaxonomy 845, 846
Ektron.Cms.API.Content.Taxonomy.LoadTaxonomy 846
Ektron.Cms.API.Content.Taxonomy.ReadAllAssignedCategory 846
Ektron.Cms.API.Content.Taxonomy.ReadTaxonomy 847
Ektron.Cms.API.Content.Taxonomy.RemoveTaxonomyFolder 847
Ektron.Cms.API.Content.Taxonomy.RemoveTaxonomyItem 847
Ektron.Cms.API.Content.Taxonomy.ReOrderTaxonomyItems 847
Ektron.Cms.API.Content.Taxonomy.UpdateTaxonomy 848
Ektron.Cms.API.Content.ThreadedDiscussion 848
Ektron.Cms.API.Content.ThreadedDiscussion.AddBoard 849
Ektron.Cms.API.Content.ThreadedDiscussion.AddForum 850
Ektron.Cms.API.Content.ThreadedDiscussion.AddReply 852, 853
Ektron.Cms.API.Content.ThreadedDiscussion.AddTopic 854
Ektron.Cms.API.Content.ThreadedDiscussion.DeleteBoard 854

Ektron.Cms.API.Content.ThreadedDiscussion.DeleteForum 855
Ektron.Cms.API.Content.ThreadedDiscussion.DeleteTopic 855
Ektron.Cms.API.Content.ThreadedDiscussion.GetActiveTopics 855, 856
Ektron.Cms.API.Content.ThreadedDiscussion.GetActiveTopicsForForum 856
Ektron.Cms.API.Content.ThreadedDiscussion.GetActiveUsers 856
Ektron.Cms.API.Content.ThreadedDiscussion.GetBoardCategories 857
Ektron.Cms.API.Content.ThreadedDiscussion.GetBoardForums 857
Ektron.Cms.API.Content.ThreadedDiscussion.GetDiscussionBoard 858
Ektron.Cms.API.Content.ThreadedDiscussion.GetDiscussionBoardString 858
Ektron.Cms.API.Content.ThreadedDiscussion.GetForum 859
Ektron.Cms.API.Content.ThreadedDiscussion.GetRepliesForTopics 860
Ektron.Cms.API.Content.ThreadedDiscussion.GetTopic 861
Ektron.Cms.API.Content.ThreadedDiscussion.SearchReplies 861
Ektron.Cms.API.Content.ThreadedDiscussion.SubscribeToBoard 862
Ektron.Cms.API.Content.ThreadedDiscussion.SubscribeToForum 862
Ektron.Cms.API.Content.ThreadedDiscussion.SubscribeToTopic 862
Ektron.Cms.API.Content.ThreadedDiscussion.UnSubscribeToBoard 863
Ektron.Cms.API.Content.ThreadedDiscussion.UnSubscribeToForum 863
Ektron.Cms.API.Content.ThreadedDiscussion.UnSubscribeToTopic 864
Ektron.Cms.API.CustomFields 642
Ektron.Cms.API.CustomFields.AddNewEditableCustomFieldAssignments 644
Ektron.Cms.API.CustomFields.BoolToYesNo 644
Ektron.Cms.API.CustomFields.GetEditableCustomFieldAssignments 644
Ektron.Cms.API.CustomFields.GetEnhancedMetadataArea 645
Ektron.Cms.API.CustomFields.GetEnhancedMetadataScript 645
Ektron.Cms.API.CustomFields.GetFieldsByFolder 645
Ektron.Cms.API.CustomFields.GetSearchProperties 646, 647
Ektron.Cms.API.CustomFields.Populate_AssetRequestObjectFromForm 647
Ektron.Cms.API.CustomFields.ProcessCustomFields 648
Ektron.Cms.API.CustomFields.SearchAllAssets 648
Ektron.Cms.API.CustomFields.WriteFilteredMetadataForEdit 649
Ektron.Cms.API.CustomFields.WriteFilteredMetadataForView 649
Ektron.Cms.API.CustomFields.WriteLibrarySearchExtended 649
Ektron.Cms.API.CustomFields.WriteMetadataDefaultForEdit 650
Ektron.Cms.API.CustomFields.WriteMetadataTypeForView 650
Ektron.Cms.API.CustomFields.WritePublicSearch 650
Ektron.Cms.API.CustomFields.WritePublicSearchEx 651
Ektron.Cms.API.Folder 652
Ektron.Cms.API.Folder.AddCommunityFolder 653
Ektron.Cms.API.Folder.AddFolder 653
Ektron.Cms.API.Folder.CreateFolder 654
Ektron.Cms.API.Folder.DeleteFolderById 655
Ektron.Cms.API.Folder.GetChildFolders 656, 658
Ektron.Cms.API.Folder.GetFolder 659, 660
Ektron.Cms.API.Folder.GetFolderDataWithPermission 661
Ektron.Cms.API.Folder.GetFolderId 661
Ektron.Cms.API.Folder.GetFolderInfo 662
Ektron.Cms.API.Folder.GetPath 662
Ektron.Cms.API.Folder.IsFolderHidden 662
Ektron.Cms.API.Folder.RenameFolder 663
Ektron.Cms.API.Folder.UpdateFolder 663, 664
Ektron.Cms.API.Folder.UpdateFolderApprovalMethod 664
Ektron.Cms.API.Library 664
Ektron.Cms.API.Library.AddLibraryItem 665
Ektron.Cms.API.Library.AddLoadBalanceItem 665
Ektron.Cms.API.Library.DeleteLibraryItem 666

Ektron.Cms.API.Library.DeleteLibraryItemById 666
Ektron.Cms.API.Library.DeleteLoadBalanceItem 666
Ektron.Cms.API.Library.GetAllChildLibItems 666
Ektron.Cms.API.Library.GetAllLoadBalancePaths 669
Ektron.Cms.API.Library.GetLibraryItem 670
Ektron.Cms.API.Library.GetLibrarySettings 670
Ektron.Cms.API.Library.GetLibraryTypes 671
Ektron.Cms.API.Library.GetLoadBalancePath 671
Ektron.Cms.API.Library.InternalLibrarySearch 671
Ektron.Cms.API.Library.UpdateLibrary 672
Ektron.Cms.API.Library.UpdateLibrarySettings 672
Ektron.Cms.API.Library.UpdateLoadBalanceSettings 672
Ektron.Cms.API.Library.UpdateQlinkTemplates 672
Ektron.Cms.API.Localization 673
Ektron.Cms.API.Localization.GetAddViewLanguage 674
Ektron.Cms.API.Localization.GetAddViewLanguageForAllContent 674
Ektron.Cms.API.Localization.GetAddViewLanguageForMenus 674
Ektron.Cms.API.Localization.GetContentItems 675
Ektron.Cms.API.Localization.GetFile 675
Ektron.Cms.API.Localization.GetFiles 675
Ektron.Cms.API.Localization.GetFilesInZipFile 676
Ektron.Cms.API.Localization.GetFlagUrl 676, 677
Ektron.Cms.API.Localization.GetInterchangeFile 678
Ektron.Cms.API.Localization.GetInterchangeFiles 678
Ektron.Cms.API.Localization.GetInterchangeFilesInZipFile 678
Ektron.Cms.API.Localization.GetJobs 679
Ektron.Cms.API.Localization.GetLocalizationUrl 680
Ektron.Cms.API.Localization.GetTranslationUploadDirectory 680
Ektron.Cms.API.Localization.StartExportContentForTranslation 680
Ektron.Cms.API.Localization.StartExportFolderForTranslation 681
Ektron.Cms.API.Localization.StartExportMenusForTranslation 681
Ektron.Cms.API.Localization.StartImportTranslation 682
Ektron.Cms.API.Messaging 682
Ektron.Cms.API.Messaging.GetUnreadMessageCountForUser 682
Ektron.Cms.API.Metadata 683
Ektron.Cms.API.Metadata.AddMetaDataType 684
Ektron.Cms.API.Metadata.DeleteMetadataType 686
Ektron.Cms.API.Metadata.GetContentMetadataList 686, 688
Ektron.Cms.API.Metadata.GetIntegratedSearchFolder 688
Ektron.Cms.API.Metadata.GetIntegratedSearchFolders 688
Ektron.Cms.API.Metadata.GetMetadataByFolder 689
Ektron.Cms.API.Metadata.GetMetadataType 689
Ektron.Cms.API.Metadata.GetMetaDataTypes 689
Ektron.Cms.API.Metadata.UpdateMetadataType 690
Ektron.Cms.API.Permissions 690
Ektron.Cms.API.Permissions.AddItemPermission 691
Ektron.Cms.API.Permissions.DeleteItemPermission 691
Ektron.Cms.API.Permissions.DisableContentInheritance 691
Ektron.Cms.API.Permissions.DisableFolderInheritance 692
Ektron.Cms.API.Permissions.EnableContentInheritance 692
Ektron.Cms.API.Permissions.EnableFolderInheritance 692
Ektron.Cms.API.Permissions.GetUserPermissions 693
Ektron.Cms.API.Permissions.LoadPermissions 694
Ektron.Cms.API.Permissions.UpdateItemPermission 694
Ektron.Cms.API.Search 864
Ektron.Cms.API.Search namespace 864
Classes 865
Ektron.Cms.API.Search.SearchManager 865
Ektron.Cms.API.Search.SearchManager.Execute 866, 867
Ektron.Cms.API.Search.SearchManager.Search 868, 869
Ektron.Cms.API.Search.SearchManager.SearchForCategories 870
Ektron.Cms.API.Search.SearchManager.SearchTable 870
Ektron.Cms.API.Site 694
Ektron.Cms.API.Site.BreakTaskPermissionInherit 695
Ektron.Cms.API.Site.DeleteCookie 696
Ektron.Cms.API.Site.GetAllActiveLanguages 696
Ektron.Cms.API.Site.GetAllLanguages 696, 697
Ektron.Cms.API.Site.GetClientDayNames 697
Ektron.Cms.API.Site.GetClientMonthNames 697
Ektron.Cms.API.Site.GetCookieValue 697
Ektron.Cms.API.Site.GetLanguage 698
Ektron.Cms.API.Site.GetLanguages 698
Ektron.Cms.API.Site.GetSiteVariables 698
Ektron.Cms.API.Site.GetTaskPermission 699

Ektron.Cms.API.Site.IsTaskPermissionInherited 699
Ektron.Cms.API.Site.SetCookieValue 699
Ektron.Cms.API.Site.SetTaskPermission 700
Ektron.Cms.API.Site.UpdateSiteEnabledLanguages 700
Ektron.Cms.API.Site.UpdateSiteVariables 700
Ektron.Cms.API.SiteMap 701
Ektron.Cms.API.SiteMap.GetFolderBreadcrumbPath 701
Ektron.Cms.API.SiteMap.GetWebsiteMap 703
Ektron.Cms.API.SiteMap.RenderFolderBreadcrumbHtml 704
Ektron.Cms.API.SiteMap.RenderWebsiteMap 705
Ektron.Cms.API.Subscription 705
Ektron.Cms.API.Subscription.AddEmailFrom 706
Ektron.Cms.API.Subscription.AddSubscription 707
Ektron.Cms.API.Subscription.AddSubscriptionMessage 707
Ektron.Cms.API.Subscription.DeleteEmailFrom 707
Ektron.Cms.API.Subscription.DeleteSubscription 707
Ektron.Cms.API.Subscription.DeleteSubscriptionMessage 708
Ektron.Cms.API.Subscription.GetAllActiveSubscriptions 708
Ektron.Cms.API.Subscription.GetAllEmailFrom 708
Ektron.Cms.API.Subscription.GetAllSubscriptionMessages 708
Ektron.Cms.API.Subscription.GetAllSubscriptions 709
Ektron.Cms.API.Subscription.GetFolderInheritedFrom 709
Ektron.Cms.API.Subscription.GetSubscription 709
Ektron.Cms.API.Subscription.GetSubscriptionMessage 709
Ektron.Cms.API.Subscription.GetSubscriptionMessagesForType 710
Ektron.Cms.API.Subscription.GetSubscriptionPropertiesForContent 710
Ektron.Cms.API.Subscription.GetSubscriptionPropertiesForFolder 710
Ektron.Cms.API.Subscription.GetSubscriptionsForContent 711
Ektron.Cms.API.Subscription.GetSubscriptionsForFolder 711
Ektron.Cms.API.Subscription.SendWebAlert 711
Ektron.Cms.API.Subscription.UpdateEmailFrom 712
Ektron.Cms.API.Subscription.UpdateSubscription 712
Ektron.Cms.API.Subscription.UpdateSubscriptionMessage 712
Ektron.Cms.API.Subscription.UpdateSubscriptionPropertiesForContent 712
Ektron.Cms.API.Subscription.UpdateSubscriptionPropertiesForFolder 713
Ektron.Cms.API.Task 871
Ektron.Cms.API.Task namespace 871
Classes 871
Ektron.Cms.API.Task.Task 871
Ektron.Cms.API.Task.Task.AddTask 872
Ektron.Cms.API.Task.Task.AssignTaskToUser 872, 873
Ektron.Cms.API.Task.Task.AssignTaskToUsergroup 873, 874
Ektron.Cms.API.Task.Task.DeleteTask 874, 875
Ektron.Cms.API.Task.Task.DeleteTasks 875, 876
Ektron.Cms.API.Task.Task.GetTask 876
Ektron.Cms.API.Task.Task.GetTasks 876, 877
Ektron.Cms.API.Task.Task.GetTasksAssignedByUser 877, 878
Ektron.Cms.API.Task.Task.GetTasksAssignedToUser 878, 879
Ektron.Cms.API.Task.Task.GetTasksAssignedToUsergroup 880
Ektron.Cms.API.Task.Task.GetTaskState 881
Ektron.Cms.API.Task.Task.SetStateForTask 881
Ektron.Cms.API.Task.Task.UpdateTask 881
Ektron.Cms.API.Task.TaskCategory 882
Ektron.Cms.API.Task.TaskCategory.AddTaskCategory 882
Ektron.Cms.API.Task.TaskCategory.DeleteTaskCategory 882
Ektron.Cms.API.Task.TaskCategory.GetAllCategories 883
Ektron.Cms.API.Task.TaskCategory.GetTaskCategory 883
Ektron.Cms.API.Task.TaskCategory.UpdateTaskCategory 883
Ektron.Cms.API.Task.TaskCategoryType 884
Ektron.Cms.API.Task.TaskCategoryType.AddTaskType 884
Ektron.Cms.API.Task.TaskCategoryType.DeleteTaskType 885
Ektron.Cms.API.Task.TaskCategoryType.GetTaskType 885
Ektron.Cms.API.Task.TaskCategoryType.UpdateTaskType 885
Ektron.Cms.API.UrlAliasing 886
Ektron.Cms.API.UrlAliasing namespace 886
Classes 886
Ektron.Cms.API.UrlAliasing.UrlAliasAuto 886
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.Add 887
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.ClearCache 887
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.Delete 888
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetAutoAliasNameTypes 888

Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetAutoAliasTypes 888
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetFolderItem 888
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetFolderList 889
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetFolderSource 889
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetItem 889
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetItemForContent 890
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetList 890
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetListForContent 891
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetTarget 892
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetTaxonomyItem 893
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetTaxonomyList 893
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.GetTaxonomySource 893
Ektron.Cms.API.UrlAliasing.UrlAliasAuto.Update 893
Ektron.Cms.API.UrlAliasing.UrlAliasCommon 894
Ektron.Cms.API.UrlAliasing.UrlAliasCommon.GetAliasForContent 894
Ektron.Cms.API.UrlAliasing.UrlAliasCommon.GetTarget 895
Ektron.Cms.API.UrlAliasing.UrlAliasManual 895
Ektron.Cms.API.UrlAliasing.UrlAliasManual.Add 896
Ektron.Cms.API.UrlAliasing.UrlAliasManual.ClearCache 896
Ektron.Cms.API.UrlAliasing.UrlAliasManual.Delete 896
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetDefaultAlias 897
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetFileExtensions 897
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetItem 897
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetList 898, 899, 900, 901
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetSiteList 902
Ektron.Cms.API.UrlAliasing.UrlAliasManual.GetTemplateList 902
Ektron.Cms.API.UrlAliasing.UrlAliasManual.Update 902
Ektron.Cms.API.UrlAliasing.UrlAliasManual.ValidateName 903
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx 903
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.Add 904
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.ClearCache 904
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.Delete 904
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.EvaluateExpression 904
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.GetItem 905
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.GetList 905
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.GetPriorityList 906
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.GetTarget 906
Ektron.Cms.API.UrlAliasing.UrlAliasRegEx.Update 906
Ektron.Cms.API.UrlAliasing.UrlAliasSettings 907
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.AutoCacheDuration 909
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.AutoCacheSize 909
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.DefaultPage 909
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.DisableLanguageAwareness 909
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableAutoAliasing 909
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableAutoCaching 910
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnabledTracing 910
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableManualAliasing 910
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableManualCaching 910
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableRegExAliasing 910
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.EnableRegExCaching 911
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.IsAutoAliasEnabled 911
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.IsManualAliasEnabled 911
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.IsRegExAliasEnabled 911
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.ManualCacheDuration 912
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.ManualCacheSize 912

Ektron.Cms.API.UrlAliasing.UrlAliasSettings.OverrideTemplate 912
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.PageExtensions 912
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.QueryStringAction 912
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.RefreshSettingsCache 907
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.RegExCacheDuration 912
Ektron.Cms.API.UrlAliasing.UrlAliasSettings.RegExCacheSize 913
Ektron.Cms.API.User 913
Ektron.Cms.API.User namespace 913
Classes 913
Ektron.Cms.API.User.ADUser 913
Ektron.Cms.API.User.ADUser.AddADGroupToCMS 914
Ektron.Cms.API.User.ADUser.AddADUsersToCMSByUsername 914
Ektron.Cms.API.User.ADUser.DeSynchUserGroups 915
Ektron.Cms.API.User.ADUser.DeSynchUsers 915
Ektron.Cms.API.User.ADUser.GetADMapping 915
Ektron.Cms.API.User.ADUser.GetADStatus 916
Ektron.Cms.API.User.ADUser.GetAvailableADGroups 916
Ektron.Cms.API.User.ADUser.GetAvailableADUsers 916
Ektron.Cms.API.User.ADUser.GetCMSGroupsToSync 917
Ektron.Cms.API.User.ADUser.GetCMSRelationshipsToSync 917
Ektron.Cms.API.User.ADUser.GetCMSUsersToSync 917
Ektron.Cms.API.User.ADUser.GetDomains 918
Ektron.Cms.API.User.ADUser.MapCMSUserGroupToAD 918
Ektron.Cms.API.User.ADUser.MapCMSUserToAD 918
Ektron.Cms.API.User.ADUser.SynchCMSGroupsToAD 919
Ektron.Cms.API.User.ADUser.SynchCMSRelationshipsToAD 919
Ektron.Cms.API.User.ADUser.SynchCMSUsersToAD 919
Ektron.Cms.API.User.ADUser.UpdateADMapping 920
Ektron.Cms.API.User.ADUser.UpdateADUsersGroups 920
Ektron.Cms.API.User.ADUser.UpdateUsersInfoFromAD 920
Ektron.Cms.API.User.User 920
Ektron.Cms.API.User.User.ActivateUserAccount 923
Ektron.Cms.API.User.User.AddMembershipUser 923
Ektron.Cms.API.User.User.AddNewUser 925
Ektron.Cms.API.User.User.AddRolePermission 926
Ektron.Cms.API.User.User.AddUser 926, 927
Ektron.Cms.API.User.User.AddUserGroup 927
Ektron.Cms.API.User.User.AddUserToGroup 927
Ektron.Cms.API.User.User.AutoLogInUser 928
Ektron.Cms.API.User.User.CreateBlankUserObject 928, 929
Ektron.Cms.API.User.User.DeleteUserByID 929
Ektron.Cms.API.User.User.DeleteUserByIds 929
Ektron.Cms.API.User.User.DeleteUserFromGroup 929
Ektron.Cms.API.User.User.DeleteUserGroup 930
Ektron.Cms.API.User.User.EditUserCustomProperties 930
Ektron.Cms.API.User.User.GetActiveUser 930
Ektron.Cms.API.User.User.GetADMapping 931
Ektron.Cms.API.User.User.GetAllCmsUsers 931
Ektron.Cms.API.User.User.GetAllCustomProperties 931
Ektron.Cms.API.User.User.GetAllUserByCustomProperty 932
Ektron.Cms.API.User.User.GetAllUserGroups 932, 933
Ektron.Cms.API.User.User.GetAllUsers 933, 934
Ektron.Cms.API.User.User.GetGroupsUserIsIn 934
Ektron.Cms.API.User.User.GetMembershipUser 934
Ektron.Cms.API.User.User.GetUser 935
Ektron.Cms.API.User.User.GetUserByUsername 935
Ektron.Cms.API.User.User.GetUserCustomProperties 937
Ektron.Cms.API.User.User.GetUserEmailInfo 937
Ektron.Cms.API.User.User.GetUserForFolderAdmin 937
Ektron.Cms.API.User.User.GetUserGroup 937
Ektron.Cms.API.User.User.GetUserGroupByName 938
Ektron.Cms.API.User.User.GetUserGroupForFolderAdmin 938
Ektron.Cms.API.User.User.GetUserPermissions 938
Ektron.Cms.API.User.User.GetUserPreference 939
Ektron.Cms.API.User.User.GetUsers 940
Ektron.Cms.API.User.User.GetUsersForTask 940
Ektron.Cms.API.User.User.GetUsersNotInGroup 941
Ektron.Cms.API.User.User.IsAdmin 941
Ektron.Cms.API.User.User.IsAGroupMember 942
Ektron.Cms.API.User.User.IsARoleMember 943, 944
Ektron.Cms.API.User.User.IsARoleMemberForContent 945

Ektron.Cms.API.User.User.IsARoleMemberForContent_FolderUserAdmin 945

Ektron.Cms.API.User.User.IsARoleMemberForFolder 946

Ektron.Cms.API.User.User.IsARoleMemberForFolder_FolderUserAdmin 946

Ektron.Cms.API.User.User.IsARoleMemberForFolder_MetadataAdmin 946

Ektron.Cms.API.User.User.LogInUser 947, 948

Ektron.Cms.API.User.User.ReadCustomProperties 950

Ektron.Cms.API.User.User.ResetMembershipUserPassword 950, 951

Ektron.Cms.API.User.User.SetAuthenticationCookie 951

Ektron.Cms.API.User.User.UnsubscribeUser 951

Ektron.Cms.API.User.User.UpdateCustomPropertiesItemOrder 952

Ektron.Cms.API.User.User.UpdateUser 952, 953

Ektron.Cms.API.User.User.UpDateUserGroup 953

Ektron.Cms.API.User.User.UpdateUserPreferences 954

Ektron.Cms.API.User.User.UserIdAuthenticated 955

Ektron.Cms.API.User.User.UserObject 954

Ektron.Cms.API.Utilities 713

Ektron.Cms.API.Utilities.LoadPermissions 714

Ektron.Cms.Commerce 1201

Ektron.Cms.Commerce namespace 1201

- Classes 1204
- Structs, Records, Enums 1356

Ektron.Cms.Commerce.AddressApi 574

Ektron.Cms.Commerce.AddressApi.Add 575

Ektron.Cms.Commerce.AddressApi.Delete 576

Ektron.Cms.Commerce.AddressApi.GetItem 576

Ektron.Cms.Commerce.AddressApi.GetList 576

Ektron.Cms.Commerce.AddressApi.Update 576

Ektron.Cms.Commerce.AddressApi.UpdateOrderAddress 577

Ektron.Cms.Commerce.AddressData 1206

Ektron.Cms.Commerce.AddressData.AddressLine1 1207

Ektron.Cms.Commerce.AddressData.AddressLine2 1207

Ektron.Cms.Commerce.AddressData.City 1207

Ektron.Cms.Commerce.AddressData.Company 1207

Ektron.Cms.Commerce.AddressData.Country 1207

Ektron.Cms.Commerce.AddressData.Id 1208

Ektron.Cms.Commerce.AddressData.Name 1208

Ektron.Cms.Commerce.AddressData.Phone 1208

Ektron.Cms.Commerce.AddressData.PostalCode 1208

Ektron.Cms.Commerce.AddressData.Region 1208

Ektron.Cms.Commerce.AddressProperty 1357

Ektron.Cms.Commerce.AddressProperty enumeration 1357

Ektron.Cms.Commerce.AdjustedBasketItem 1208

Ektron.Cms.Commerce.AdjustedBasketItem.AdjustedBasketItem 1209

Ektron.Cms.Commerce.AdjustedBasketItem.AdjustedTotal 1210

Ektron.Cms.Commerce.AdjustedBasketItem.ApplicableTaxRate 1210

Ektron.Cms.Commerce.AdjustedBasketItem.AppliedCoupons 1210

Ektron.Cms.Commerce.AdjustedBasketItem.HasFreeShipping 1210

Ektron.Cms.Commerce.AdjustedBasketItem.TaxesApplied 1211

Ektron.Cms.Commerce.AdjustedBasketItem.TotalCouponDiscount 1211

Ektron.Cms.Commerce.AppliedCouponData 1211

Ektron.Cms.Commerce.AppliedCouponData.AppliedCouponData 1212

Ektron.Cms.Commerce.AppliedCouponData.Coupon 1212

Ektron.Cms.Commerce.AppliedCouponData.CouponDiscount 1212

Ektron.Cms.Commerce.AppliedCouponDataCollection 1213

Ektron.Cms.Commerce.AppliedCouponDataCollection.FindByCouponId 1213

Ektron.Cms.Commerce.AppliedCouponDataCollection.ItemAdded 1213

Ektron.Cms.Commerce.AppliedCouponEventArgs 1214

Ektron.Cms.Commerce.AppliedCouponEventArgs.AppliedCoupon 1215

Ektron.Cms.Commerce.AppliedCouponEventArgs.AppliedCouponEventArgs 1214

Ektron.Cms.Commerce.Basket 1215

Ektron.Cms.Commerce.Basket.AddProduct 1216, 1217

Ektron.Cms.Commerce.Basket.AddProductVariant 1217

Ektron.Cms.Commerce.Basket.Decrement 1217
Ektron.Cms.Commerce.Basket.GetItemIndex 1217
Ektron.Cms.Commerce.Basket.GetKitConfiguration 1218
Ektron.Cms.Commerce.Basket.GetKitConfigurationAsAdmin 1218
Ektron.Cms.Commerce.Basket.IncrementItemQuantity 1218
Ektron.Cms.Commerce.Basket.MakeDefaultBasket 1218
Ektron.Cms.Commerce.Basket.RemoveItem 1218
Ektron.Cms.Commerce.Basket.RequestInformation 1220
Ektron.Cms.Commerce.Basket.SaveBasket 1219
Ektron.Cms.Commerce.Basket.UpdateItemQuantity 1219
Ektron.Cms.Commerce.Basket.UpdateKitConfiguration 1219
Ektron.Cms.Commerce.BasketApi 577
Ektron.Cms.Commerce.BasketApi.CalculateBasket 578, 579
Ektron.Cms.Commerce.BasketApi.CreateNewBasket 579
Ektron.Cms.Commerce.BasketApi.Delete 580
Ektron.Cms.Commerce.BasketApi.Empty 580
Ektron.Cms.Commerce.BasketApi.GetDefaultBasket 581
Ektron.Cms.Commerce.BasketApi.GetItem 581
Ektron.Cms.Commerce.BasketApi.GetList 582
Ektron.Cms.Commerce.BasketApi.SetDefaultBasket 582, 583
Ektron.Cms.Commerce.BasketCalculatorData 1220
Ektron.Cms.Commerce.BasketCalculatorData.AdjustedItems 1221
Ektron.Cms.Commerce.BasketCalculatorData.Basket 1221
Ektron.Cms.Commerce.BasketCalculatorData.BasketCalculatorData 1220, 1221
Ektron.Cms.Commerce.BasketCalculatorData.BasketCoupons 1222
Ektron.Cms.Commerce.BasketCalculatorData.BasketTotal 1222
Ektron.Cms.Commerce.BasketCalculatorData.HasFreeShipping 1222
Ektron.Cms.Commerce.BasketCalculatorData.ShipFrom 1222
Ektron.Cms.Commerce.BasketCalculatorData.ShippingRates 1222
Ektron.Cms.Commerce.BasketCalculatorData.ShipTo 1222
Ektron.Cms.Commerce.BasketCalculatorData.TaxClassSubtotals 1223
Ektron.Cms.Commerce.BasketCalculatorData.TotalCouponDiscount 1223
Ektron.Cms.Commerce.BasketCalculatorData.TotalShippingCost 1223
Ektron.Cms.Commerce.BasketCalculatorData.TotalTaxes 1223
Ektron.Cms.Commerce.BasketCalculatorData.ValidationResults 1223
Ektron.Cms.Commerce.BasketData 1224
Ektron.Cms.Commerce.BasketData.DateCreated 1224
Ektron.Cms.Commerce.BasketData.DateModified 1224
Ektron.Cms.Commerce.BasketData.Id 1225
Ektron.Cms.Commerce.BasketData.IsDefault 1225
Ektron.Cms.Commerce.BasketData.Items 1225
Ektron.Cms.Commerce.BasketData.Name 1225
Ektron.Cms.Commerce.BasketData.ShippingAddressId 1225
Ektron.Cms.Commerce.BasketData.ShippingMethodId 1226
Ektron.Cms.Commerce.BasketData.ShopperId 1226
Ektron.Cms.Commerce.BasketData.Subtotal 1226
Ektron.Cms.Commerce.BasketData.UserId 1226
Ektron.Cms.Commerce.BasketItem 1226
Ektron.Cms.Commerce.BasketItem.AdjustedTotal 1228
Ektron.Cms.Commerce.BasketItem.BasketItem 1227
Ektron.Cms.Commerce.BasketItem.Configuration 1228
Ektron.Cms.Commerce.BasketItem.Dimensions 1228
Ektron.Cms.Commerce.BasketItem.Id 1228
Ektron.Cms.Commerce.BasketItem.IsTangible 1228
Ektron.Cms.Commerce.BasketItem.ListPrice 1229
Ektron.Cms.Commerce.BasketItem.Pricing 1229
Ektron.Cms.Commerce.BasketItem.ProductId 1229
Ektron.Cms.Commerce.BasketItem.ProductLanguage 1229
Ektron.Cms.Commerce.BasketItem.ProductSku 1229
Ektron.Cms.Commerce.BasketItem.ProductTitle 1230
Ektron.Cms.Commerce.BasketItem.ProductType 1230
Ektron.Cms.Commerce.BasketItem.Quantity 1230
Ektron.Cms.Commerce.BasketItem.Quicklink 1230
Ektron.Cms.Commerce.BasketItem.SalePrice 1230
Ektron.Cms.Commerce.BasketItem.TaxClassId 1231
Ektron.Cms.Commerce.BasketItem.Variant 1231
Ektron.Cms.Commerce.BasketItem.Weight 1231
Ektron.Cms.Commerce.BasketItemCollection 1231
Ektron.Cms.Commerce.BasketItemCollection.AddRange 1232

Ektron.Cms.Commerce.BasketItemCollection.BasketItemCollection 1231
Ektron.Cms.Commerce.BasketItemCollection.Find 1232
Ektron.Cms.Commerce.BasketItemCollection.FindAll 1232
Ektron.Cms.Commerce.BasketItemCollection.FindIndex 1232
Ektron.Cms.Commerce.BasketVariantData 1233
Ektron.Cms.Commerce.BasketVariantData.BasketVariantData 1233
Ektron.Cms.Commerce.BasketVariantData.Dimensions 1233
Ektron.Cms.Commerce.BasketVariantData.Id 1234
Ektron.Cms.Commerce.BasketVariantData.Language 1234
Ektron.Cms.Commerce.BasketVariantData.ListPrice 1234
Ektron.Cms.Commerce.BasketVariantData.Pricing 1234
Ektron.Cms.Commerce.BasketVariantData.SalePrice 1235
Ektron.Cms.Commerce.BasketVariantData.Sku 1235
Ektron.Cms.Commerce.BasketVariantData.TaxClassId 1235
Ektron.Cms.Commerce.BasketVariantData.Title 1235
Ektron.Cms.Commerce.BasketVariantData.Weight 1235
Ektron.Cms.Commerce.BundleData 1236
Ektron.Cms.Commerce.BundleData.BundleData 1236
Ektron.Cms.Commerce.BundleData.BundledItems 1236
Ektron.Cms.Commerce.CatalogData 1236
Ektron.Cms.Commerce.CatalogData.Id 1237
Ektron.Cms.Commerce.CatalogData.Name 1237
Ektron.Cms.Commerce.CatalogEntryApi 583
Ektron.Cms.Commerce.CatalogEntryApi.Add 584
Ektron.Cms.Commerce.CatalogEntryApi.AddAndCheckIn 584
Ektron.Cms.Commerce.CatalogEntryApi.AddAndPublish 584
Ektron.Cms.Commerce.CatalogEntryApi.AddAndSave 584
Ektron.Cms.Commerce.CatalogEntryApi.Checkout 585
Ektron.Cms.Commerce.CatalogEntryApi.Delete 585
Ektron.Cms.Commerce.CatalogEntryApi.GetAttributeEntryList 585
Ektron.Cms.Commerce.CatalogEntryApi.GetItem 586
Ektron.Cms.Commerce.CatalogEntryApi.GetItemEdit 586
Ektron.Cms.Commerce.CatalogEntryApi.GetList 587
Ektron.Cms.Commerce.CatalogEntryApi.GetTaxonomyEntryList 587
Ektron.Cms.Commerce.CatalogEntryApi.Restore 587
Ektron.Cms.Commerce.CatalogEntryApi.UndoCheckOut 587
Ektron.Cms.Commerce.CatalogEntryApi.Update 588
Ektron.Cms.Commerce.CatalogEntryApi.UpdateAndCheckIn 588
Ektron.Cms.Commerce.CatalogEntryApi.UpdateAndPublish 588
Ektron.Cms.Commerce.ContractApi 588
Ektron.Cms.Commerce.ContractApi.Add 589
Ektron.Cms.Commerce.ContractApi.Delete 589
Ektron.Cms.Commerce.ContractApi.GetItem 589
Ektron.Cms.Commerce.ContractApi.GetList 590
Ektron.Cms.Commerce.ContractApi.Update 590
Ektron.Cms.Commerce.ContractData 1237
Ektron.Cms.Commerce.ContractData.ContractData 1238
Ektron.Cms.Commerce.ContractData.Id 1238
Ektron.Cms.Commerce.ContractData.Name 1239
Ektron.Cms.Commerce.CountryApi 590
Ektron.Cms.Commerce.CountryApi.Add 591
Ektron.Cms.Commerce.CountryApi.CanDelete 591
Ektron.Cms.Commerce.CountryApi.Delete 591
Ektron.Cms.Commerce.CountryApi.GetItem 592
Ektron.Cms.Commerce.CountryApi.GetList 592
Ektron.Cms.Commerce.CountryApi.Update 592
Ektron.Cms.Commerce.CountryData 1239
Ektron.Cms.Commerce.CountryData.CountryData 1239, 1240
Ektron.Cms.Commerce.CountryData.Enabled 1241
Ektron.Cms.Commerce.CountryData.Id 1241
Ektron.Cms.Commerce.CountryData.LongIsoCode 1241
Ektron.Cms.Commerce.CountryData.Name 1241
Ektron.Cms.Commerce.CountryData.ShortIsoCode 1241
Ektron.Cms.Commerce.CountryData.ToString 1240
Ektron.Cms.Commerce.CountryData.Validate 1240
Ektron.Cms.Commerce.CountryProperty 1357
Ektron.Cms.Commerce.CountryProperty enumeration 1357
Ektron.Cms.Commerce.CountryTaxRateData 1242
Ektron.Cms.Commerce.CountryTaxRateData.CountryData 1243
Ektron.Cms.Commerce.CountryTaxRateData.CountryTaxRateData 1242, 1243
Ektron.Cms.Commerce.CouponApi 592
Ektron.Cms.Commerce.CouponApi.Add 593
Ektron.Cms.Commerce.CouponApi.AddCouponToCatalog 594

Ektron.Cms.Commerce.CouponApi.AddCouponToProduct 594
Ektron.Cms.Commerce.CouponApi.AddCouponToTaxonomy 594
Ektron.Cms.Commerce.CouponApi.CalculateBasketCoupons 595
Ektron.Cms.Commerce.CouponApi.Deactivate 595
Ektron.Cms.Commerce.CouponApi.Delete 595
Ektron.Cms.Commerce.CouponApi.DeleteCouponProducts 595
Ektron.Cms.Commerce.CouponApi.GetCatalogList 596
Ektron.Cms.Commerce.CouponApi.GetCouponsAppliedToBasket 596
Ektron.Cms.Commerce.CouponApi.GetItem 596
Ektron.Cms.Commerce.CouponApi.GetList 596
Ektron.Cms.Commerce.CouponApi.GetProductList 597
Ektron.Cms.Commerce.CouponApi.GetTaxonomyList 597
Ektron.Cms.Commerce.CouponApi.IsCouponAppliedToProduct 598
Ektron.Cms.Commerce.CouponApi.IsCouponUsedForBasket 598
Ektron.Cms.Commerce.CouponApi.IsCouponUsedForOrder 598
Ektron.Cms.Commerce.CouponApi.SaveCouponApplications 598
Ektron.Cms.Commerce.CouponApi.Update 599
Ektron.Cms.Commerce.CouponApi.ValidateCoupon 599
Ektron.Cms.Commerce.CouponData 1243
Ektron.Cms.Commerce.CouponData.Code 1244
Ektron.Cms.Commerce.CouponData.CouponData 1244
Ektron.Cms.Commerce.CouponData.CouponType 1245
Ektron.Cms.Commerce.CouponData.CurrencyId 1245
Ektron.Cms.Commerce.CouponData.Description 1245
Ektron.Cms.Commerce.CouponData.DiscountType 1245
Ektron.Cms.Commerce.CouponData.DiscountValue 1245
Ektron.Cms.Commerce.CouponData.ExpirationDate 1245
Ektron.Cms.Commerce.CouponData.Id 1246
Ektron.Cms.Commerce.CouponData.IsActive 1246
Ektron.Cms.Commerce.CouponData.IsCombinable 1246
Ektron.Cms.Commerce.CouponData.MaximumAmount 1246
Ektron.Cms.Commerce.CouponData.MaximumUses 1246
Ektron.Cms.Commerce.CouponData.MinimumAmount 1247
Ektron.Cms.Commerce.CouponData.OnePerCustomer 1247
Ektron.Cms.Commerce.CouponData.StartDate 1247
Ektron.Cms.Commerce.CouponEntryData 1247
Ektron.Cms.Commerce.CouponEntryData.CouponEntryData 1248
Ektron.Cms.Commerce.CouponEntryData.CouponId 1248
Ektron.Cms.Commerce.CouponEntryData.DataState 1249
Ektron.Cms.Commerce.CouponEntryData.IsIncluded 1249
Ektron.Cms.Commerce.CouponEntryData.ObjectId 1249
Ektron.Cms.Commerce.CouponEntryData.ObjectType 1249
Ektron.Cms.Commerce.CouponEntryData.Path 1249
Ektron.Cms.Commerce.CouponEntryData.Title 1250
Ektron.Cms.Commerce.CouponProperty 1358
Ektron.Cms.Commerce.CouponProperty enumeration 1358
Ektron.Cms.Commerce.CurrencyApi 600
Ektron.Cms.Commerce.CurrencyApi.Add 600
Ektron.Cms.Commerce.CurrencyApi.CanDelete 601
Ektron.Cms.Commerce.CurrencyApi.Delete 601
Ektron.Cms.Commerce.CurrencyApi.GetItem 601
Ektron.Cms.Commerce.CurrencyApi.GetList 601
Ektron.Cms.Commerce.CurrencyApi.Update 602
Ektron.Cms.Commerce.CurrencyData 1250
Ektron.Cms.Commerce.CurrencyData.AlphaIsoCode 1250
Ektron.Cms.Commerce.CurrencyData.CultureCode 1251
Ektron.Cms.Commerce.CurrencyData.CurrencySymbol 1251
Ektron.Cms.Commerce.CurrencyData.Enabled 1251
Ektron.Cms.Commerce.CurrencyData.Id 1251
Ektron.Cms.Commerce.CurrencyData.ISOCurrencySymbol 1251
Ektron.Cms.Commerce.CurrencyData.Name 1252
Ektron.Cms.Commerce.CurrencyPricingData 1252
Ektron.Cms.Commerce.CurrencyPricingData.AlphaIsoCode 1254
Ektron.Cms.Commerce.CurrencyPricingData.Clone 1253
Ektron.Cms.Commerce.CurrencyPricingData.CurrencyId 1254
Ektron.Cms.Commerce.CurrencyPricingData.CurrencyPricingData 1252, 1253
Ektron.Cms.Commerce.CurrencyPricingData.GetSalePrice 1254
Ektron.Cms.Commerce.CurrencyPricingData.ListPrice 1255
Ektron.Cms.Commerce.CurrencyPricingData.NumericIsoCode 1255
Ektron.Cms.Commerce.CurrencyPricingData.PricingType 1255

Ektron.Cms.Commerce.CurrencyPricingData.SalePrice 1255
Ektron.Cms.Commerce.CurrencyPricingData.TierPrices 1255
Ektron.Cms.Commerce.CurrencyPricingData.Validate 1254
Ektron.Cms.Commerce.CurrencyProperty 1358
Ektron.Cms.Commerce.CurrencyProperty enumeration 1358
Ektron.Cms.Commerce.CustomerApi 602
Ektron.Cms.Commerce.CustomerApi.Add 603
Ektron.Cms.Commerce.CustomerApi.Delete 603
Ektron.Cms.Commerce.CustomerApi.GetItem 604
Ektron.Cms.Commerce.CustomerApi.GetList 604
Ektron.Cms.Commerce.CustomerApi.GetOrderList 604
Ektron.Cms.Commerce.CustomerApi.GetOrderTotal 605
Ektron.Cms.Commerce.CustomerApi.Update 605
Ektron.Cms.Commerce.CustomerData 1255
Ektron.Cms.Commerce.CustomerData.AverageOrderValue 1256
Ektron.Cms.Commerce.CustomerData.BillingAddressId 1256
Ektron.Cms.Commerce.CustomerData.DisplayName 1257
Ektron.Cms.Commerce.CustomerData.EmailAddress 1257
Ektron.Cms.Commerce.CustomerData.FirstName 1257
Ektron.Cms.Commerce.CustomerData.Id 1257
Ektron.Cms.Commerce.CustomerData.IsDeleted 1257
Ektron.Cms.Commerce.CustomerData.IsMembershipUser 1257
Ektron.Cms.Commerce.CustomerData.LastName 1258
Ektron.Cms.Commerce.CustomerData.Password 1258
Ektron.Cms.Commerce.CustomerData.ShippingAddressId 1258
Ektron.Cms.Commerce.CustomerData.TotalOrders 1258
Ektron.Cms.Commerce.CustomerData.TotalOrderValue 1258
Ektron.Cms.Commerce.CustomerData.UserId 1258
Ektron.Cms.Commerce.CustomerData.UserName 1259
Ektron.Cms.Commerce.CustomerProperty 1358
Ektron.Cms.Commerce.CustomerProperty enumeration 1358
Ektron.Cms.Commerce.Dimensions 1259
Ektron.Cms.Commerce.Dimensions.Clone 1260
Ektron.Cms.Commerce.Dimensions.CompareTo 1260
Ektron.Cms.Commerce.Dimensions.ConvertUnits 1261
Ektron.Cms.Commerce.Dimensions.CubicUnits 1261
Ektron.Cms.Commerce.Dimensions.Dimensions 1259, 1260
Ektron.Cms.Commerce.Dimensions.Height 1261
Ektron.Cms.Commerce.Dimensions.Length 1261
Ektron.Cms.Commerce.Dimensions.Units 1262
Ektron.Cms.Commerce.Dimensions.Width 1262
Ektron.Cms.Commerce.EntryAttributeData 1262
Ektron.Cms.Commerce.EntryAttributeData.AttributeTypeId 1264
Ektron.Cms.Commerce.EntryAttributeData.CurrentValue 1264
Ektron.Cms.Commerce.EntryAttributeData.EntryAttributeData 1263
Ektron.Cms.Commerce.EntryAttributeData.VersionId 1264
Ektron.Cms.Commerce.EntryAttributeProperty 1359
Ektron.Cms.Commerce.EntryAttributeProperty enumeration 1359
Ektron.Cms.Commerce.EntryData 1264
Ektron.Cms.Commerce.EntryData.Attributes 1267
Ektron.Cms.Commerce.EntryData.Comment 1267
Ektron.Cms.Commerce.EntryData.ContentStatus 1267
Ektron.Cms.Commerce.EntryData.CurrencyId 1267
Ektron.Cms.Commerce.EntryData.DateModified 1268
Ektron.Cms.Commerce.EntryData.Dimensions 1268
Ektron.Cms.Commerce.EntryData.DisableInventoryManagement 1268
Ektron.Cms.Commerce.EntryData.EndDate 1268
Ektron.Cms.Commerce.EntryData.EndDateAction 1268
Ektron.Cms.Commerce.EntryData.EntryData 1265
Ektron.Cms.Commerce.EntryData.EntryType 1269
Ektron.Cms.Commerce.EntryData.FolderId 1269
Ektron.Cms.Commerce.EntryData.GoLive 1269
Ektron.Cms.Commerce.EntryData.Html 1269
Ektron.Cms.Commerce.EntryData.Id 1269
Ektron.Cms.Commerce.EntryData.Image 1270
Ektron.Cms.Commerce.EntryData.ImageThumbnail 1270
Ektron.Cms.Commerce.EntryData.IsArchived 1270
Ektron.Cms.Commerce.EntryData.IsBuyable 1270
Ektron.Cms.Commerce.EntryData.IsMarkedForDeletion 1271
Ektron.Cms.Commerce.EntryData.IsPublished 1271
Ektron.Cms.Commerce.EntryData.IsTangible 1271
Ektron.Cms.Commerce.EntryData.LanguageId 1271
Ektron.Cms.Commerce.EntryData.LastEditorFirstName 1271
Ektron.Cms.Commerce.EntryData.LastEditorId 1272
Ektron.Cms.Commerce.EntryData.LastEditorLastName 1272
Ektron.Cms.Commerce.EntryData.ListPrice 1272

Ektron.Cms.Commerce.EntryData.Media 1272
Ektron.Cms.Commerce.EntryData.Metadata 1273
Ektron.Cms.Commerce.EntryData.NumberSold 1273
Ektron.Cms.Commerce.EntryData.Pricing 1273
Ektron.Cms.Commerce.EntryData.ProductType 1273
Ektron.Cms.Commerce.EntryData.QuantityMultiple 1273
Ektron.Cms.Commerce.EntryData.Quicklink 1274
Ektron.Cms.Commerce.EntryData.SalePrice 1274
Ektron.Cms.Commerce.EntryData.Sku 1274
Ektron.Cms.Commerce.EntryData.Status 1274
Ektron.Cms.Commerce.EntryData.StatusLanguage 1275
Ektron.Cms.Commerce.EntryData.Summary 1275
Ektron.Cms.Commerce.EntryData.TaxClassId 1275
Ektron.Cms.Commerce.EntryData.TemplateId 1275
Ektron.Cms.Commerce.EntryData.Title 1275
Ektron.Cms.Commerce.EntryData.Url 1276
Ektron.Cms.Commerce.EntryData.VersionId 1276
Ektron.Cms.Commerce.EntryData.Weight 1276
Ektron.Cms.Commerce.EntryGroupData 1276
Ektron.Cms.Commerce.EntryGroupData.Description 1278
Ektron.Cms.Commerce.EntryGroupData.DisplayOrder 1278
Ektron.Cms.Commerce.EntryGroupData.EntryGroupData 1277
Ektron.Cms.Commerce.EntryGroupData.EntryId 1278
Ektron.Cms.Commerce.EntryGroupData.GroupType 1278
Ektron.Cms.Commerce.EntryGroupData.Id 1278
Ektron.Cms.Commerce.EntryGroupData.Name 1278
Ektron.Cms.Commerce.EntryGroupItemData 1279
Ektron.Cms.Commerce.EntryGroupItemData.DisplayOrder 1280
Ektron.Cms.Commerce.EntryGroupItemData.EntryGroupItemData 1279
Ektron.Cms.Commerce.EntryGroupItemData.EntryId 1280
Ektron.Cms.Commerce.EntryGroupItemData.GroupId 1280
Ektron.Cms.Commerce.EntryGroupItemData.Id 1281
Ektron.Cms.Commerce.EntryGroupItemProperty 1359
Ektron.Cms.Commerce.EntryGroupItemProperty enumeration 1359
Ektron.Cms.Commerce.EntryGroupProperty 1359
Ektron.Cms.Commerce.EntryGroupProperty enumeration 1359
Ektron.Cms.Commerce.EntryProperty 1360
Ektron.Cms.Commerce.EntryProperty enumeration 1360
Ektron.Cms.Commerce.ExchangeRateApi 605
Ektron.Cms.Commerce.ExchangeRateApi.Add 606
Ektron.Cms.Commerce.ExchangeRateApi.ConvertPrice 606
Ektron.Cms.Commerce.ExchangeRateApi.Delete 606
Ektron.Cms.Commerce.ExchangeRateApi.GetCurrentExchangeRate 606
Ektron.Cms.Commerce.ExchangeRateApi.GetCurrentList 607
Ektron.Cms.Commerce.ExchangeRateApi.GetList 607
Ektron.Cms.Commerce.ExchangeRateApi.Update 607
Ektron.Cms.Commerce.ExchangeRateData 1281
Ektron.Cms.Commerce.ExchangeRateData.BaseCurrencyId 1282
Ektron.Cms.Commerce.ExchangeRateData.ConvertPrice 1281
Ektron.Cms.Commerce.ExchangeRateData.EffectiveDate 1282
Ektron.Cms.Commerce.ExchangeRateData.ExchangeCurrencyId 1283
Ektron.Cms.Commerce.ExchangeRateData.ExchangeRateData 1281
Ektron.Cms.Commerce.ExchangeRateData.Rate 1283
Ektron.Cms.Commerce.ExchangeRateData.Validate 1282
Ektron.Cms.Commerce.ExchangeRateProperty 1360
Ektron.Cms.Commerce.ExchangeRateProperty enumeration 1360
Ektron.Cms.Commerce.InventoryApi 608
Ektron.Cms.Commerce.InventoryApi.SaveInventory 608
Ektron.Cms.Commerce.InventoryData 1284
Ektron.Cms.Commerce.InventoryData.EntryId 1284
Ektron.Cms.Commerce.InventoryData.ReorderLevel 1285
Ektron.Cms.Commerce.InventoryData.UnitsInStock 1285
Ektron.Cms.Commerce.InventoryData.UnitsOnOrder 1285
Ektron.Cms.Commerce.KitConfigData 1285
Ektron.Cms.Commerce.KitConfigData.BasketItemId 1286
Ektron.Cms.Commerce.KitConfigData.GetGroupIndex 1286
Ektron.Cms.Commerce.KitConfigData.Groups 1286
Ektron.Cms.Commerce.KitConfigData.Id 1287
Ektron.Cms.Commerce.KitConfigData.KitConfigData 1285
Ektron.Cms.Commerce.KitConfigData.PriceModifier 1287
Ektron.Cms.Commerce.KitData 1287
Ektron.Cms.Commerce.KitData.KitData 1287

Ektron.Cms.Commerce.KitData.OptionGroups 1288
Ektron.Cms.Commerce.LinearUnit 1360
Ektron.Cms.Commerce.LinearUnit enumeration 1360
Ektron.Cms.Commerce.OptionGroupData 1292
Ektron.Cms.Commerce.OptionGroupData.Image 1293
Ektron.Cms.Commerce.OptionGroupData.OptionGroupData 1292, 1293
Ektron.Cms.Commerce.OptionGroupData.Options 1293
Ektron.Cms.Commerce.OptionGroupItemCollection 1293
Ektron.Cms.Commerce.OptionGroupItemCollection.AddRange 1294
Ektron.Cms.Commerce.OptionGroupItemCollection.Find 1294
Ektron.Cms.Commerce.OptionGroupItemCollection.FindAll 1295
Ektron.Cms.Commerce.OptionGroupItemCollection.FindIndex 1295
Ektron.Cms.Commerce.OptionGroupItemCollection.OptionGroupItemCollection 1294
Ektron.Cms.Commerce.OptionGroupItemData 1295
Ektron.Cms.Commerce.OptionGroupItemData.CurrencyId 1297
Ektron.Cms.Commerce.OptionGroupItemData.ExtraText 1297
Ektron.Cms.Commerce.OptionGroupItemData.Name 1297
Ektron.Cms.Commerce.OptionGroupItemData.OptionGroupItemData 1296
Ektron.Cms.Commerce.OptionGroupItemData.PriceModification 1297
Ektron.Cms.Commerce.OptionGroupItemData.PriceModificationType 1297
Ektron.Cms.Commerce.OrderApi 608
Ektron.Cms.Commerce.OrderApi.Capture 609, 610
Ektron.Cms.Commerce.OrderApi.GetCustomerOrderList 610
Ektron.Cms.Commerce.OrderApi.GetItem 610
Ektron.Cms.Commerce.OrderApi.GetList 611
Ektron.Cms.Commerce.OrderApi.GetOnHoldOrderList 612
Ektron.Cms.Commerce.OrderApi.GetOrderMessage 613
Ektron.Cms.Commerce.OrderApi.GetOrderPayment 613
Ektron.Cms.Commerce.OrderApi.GetOrderStatus 614
Ektron.Cms.Commerce.OrderApi.GetProductOrderList 614
Ektron.Cms.Commerce.OrderApi.GetReport 615
Ektron.Cms.Commerce.OrderApi.ItemsInStock 615
Ektron.Cms.Commerce.OrderApi.MarkAsFraud 615
Ektron.Cms.Commerce.OrderApi.ProcessOrderInventory 615
Ektron.Cms.Commerce.OrderApi.RollbackOrderInventory 616
Ektron.Cms.Commerce.OrderApi.SetOrderStatus 616
Ektron.Cms.Commerce.OrderApi.SetOrderToHold 616
Ektron.Cms.Commerce.OrderApi.SetTrackingNumber 616
Ektron.Cms.Commerce.OrderData 1298
Ektron.Cms.Commerce.OrderData.BillingAddressId 1299
Ektron.Cms.Commerce.OrderData.ClientIP 1299
Ektron.Cms.Commerce.OrderData.CouponTotal 1299
Ektron.Cms.Commerce.OrderData.CouponUsed 1299
Ektron.Cms.Commerce.OrderData.Currency 1299
Ektron.Cms.Commerce.OrderData.Customer 1300
Ektron.Cms.Commerce.OrderData.CustomerType 1300
Ektron.Cms.Commerce.OrderData.DateCompleted 1300
Ektron.Cms.Commerce.OrderData.DateCreated 1300
Ektron.Cms.Commerce.OrderData.DateRequired 1300
Ektron.Cms.Commerce.OrderData.HandlingTotal 1301
Ektron.Cms.Commerce.OrderData.HttpHost 1301
Ektron.Cms.Commerce.OrderData.Id 1301
Ektron.Cms.Commerce.OrderData.OrderTotal 1301
Ektron.Cms.Commerce.OrderData.Parts 1301
Ektron.Cms.Commerce.OrderData.ShippingTotal 1301
Ektron.Cms.Commerce.OrderData.SpecialInstructions 1302
Ektron.Cms.Commerce.OrderData.StageId 1302
Ektron.Cms.Commerce.OrderData.StageName 1302
Ektron.Cms.Commerce.OrderData.Status 1302
Ektron.Cms.Commerce.OrderData.Subtotal 1302
Ektron.Cms.Commerce.OrderData.TaxTotal 1303
Ektron.Cms.Commerce.OrderData.VisitorId 1303
Ektron.Cms.Commerce.OrderLineData 1303
Ektron.Cms.Commerce.OrderLineData.Configuration 1304
Ektron.Cms.Commerce.OrderLineData.ItemType 1304
Ektron.Cms.Commerce.OrderLineData.LineItemId 1304
Ektron.Cms.Commerce.OrderLineData.OrderPartId 1304
Ektron.Cms.Commerce.OrderLineData.PriceEach 1304
Ektron.Cms.Commerce.OrderLineData.PriceSubTotal 1305
Ektron.Cms.Commerce.OrderLineData.PriceTotal 1305
Ektron.Cms.Commerce.OrderLineData.ProductId 1305
Ektron.Cms.Commerce.OrderLineData.ProductLanguage 1305

Ektron.Cms.Commerce.OrderLineData.Quantity 1305
Ektron.Cms.Commerce.OrderPartData 1306
Ektron.Cms.Commerce.OrderPartData.DateShipped 1306
Ektron.Cms.Commerce.OrderPartData.HandlingTotal 1307
Ektron.Cms.Commerce.OrderPartData.Lines 1307
Ektron.Cms.Commerce.OrderPartData.OrderId 1307
Ektron.Cms.Commerce.OrderPartData.PartId 1307
Ektron.Cms.Commerce.OrderPartData.PartTotal 1307
Ektron.Cms.Commerce.OrderPartData.ShippingAddressId 1308
Ektron.Cms.Commerce.OrderPartData.ShippingTotal 1308
Ektron.Cms.Commerce.OrderPartData.Status 1308
Ektron.Cms.Commerce.OrderPartData.Subtotal 1308
Ektron.Cms.Commerce.OrderPartData.SupplierId 1308
Ektron.Cms.Commerce.OrderPartData.TaxTotal 1309
Ektron.Cms.Commerce.OrderPartData.TrackingNumber 1309
Ektron.Cms.Commerce.OrderProperty 1360
Ektron.Cms.Commerce.OrderProperty enumeration 1360
Ektron.Cms.Commerce.PackageApi 617
Ektron.Cms.Commerce.PackageApi.Add 617
Ektron.Cms.Commerce.PackageApi.Delete 618
Ektron.Cms.Commerce.PackageApi.GetItem 618
Ektron.Cms.Commerce.PackageApi.GetList 618
Ektron.Cms.Commerce.PackageApi.Update 618
Ektron.Cms.Commerce.PackageData 1309
Ektron.Cms.Commerce.PackageData.Dimensions 1311
Ektron.Cms.Commerce.PackageData.Id 1311
Ektron.Cms.Commerce.PackageData.MaxWeight 1311
Ektron.Cms.Commerce.PackageData.Name 1311
Ektron.Cms.Commerce.PackageData.PackageData 1310
Ektron.Cms.Commerce.PackageProperty 1361
Ektron.Cms.Commerce.PackageProperty enumeration 1361
Ektron.Cms.Commerce.PaymentAuthorization 1311
Ektron.Cms.Commerce.PaymentAuthorization.AuthorizedOn 1312
Ektron.Cms.Commerce.PaymentAuthorization.CapturedOn 1312
Ektron.Cms.Commerce.PaymentAuthorization.CreditedOn 1312
Ektron.Cms.Commerce.PaymentAuthorization.Gateway 1312
Ektron.Cms.Commerce.PaymentAuthorization.Message 1313
Ektron.Cms.Commerce.PaymentAuthorization.RecurringTransactionId 1313
Ektron.Cms.Commerce.PaymentAuthorization.TransactionId 1313
Ektron.Cms.Commerce.PaymentAuthorization.VoidedOn 1313
Ektron.Cms.Commerce.PipelineStageData 1313
Ektron.Cms.Commerce.PipelineStageData.Name 1314
Ektron.Cms.Commerce.PipelineStageData.Sequence 1314
Ektron.Cms.Commerce.PipelineStageData.StatusId 1314
Ektron.Cms.Commerce.PostalCodeTaxRateData 1314
Ektron.Cms.Commerce.PostalCodeTaxRateData.PostalCode 1316
Ektron.Cms.Commerce.PostalCodeTaxRateData.PostalCodeTaxRateData 1315, 1316
Ektron.Cms.Commerce.PostalCodeTaxRateData.RegionId 1316
Ektron.Cms.Commerce.PricingData 1316
Ektron.Cms.Commerce.PricingData.CurrencyPricelist 1320
Ektron.Cms.Commerce.PricingData.GetCurrencyById 1318
Ektron.Cms.Commerce.PricingData.GetCurrencyByIso 1318
Ektron.Cms.Commerce.PricingData.GetCurrencyByIsoNumber 1319
Ektron.Cms.Commerce.PricingData.GetListPrice 1319
Ektron.Cms.Commerce.PricingData.GetListPriceByCurrencyId 1319
Ektron.Cms.Commerce.PricingData.GetListPriceByIsoNumber 1319
Ektron.Cms.Commerce.PricingData.GetSalePrice 1319
Ektron.Cms.Commerce.PricingData.GetSalePriceByCurrencyId 1320
Ektron.Cms.Commerce.PricingData.GetSalePriceByIsoNumber 1320
Ektron.Cms.Commerce.PricingData.IsRecurringPrice 1321
Ektron.Cms.Commerce.PricingData.PricingData 1317
Ektron.Cms.Commerce.PricingData.Recurrence 1321
Ektron.Cms.Commerce.PricingData.Validate 1320
Ektron.Cms.Commerce.ProductCouponEntryData 1349
Ektron.Cms.Commerce.ProductCouponEntryData.EntryType 1351
Ektron.Cms.Commerce.ProductCouponEntryData.ProductCouponEntryData 1350
Ektron.Cms.Commerce.ProductData 1321
Ektron.Cms.Commerce.ProductData.ProductData 1321, 1322

Ektron.Cms.Commerce.ProductData.Variants 1322
Ektron.Cms.Commerce.ProductTypeApi 619
Ektron.Cms.Commerce.ProductTypeApi.Add 619
Ektron.Cms.Commerce.ProductTypeApi.AddProductAttribute 620
Ektron.Cms.Commerce.ProductTypeApi.Delete 620
Ektron.Cms.Commerce.ProductTypeApi.DeleteProductAttribute 620, 621
Ektron.Cms.Commerce.ProductTypeApi.GetFolderProductTypeList 621
Ektron.Cms.Commerce.ProductTypeApi.GetItem 621
Ektron.Cms.Commerce.ProductTypeApi.GetProductAttribute 622
Ektron.Cms.Commerce.ProductTypeApi.GetProductAttributeList 622
Ektron.Cms.Commerce.ProductTypeApi.Update 622
Ektron.Cms.Commerce.ProductTypeApi.UpdateProductAttribute 622
Ektron.Cms.Commerce.ProductTypeAttributeData 1322
Ektron.Cms.Commerce.ProductTypeAttributeData.ActiveStatus 1324
Ektron.Cms.Commerce.ProductTypeAttributeData.DataType 1324
Ektron.Cms.Commerce.ProductTypeAttributeData.DefaultValue 1324
Ektron.Cms.Commerce.ProductTypeAttributeData.DisplayOrder 1324
Ektron.Cms.Commerce.ProductTypeAttributeData.Id 1324
Ektron.Cms.Commerce.ProductTypeAttributeData.Name 1325
Ektron.Cms.Commerce.ProductTypeAttributeData.ProductTypeAttributeData 1323
Ektron.Cms.Commerce.ProductTypeAttributeProperty 1361
Ektron.Cms.Commerce.ProductTypeAttributeProperty enumeration 1361
Ektron.Cms.Commerce.ProductTypeData 1325
Ektron.Cms.Commerce.ProductTypeData.Attributes 1325
Ektron.Cms.Commerce.ProductTypeData.DefaultThumbnails 1326
Ektron.Cms.Commerce.ProductTypeData.EntryClass 1326
Ektron.Cms.Commerce.ProductTypeProperty 1361
Ektron.Cms.Commerce.ProductTypeProperty enumeration 1361
Ektron.Cms.Commerce.RecommendationApi 622
Ektron.Cms.Commerce.RecommendationApi.Add 623
Ektron.Cms.Commerce.RecommendationApi.Delete 623
Ektron.Cms.Commerce.RecommendationApi.DeleteRecommendationByEntry 624
Ektron.Cms.Commerce.RecommendationApi.GetItem 624
Ektron.Cms.Commerce.RecommendationApi.Update 624
Ektron.Cms.Commerce.RecommendationApi.UpdateCrossSell 624
Ektron.Cms.Commerce.RecommendationApi.UpdateUpSell 625
Ektron.Cms.Commerce.RecommendationItemData 1326
Ektron.Cms.Commerce.RecommendationItemData.EntryLanguage 1328
Ektron.Cms.Commerce.RecommendationItemData.EntryType 1328
Ektron.Cms.Commerce.RecommendationItemData.Html 1328
Ektron.Cms.Commerce.RecommendationItemData.Image 1328
Ektron.Cms.Commerce.RecommendationItemData.ImageThumbnail 1329
Ektron.Cms.Commerce.RecommendationItemData.Pricing 1329
Ektron.Cms.Commerce.RecommendationItemData.Quicklink 1329
Ektron.Cms.Commerce.RecommendationItemData.RecommendationItemData 1327
Ektron.Cms.Commerce.RecommendationItemData.Summary 1329
Ektron.Cms.Commerce.RecommendationItemData.Title 1329
Ektron.Cms.Commerce.RecommendationItemData.Type 1330
Ektron.Cms.Commerce.RegionApi 625
Ektron.Cms.Commerce.RegionApi.Add 625
Ektron.Cms.Commerce.RegionApi.CanDelete 626
Ektron.Cms.Commerce.RegionApi.Delete 626
Ektron.Cms.Commerce.RegionApi.GetItem 626
Ektron.Cms.Commerce.RegionApi.GetList 626
Ektron.Cms.Commerce.RegionApi.Update 627
Ektron.Cms.Commerce.RegionData 1330
Ektron.Cms.Commerce.RegionData.Code 1332
Ektron.Cms.Commerce.RegionData.CountryId 1332
Ektron.Cms.Commerce.RegionData.Enabled 1332
Ektron.Cms.Commerce.RegionData.Id 1332

Ektron.Cms.Commerce.RegionData.Name 1332
Ektron.Cms.Commerce.RegionData.RegionData 1330, 1331
Ektron.Cms.Commerce.RegionData.ToString 1331
Ektron.Cms.Commerce.RegionData.Validate 1331
Ektron.Cms.Commerce.RegionProperty 1362
Ektron.Cms.Commerce.RegionProperty enumeration 1362
Ektron.Cms.Commerce.RegionTaxRateData 1333
Ektron.Cms.Commerce.RegionTaxRateData.RegionData 1334
Ektron.Cms.Commerce.RegionTaxRateData.RegionTaxRateData 1333, 1334
Ektron.Cms.Commerce.ShippingBaseData 1334
Ektron.Cms.Commerce.ShippingBaseData.Id 1335
Ektron.Cms.Commerce.ShippingBaseData.Name 1335
Ektron.Cms.Commerce.ShippingBaseData.ProviderService 1335
Ektron.Cms.Commerce.ShippingBaseData.ShippingBaseData 1334
Ektron.Cms.Commerce.ShippingMethodApi 627
Ektron.Cms.Commerce.ShippingMethodApi.Add 627
Ektron.Cms.Commerce.ShippingMethodApi.Update 628
Ektron.Cms.Commerce.ShippingMethodData 1355
Ektron.Cms.Commerce.ShippingMethodData.DisplayOrder 1356
Ektron.Cms.Commerce.ShippingMethodData.IsActive 1356
Ektron.Cms.Commerce.ShippingMethodData.ShippingMethodData 1356
Ektron.Cms.Commerce.ShippingMethodProperty 1362
Ektron.Cms.Commerce.ShippingMethodProperty enumeration 1362
Ektron.Cms.Commerce.ShippingOptionData 1335
Ektron.Cms.Commerce.ShippingOptionData.CurrencyId 1336
Ektron.Cms.Commerce.ShippingOptionData.ShippingFee 1336
Ektron.Cms.Commerce.ShippingOptionData.ShippingOptionData 1336
Ektron.Cms.Commerce.TaxApi 628
Ektron.Cms.Commerce.TaxApi.Add 629
Ektron.Cms.Commerce.TaxApi.Delete 629
Ektron.Cms.Commerce.TaxApi.GetApplicableTaxRate 629
Ektron.Cms.Commerce.TaxApi.GetApplicableTaxRateList 629
Ektron.Cms.Commerce.TaxApi.GetItem 630
Ektron.Cms.Commerce.TaxApi.GetItemByCountryId 630
Ektron.Cms.Commerce.TaxApi.GetItemByPostalCode 631
Ektron.Cms.Commerce.TaxApi.GetItemByPostalId 632
Ektron.Cms.Commerce.TaxApi.GetItemByRegionId 632
Ektron.Cms.Commerce.TaxApi.GetList 632
Ektron.Cms.Commerce.TaxApi.Update 632
Ektron.Cms.Commerce.TaxClassApi 633
Ektron.Cms.Commerce.TaxClassApi.Add 633
Ektron.Cms.Commerce.TaxClassApi.Delete 633
Ektron.Cms.Commerce.TaxClassApi.GetItem 634
Ektron.Cms.Commerce.TaxClassApi.GetList 634
Ektron.Cms.Commerce.TaxClassApi.Update 634
Ektron.Cms.Commerce.TaxClassData 1336
Ektron.Cms.Commerce.TaxClassData.Id 1338
Ektron.Cms.Commerce.TaxClassData.Name 1338
Ektron.Cms.Commerce.TaxClassData.TaxClassData 1337, 1338
Ektron.Cms.Commerce.TaxClassProperty 1362
Ektron.Cms.Commerce.TaxClassProperty enumeration 1362
Ektron.Cms.Commerce.TaxRateData 1338
Ektron.Cms.Commerce.TaxRateData.Id 1340
Ektron.Cms.Commerce.TaxRateData.Rate 1340
Ektron.Cms.Commerce.TaxRateData.TaxClassId 1340
Ektron.Cms.Commerce.TaxRateData.TaxRateData 1339
Ektron.Cms.Commerce.TaxRateData.TypeId 1340
Ektron.Cms.Commerce.TaxRateData.TypeItemId 1341
Ektron.Cms.Commerce.TaxRateProperty 1363
Ektron.Cms.Commerce.TaxRateProperty enumeration 1363
Ektron.Cms.Commerce.TaxRateType 1363
Ektron.Cms.Commerce.TaxRateType enumeration 1363
Ektron.Cms.Commerce.TaxTypeApi 635
Ektron.Cms.Commerce.TaxTypeApi.Add 635
Ektron.Cms.Commerce.TaxTypeApi.Delete 635
Ektron.Cms.Commerce.TaxTypeApi.GetItem 636
Ektron.Cms.Commerce.TaxTypeApi.GetList 636
Ektron.Cms.Commerce.TaxTypeApi.Update 636
Ektron.Cms.Commerce.TaxTypeData 1341
Ektron.Cms.Commerce.TaxTypeData.Id 1342
Ektron.Cms.Commerce.TaxTypeData.Name 1343
Ektron.Cms.Commerce.TaxTypeData.Precedence 1343

Ektron.Cms.Commerce.TaxTypeData.TaxTypeData 1341, 1342
Ektron.Cms.Commerce.TaxTypeProperty 1363
Ektron.Cms.Commerce.TaxTypeProperty enumeration 1363
Ektron.Cms.Commerce.TierPriceData 1346
Ektron.Cms.Commerce.TierPriceData.Clone 1347
Ektron.Cms.Commerce.TierPriceData.CompareTo 1348
Ektron.Cms.Commerce.TierPriceData.DataState 1348
Ektron.Cms.Commerce.TierPriceData.Id 1348
Ektron.Cms.Commerce.TierPriceData.Quantity 1349
Ektron.Cms.Commerce.TierPriceData.RecurrenceId 1349
Ektron.Cms.Commerce.TierPriceData.SalePrice 1349
Ektron.Cms.Commerce.TierPriceData.TierPriceData 1347
Ektron.Cms.Commerce.TierPriceData.Validate 1348
Ektron.Cms.Commerce.WarehouseApi 636
Ektron.Cms.Commerce.WarehouseApi.Add 637
Ektron.Cms.Commerce.WarehouseApi.Delete 637
Ektron.Cms.Commerce.WarehouseApi.GetDefaultWarehouse 637
Ektron.Cms.Commerce.WarehouseApi.GetItem 638
Ektron.Cms.Commerce.WarehouseApi.GetList 638
Ektron.Cms.Commerce.WarehouseApi.MarkAsDefault 638
Ektron.Cms.Commerce.WarehouseApi.Update 638
Ektron.Cms.Commerce.WarehouseData 1351
Ektron.Cms.Commerce.WarehouseData.Address 1352
Ektron.Cms.Commerce.WarehouseData.Id 1352
Ektron.Cms.Commerce.WarehouseData.IsDefaultWarehouse 1352
Ektron.Cms.Commerce.WarehouseData.Name 1353
Ektron.Cms.Commerce.WarehouseData.WarehouseData 1351, 1352
Ektron.Cms.Commerce.WarehouseProperty 1364
Ektron.Cms.Commerce.WarehouseProperty enumeration 1364
Ektron.Cms.Commerce.Weight 1353
Ektron.Cms.Commerce.Weight.Amount 1355
Ektron.Cms.Commerce.Weight.Clone 1354
Ektron.Cms.Commerce.Weight.CompareTo 1354
Ektron.Cms.Commerce.Weight.ConvertUnits 1354
Ektron.Cms.Commerce.Weight.Units 1355
Ektron.Cms.Commerce.Weight.Weight 1353, 1354
Ektron.Cms.Commerce.WeightUnit 1364
Ektron.Cms.Commerce.WeightUnit enumeration 1364
Ektron.Cms.Common 1364
Ektron.Cms.Common namespace 1364
Classes 1364
Ektron.Cms.Common.UrlAliasAutoData 1365
Ektron.Cms.Common.UrlAliasAutoData.AliasName 1366
Ektron.Cms.Common.UrlAliasAutoData.AutoAliasType 1366
Ektron.Cms.Common.UrlAliasAutoData.AutoId 1366
Ektron.Cms.Common.UrlAliasAutoData.Example 1366
Ektron.Cms.Common.UrlAliasAutoData.ExcludedPath 1367
Ektron.Cms.Common.UrlAliasAutoData.FileExtension 1367
Ektron.Cms.Common.UrlAliasAutoData.FormattedSelectedPath 1367
Ektron.Cms.Common.UrlAliasAutoData.HostName 1367
Ektron.Cms.Common.UrlAliasAutoData.IsEnabled 1367
Ektron.Cms.Common.UrlAliasAutoData.LanguageId 1368
Ektron.Cms.Common.UrlAliasAutoData.PageNameType 1368
Ektron.Cms.Common.UrlAliasAutoData.ReplacementCharacter 1368
Ektron.Cms.Common.UrlAliasAutoData.SelectedPath 1368
Ektron.Cms.Common.UrlAliasAutoData.SiteId 1369
Ektron.Cms.Common.UrlAliasAutoData.SourceId 1369
Ektron.Cms.Common.UrlAliasAutoData.SourceName 1369
Ektron.Cms.Common.UrlAliasAutoData.Target 1369
Ektron.Cms.Common.UrlAliasManualData 1370
Ektron.Cms.Common.UrlAliasManualData.AliasId 1370
Ektron.Cms.Common.UrlAliasManualData.AliasName 1371
Ektron.Cms.Common.UrlAliasManualData.AliasPageName 1371
Ektron.Cms.Common.UrlAliasManualData.ContentId 1371
Ektron.Cms.Common.UrlAliasManualData.ContentLanguage 1371
Ektron.Cms.Common.UrlAliasManualData.ContentTitle 1372
Ektron.Cms.Common.UrlAliasManualData.DisplayAlias 1372
Ektron.Cms.Common.UrlAliasManualData.FileExtension 1372
Ektron.Cms.Common.UrlAliasManualData.HostName 1372
Ektron.Cms.Common.UrlAliasManualData.IsDefault 1373
Ektron.Cms.Common.UrlAliasManualData.IsEnabled 1373
Ektron.Cms.Common.UrlAliasManualData.LibraryId 1373
Ektron.Cms.Common.UrlAliasManualData.QueryString 1373
Ektron.Cms.Common.UrlAliasManualData.QueryStringAction 1373

Ektron.Cms.Common.UrlAliasManualData.SiteId 1374
Ektron.Cms.Common.UrlAliasManualData.Target 1374
Ektron.Cms.Common.UrlAliasRegExData 1374
Ektron.Cms.Common.UrlAliasRegExData.Expression 1375
Ektron.Cms.Common.UrlAliasRegExData.ExpressionMap 1375
Ektron.Cms.Common.UrlAliasRegExData.ExpressionName 1375
Ektron.Cms.Common.UrlAliasRegExData.IsEnabled 1376
Ektron.Cms.Common.UrlAliasRegExData.Priority 1376
Ektron.Cms.Common.UrlAliasRegExData.RegExId 1376
Ektron.Cms.Common.UrlAliasRegExData.SiteId 1377
Ektron.Cms.Common.UrlAliasRegExData.TransformedUrl 1377
Ektron.Cms.CommunityUserData 1377
Ektron.Cms.CommunityUserData.Tags 1378
Ektron.Cms.ContentData 1378
Ektron.Cms.ContentData.ApprovalMethod 1380
Ektron.Cms.ContentData.Approver 1381
Ektron.Cms.ContentData.Comment 1381
Ektron.Cms.ContentData.ContType 1381
Ektron.Cms.ContentData.DateCreated 1382
Ektron.Cms.ContentData.DisplayDateCreated 1382
Ektron.Cms.ContentData.DisplayEndDate 1382
Ektron.Cms.ContentData.DisplayGoLive 1383
Ektron.Cms.ContentData.DisplayLastEditDate 1383
Ektron.Cms.ContentData.EditorFirstName 1383
Ektron.Cms.ContentData.EditorLastName 1383
Ektron.Cms.ContentData.EditorUserNames 1384
Ektron.Cms.ContentData.EndDate 1384
Ektron.Cms.ContentData.EndDateAction 1384
Ektron.Cms.ContentData.FolderId 1385
Ektron.Cms.ContentData.FolderName 1385
Ektron.Cms.ContentData.GoLive 1385
Ektron.Cms.ContentData.HistoryId 1385
Ektron.Cms.ContentData.HyperLink 1386
Ektron.Cms.ContentData.InheritedFrom 1386
Ektron.Cms.ContentData.IsInherited 1386
Ektron.Cms.ContentData.IsPublished 1386
Ektron.Cms.ContentData.IsSearchable 1387
Ektron.Cms.ContentData.IsXmlInherited 1387
Ektron.Cms.ContentData.LastEditDate 1387
Ektron.Cms.ContentData.ManualAlias 1387
Ektron.Cms.ContentData.ManualAliasId 1388
Ektron.Cms.ContentData.MetaData 1388
Ektron.Cms.ContentData.Path 1388
Ektron.Cms.ContentData.Permissions 1389
Ektron.Cms.ContentData.Status 1389
Ektron.Cms.ContentData.StyleSheet 1389
Ektron.Cms.ContentData.TemplateConfiguration 1390
Ektron.Cms.ContentData.UserId 1390
Ektron.Cms.ContentData.UserName 1390
Ektron.Cms.ContentData.XmlConfiguration 1390
Ektron.Cms.ContentData.XmlInheritedFrom 1391
Ektron.Cms.ContentRatingLabels 1391
Ektron.Cms.ContentRatingLabels.AlreadyRatedMessage 1392
Ektron.Cms.ContentRatingLabels.BadLabel 1392
Ektron.Cms.ContentRatingLabels.CharsRemainingLabel 1392
Ektron.Cms.ContentRatingLabels.CookiesRequiredMessage 1392
Ektron.Cms.ContentRatingLabels.FeedbackHeader 1393
Ektron.Cms.ContentRatingLabels.GoodLabel 1393
Ektron.Cms.ContentRatingLabels.RatingLevelLabel 1393
Ektron.Cms.ContentRatingLabels.ThankYouMessage 1393
Ektron.Cms.ContentRatingLabels.TotalRatingsLabel 1393
Ektron.Cms.ContentRatingLabels.UserCommentsHeader 1394
Ektron.Cms.ContentRatingLabels.ValidationMessage 1394
Ektron.Cms.Controls 7
Ektron.Cms.Controls namespace 7
Classes 11
Ektron.Cms.Controls.ActiveTopics 14
Ektron.Cms.Controls.ActiveTopics.BoardID 15
Ektron.Cms.Controls.ActiveTopics.CacheInterval 16
Ektron.Cms.Controls.ActiveTopics.DisplayMode 16
Ektron.Cms.Controls.ActiveTopics.DynamicForumParameter 17
Ektron.Cms.Controls.ActiveTopics.DynamicGroupParameter 17
Ektron.Cms.Controls.ActiveTopics.DynamicThreadParameter 17
Ektron.Cms.Controls.ActiveTopics.Fill 14
Ektron.Cms.Controls.ActiveTopics.MaxNumber 17
Ektron.Cms.Controls.ActiveTopics.ToString 15
Ektron.Cms.Controls.ActiveTopics.URLPath 18
Ektron.Cms.Controls.ActiveTopics.Wrap 15

Ektron.Cms.Controls.AnalyticsTracker 18
Ektron.Cms.Controls.AnalyticsTracker.DefaultContentID 19
Ektron.Cms.Controls.AnalyticsTracker.DynamicParameter 20
Ektron.Cms.Controls.AnalyticsTracker.EnableAnalytics 20
Ektron.Cms.Controls.AnalyticsTracker.Wrap 19
Ektron.Cms.Controls.AssetControl 20
Ektron.Cms.Controls.AssetControl.DefaultAssetID 22
Ektron.Cms.Controls.AssetControl.DefaultFolderID 22
Ektron.Cms.Controls.AssetControl.DynamicParameter 22
Ektron.Cms.Controls.AssetControl.Fill 21
Ektron.Cms.Controls.AssetControl.IsImage 23
Ektron.Cms.Controls.AssetControl.OverrideExtension 23
Ektron.Cms.Controls.AssetControl.TaxonomyID 24
Ektron.Cms.Controls.AssetControl.ToString 21
Ektron.Cms.Controls.AssetControl.UploadType 24
Ektron.Cms.Controls.Blog 24
Ektron.Cms.Controls.Blog.ArchiveMode 26
Ektron.Cms.Controls.Blog.BlogID 27
Ektron.Cms.Controls.Blog.CacheInterval 27
Ektron.Cms.Controls.Blog.DateToStart 28
Ektron.Cms.Controls.Blog.DefaultUserID 28
Ektron.Cms.Controls.Blog.DynamicParameter 28
Ektron.Cms.Controls.Blog.DynamicUserParameter 29
Ektron.Cms.Controls.Blog.EditorHeight 29
Ektron.Cms.Controls.Blog.EditorWidth 29
Ektron.Cms.Controls.Blog.Fill 25
Ektron.Cms.Controls.Blog.JavascriptEditorHTMLMode 30
Ektron.Cms.Controls.Blog.MaxResults 30
Ektron.Cms.Controls.Blog.PostParameter 30
Ektron.Cms.Controls.Blog.RecentPosts 31
Ektron.Cms.Controls.Blog.ShowHeader 31
Ektron.Cms.Controls.Blog.ShowRSS 31
Ektron.Cms.Controls.Blog.ToString 25
Ektron.Cms.Controls.Blog.Wrap 25
Ektron.Cms.Controls.BlogArchive 32
Ektron.Cms.Controls.BlogArchive.ArchiveMode 33
Ektron.Cms.Controls.BlogArchive.BlogID 33
Ektron.Cms.Controls.BlogArchive.CacheInterval 33
Ektron.Cms.Controls.BlogArchive.DynamicParameter 34
Ektron.Cms.Controls.BlogArchive.Fill 32
Ektron.Cms.Controls.BlogArchive.Text 34
Ektron.Cms.Controls.BlogArchive.Wrap 32
Ektron.Cms.Controls.BlogArchive.WrapTag 34
Ektron.Cms.Controls.BlogCalendar 35
Ektron.Cms.Controls.BlogCalendar.BlogID 36
Ektron.Cms.Controls.BlogCalendar.CacheInterval 36
Ektron.Cms.Controls.BlogCalendar.Fill 35
Ektron.Cms.Controls.BlogCalendar.Text 36
Ektron.Cms.Controls.BlogCalendar.Wrap 35
Ektron.Cms.Controls.BlogCalendar.WrapTag 37
Ektron.Cms.Controls.BlogCategories 37
Ektron.Cms.Controls.BlogCategories.BlogID 38
Ektron.Cms.Controls.BlogCategories.CacheInterval 39
Ektron.Cms.Controls.BlogCategories.Fill 37
Ektron.Cms.Controls.BlogCategories.Text 39
Ektron.Cms.Controls.BlogCategories.Wrap 38
Ektron.Cms.Controls.BlogCategories.WrapTag 39
Ektron.Cms.Controls.BlogEntries 39
Ektron.Cms.Controls.BlogEntries.BlogID 42
Ektron.Cms.Controls.BlogEntries.BlogPostParameter 43
Ektron.Cms.Controls.BlogEntries.CacheInterval 43
Ektron.Cms.Controls.BlogEntries.DateToStart 43
Ektron.Cms.Controls.BlogEntries.DefaultUserID 44
Ektron.Cms.Controls.BlogEntries.DisplayXslt 44
Ektron.Cms.Controls.BlogEntries.DynamicParameter 44
Ektron.Cms.Controls.BlogEntries.DynamicUserParameter 45
Ektron.Cms.Controls.BlogEntries.EditorHeight 45
Ektron.Cms.Controls.BlogEntries.EditorWidth 45
Ektron.Cms.Controls.BlogEntries.Fill 40
Ektron.Cms.Controls.BlogEntries.GetRssFeed 40
Ektron.Cms.Controls.BlogEntries.GetView 41
Ektron.Cms.Controls.BlogEntries.GetViewNames 41
Ektron.Cms.Controls.BlogEntries.JavascriptEditorHTMLMode 46
Ektron.Cms.Controls.BlogEntries.MaxResults 46
Ektron.Cms.Controls.BlogEntries.ShowHeader 46
Ektron.Cms.Controls.BlogEntries.ToString 41
Ektron.Cms.Controls.BlogEntries.Wrap 41
Ektron.Cms.Controls.BlogEntries.XmlDoc 47

Ektron.Cms.Controls.BlogPost 47
Ektron.Cms.Controls.BlogPost.DefaultContentID 48
Ektron.Cms.Controls.BlogPost.ShowType 49
Ektron.Cms.Controls.BlogPost.Wrap 48
Ektron.Cms.Controls.BlogRecentPosts 49
Ektron.Cms.Controls.BlogRecentPosts.BlogID 50
Ektron.Cms.Controls.BlogRecentPosts.CacheInterval 51
Ektron.Cms.Controls.BlogRecentPosts.Fill 50
Ektron.Cms.Controls.BlogRecentPosts.NumberofPosts 51
Ektron.Cms.Controls.BlogRecentPosts.Text 51
Ektron.Cms.Controls.BlogRecentPosts.Wrap 50
Ektron.Cms.Controls.BlogRecentPosts.WrapTag 51
Ektron.Cms.Controls.BlogRoll 52
Ektron.Cms.Controls.BlogRoll.BlogID 53
Ektron.Cms.Controls.BlogRoll.CacheInterval 53
Ektron.Cms.Controls.BlogRoll.Fill 52
Ektron.Cms.Controls.BlogRoll.Text 54
Ektron.Cms.Controls.BlogRoll.Wrap 52
Ektron.Cms.Controls.BlogRoll.WrapTag 54
Ektron.Cms.Controls.BlogRSS 54
Ektron.Cms.Controls.BlogRSS.BlogID 55
Ektron.Cms.Controls.BlogRSS.CacheInterval 55
Ektron.Cms.Controls.BlogRSS.Fill 55
Ektron.Cms.Controls.BlogRSS.Text 56
Ektron.Cms.Controls.BlogRSS.WrapTag 56
Ektron.Cms.Controls.BreadCrumb 56
Ektron.Cms.Controls.BreadCrumb.CurrentPageIndicator 59
Ektron.Cms.Controls.BreadCrumb.DefaultContentID 59
Ektron.Cms.Controls.BreadCrumb.DisplayStyle 60
Ektron.Cms.Controls.BreadCrumb.DisplayTitle 60
Ektron.Cms.Controls.BreadCrumb.DynamicParameter 60
Ektron.Cms.Controls.BreadCrumb.Fill 57
Ektron.Cms.Controls.BreadCrumb.GenerateTrail_HTML 57
Ektron.Cms.Controls.BreadCrumb.IconAlt 60
Ektron.Cms.Controls.BreadCrumb.IconPath 61
Ektron.Cms.Controls.BreadCrumb.LinkLastItem 61
Ektron.Cms.Controls.BreadCrumb.LinkTarget 61
Ektron.Cms.Controls.BreadCrumb.MaxItems 61
Ektron.Cms.Controls.BreadCrumb.MetadataName 62
Ektron.Cms.Controls.BreadCrumb.Mode 62
Ektron.Cms.Controls.BreadCrumb.ResetTrail 58
Ektron.Cms.Controls.BreadCrumb.separator 62
Ektron.Cms.Controls.BreadCrumbItem 63
Ektron.Cms.Controls.BreadCrumbItem.BreadCrumbItem 63
Ektron.Cms.Controls.BreadCrumbItem.CaseSensitiveUrlComparison 65
Ektron.Cms.Controls.BreadCrumbItem.DisplayTitle 65
Ektron.Cms.Controls.BreadCrumbItem.Equals 64
Ektron.Cms.Controls.BreadCrumbItem.GetHashCode 64
Ektron.Cms.Controls.BreadCrumbItem.IconAlt 65
Ektron.Cms.Controls.BreadCrumbItem.IconPath 65
Ektron.Cms.Controls.BreadCrumbItem.RawUrl 66
Ektron.Cms.Controls.BreadCrumbItem.Target 66
Ektron.Cms.Controls.BreadCrumbItem.ToString 64
Ektron.Cms.Controls.BusinessRules 66
Ektron.Cms.Controls.BusinessRules.Fill 67
Ektron.Cms.Controls.BusinessRules.Parameters 67
Ektron.Cms.Controls.BusinessRules.Result 68
Ektron.Cms.Controls.BusinessRules.RuleID 68
Ektron.Cms.Controls.BusinessRules.Text 68
Ektron.Cms.Controls.BusinessRules.Wrap 67
Ektron.Cms.Controls.BusinessRules.WrapTag 69
Ektron.Cms.Controls.Calendar 69
Ektron.Cms.Controls.Calendar.DefaultCalendarID 70
Ektron.Cms.Controls.Calendar.DisplayType 71
Ektron.Cms.Controls.Calendar.DynamicParameter 71
Ektron.Cms.Controls.Calendar.EndDate 72
Ektron.Cms.Controls.Calendar.Fill 69
Ektron.Cms.Controls.Calendar.HighlightEventID 72
Ektron.Cms.Controls.Calendar.HighlightEventTypeID 72
Ektron.Cms.Controls.Calendar.ShowXML 72
Ektron.Cms.Controls.Calendar.StartDate 73
Ektron.Cms.Controls.Calendar.Text 73
Ektron.Cms.Controls.Calendar.Wrap 70
Ektron.Cms.Controls.Cart 73
Ektron.Cms.Controls.Cart.DisplayXslt 74
Ektron.Cms.Controls.Cart.DynamicProductParameter 75
Ektron.Cms.Controls.Cart.EnableCoupons 75

Ektron.Cms.Controls.Cart.LoadingImage 76
Ektron.Cms.Controls.Cart.Stylesheet 76
Ektron.Cms.Controls.Cart.TemplateCheckout 76
Ektron.Cms.Controls.Cart.TemplateProduct 77
Ektron.Cms.Controls.Cart.TemplateShopping 77
Ektron.Cms.Controls.Cart.UserControlUrl 77
Ektron.Cms.Controls.Checkout 78
Ektron.Cms.Controls.Checkout.Cart 80
Ektron.Cms.Controls.Checkout.ControlPhase 78
Ektron.Cms.Controls.Checkout.ControlPhase enumeration 78
Ektron.Cms.Controls.Checkout.CurrentPhase 81
Ektron.Cms.Controls.Checkout.Customer 81
Ektron.Cms.Controls.Checkout.DefaultCountryId 81
Ektron.Cms.Controls.Checkout.DisplayXslt 82
Ektron.Cms.Controls.Checkout.DynamicPhaseParameter 82
Ektron.Cms.Controls.Checkout.Fill 79
Ektron.Cms.Controls.Checkout.FriendlyErrorMessage 82
Ektron.Cms.Controls.Checkout.IsSSLRequired 83
Ektron.Cms.Controls.Checkout.LoadingImage 83
Ektron.Cms.Controls.Checkout.PaymentMethod 83
Ektron.Cms.Controls.Checkout.PhaseChange 88
Ektron.Cms.Controls.Checkout.PhaseChangeEventHandler 87
Ektron.Cms.Controls.Checkout.PostLogin 88
Ektron.Cms.Controls.Checkout.PostLoginEventHandler 87
Ektron.Cms.Controls.Checkout.ShowErrorDetails 84
Ektron.Cms.Controls.Checkout.Stylesheet 84
Ektron.Cms.Controls.Checkout.TemplateCart 84
Ektron.Cms.Controls.Checkout.TemplateOrderHistory 85
Ektron.Cms.Controls.Checkout.TemplateRecoverPassword 85
Ektron.Cms.Controls.Checkout.TemplateShopping 86
Ektron.Cms.Controls.Checkout.ToString 79
Ektron.Cms.Controls.Checkout.UseMsAjax 86
Ektron.Cms.Controls.Checkout.ValidatePhase 86
Ektron.Cms.Controls.Checkout.Wrap 79
Ektron.Cms.Controls.Collection 88
Ektron.Cms.Controls.Collection.AddText 92
Ektron.Cms.Controls.Collection.CacheInterval 93
Ektron.Cms.Controls.Collection.CollectionDescription 93
Ektron.Cms.Controls.Collection.CollectionTitle 93
Ektron.Cms.Controls.Collection.ContentParameter 93
Ektron.Cms.Controls.Collection.CopyTo 89
Ektron.Cms.Controls.Collection.Count 94
Ektron.Cms.Controls.Collection.Current 94
Ektron.Cms.Controls.Collection.DefaultCollectionID 94
Ektron.Cms.Controls.Collection.DynamicParameter 94
Ektron.Cms.Controls.Collection.EkItems 95
Ektron.Cms.Controls.Collection.EnablePaging 95
Ektron.Cms.Controls.Collection.Fill 89
Ektron.Cms.Controls.Collection.GetEnumerator 89
Ektron.Cms.Controls.Collection.GetHtml 95
Ektron.Cms.Controls.Collection.GetRssFeed 90
Ektron.Cms.Controls.Collection.GetView 90
Ektron.Cms.Controls.Collection.GetViewNames 90
Ektron.Cms.Controls.Collection.IncludeIcons 96
Ektron.Cms.Controls.Collection.IsSynchronized 96
Ektron.Cms.Controls.Collection.LinkTarget 96
Ektron.Cms.Controls.Collection.MarkupLanguage 97
Ektron.Cms.Controls.Collection.MaxResults 97
Ektron.Cms.Controls.Collection.MoveNext 90
Ektron.Cms.Controls.Collection.Random 97
Ektron.Cms.Controls.Collection.Reset 91
Ektron.Cms.Controls.Collection.SuppressWrapperTags 97
Ektron.Cms.Controls.Collection.SyncRoot 98
Ektron.Cms.Controls.Collection.this 91
Ektron.Cms.Controls.Collection.ToString 91
Ektron.Cms.Controls.Collection.Wrap 91
Ektron.Cms.Controls.Collection.XmlDoc 98
Ektron.Cms.Controls.CommunityDocuments 98
Ektron.Cms.Controls.CommunityDocuments.AddItemFolderId 99
Ektron.Cms.Controls.CommunityDocuments.DefaultObjectID 100
Ektron.Cms.Controls.CommunityDocuments.DynamicContentBoxHeight 100
Ektron.Cms.Controls.CommunityDocuments.DynamicContentBoxWidth 100
Ektron.Cms.Controls.CommunityDocuments.DynamicContentTemplate 101

Ektron.Cms.Controls.CommunityDocuments.EnableAjax 101
Ektron.Cms.Controls.CommunityDocuments.EnableSearch 101
Ektron.Cms.Controls.CommunityDocuments.Fill 99
Ektron.Cms.Controls.CommunityDocuments.MemberMenuActive 101
Ektron.Cms.Controls.CommunityDocuments.ObjectType 102
Ektron.Cms.Controls.CommunityDocuments.Recursive 102
Ektron.Cms.Controls.CommunityDocuments.ShowAllChecked 102
Ektron.Cms.Controls.CommunityDocuments.Stylesheet 102
Ektron.Cms.Controls.CommunityDocuments.SuppressWrapperTags 103
Ektron.Cms.Controls.CommunityDocuments.TaxonomyDepth 103
Ektron.Cms.Controls.CommunityDocuments.TaxonomyId 103
Ektron.Cms.Controls.CommunityGroupBrowser 104
Ektron.Cms.Controls.CommunityGroupBrowser.AvatarHeight 105
Ektron.Cms.Controls.CommunityGroupBrowser.AvatarWidth 105
Ektron.Cms.Controls.CommunityGroupBrowser.CacheInterval 106
Ektron.Cms.Controls.CommunityGroupBrowser.DisplayXslt 106
Ektron.Cms.Controls.CommunityGroupBrowser.Fill 104
Ektron.Cms.Controls.CommunityGroupBrowser.Link 107
Ektron.Cms.Controls.CommunityGroupBrowser.LinkTarget 107
Ektron.Cms.Controls.CommunityGroupBrowser.MarkupLanguage 107
Ektron.Cms.Controls.CommunityGroupBrowser.MaxResults 107
Ektron.Cms.Controls.CommunityGroupBrowser.Stylesheet 108
Ektron.Cms.Controls.CommunityGroupBrowser.TagTemplate 108
Ektron.Cms.Controls.CommunityGroupBrowser.TagTemplateTarget 108
Ektron.Cms.Controls.CommunityGroupBrowser.TaxonomyId 108
Ektron.Cms.Controls.CommunityGroupBrowser.ToString 104
Ektron.Cms.Controls.CommunityGroupBrowser.Wrap 104
Ektron.Cms.Controls.CommunityGroupBrowser.XmlDoc 109
Ektron.Cms.Controls.CommunityGroupList 109
Ektron.Cms.Controls.CommunityGroupList.AllowGroupCreation 111
Ektron.Cms.Controls.CommunityGroupList.AvatarHeight 111
Ektron.Cms.Controls.CommunityGroupList.AvatarWidth 111
Ektron.Cms.Controls.CommunityGroupList.CacheInterval 111
Ektron.Cms.Controls.CommunityGroupList.DefaultUserID 112
Ektron.Cms.Controls.CommunityGroupList.DisplayMode 112
Ektron.Cms.Controls.CommunityGroupList.DisplayXslt 112
Ektron.Cms.Controls.CommunityGroupList.DynamicUserParameter 113
Ektron.Cms.Controls.CommunityGroupList.Fill 110
Ektron.Cms.Controls.CommunityGroupList.LanguageId 113
Ektron.Cms.Controls.CommunityGroupList.Link 114
Ektron.Cms.Controls.CommunityGroupList.LinkTarget 114
Ektron.Cms.Controls.CommunityGroupList.MarkupLanguage 114
Ektron.Cms.Controls.CommunityGroupList.MaxResults 114
Ektron.Cms.Controls.CommunityGroupList.ShowMySortingOptions 115
Ektron.Cms.Controls.CommunityGroupList.ShowSortingOptions 115
Ektron.Cms.Controls.CommunityGroupList.Stylesheet 115
Ektron.Cms.Controls.CommunityGroupList.TagTemplate 115
Ektron.Cms.Controls.CommunityGroupList.TagTemplateTarget 116
Ektron.Cms.Controls.CommunityGroupList.ToString 110
Ektron.Cms.Controls.CommunityGroupList.Wrap 110
Ektron.Cms.Controls.CommunityGroupList.XmlDoc 116
Ektron.Cms.Controls.CommunityGroupMembers 116
Ektron.Cms.Controls.CommunityGroupMembers.AvatarHeight 118
Ektron.Cms.Controls.CommunityGroupMembers.AvatarWidth 118
Ektron.Cms.Controls.CommunityGroupMembers.CacheInterval 118
Ektron.Cms.Controls.CommunityGroupMembers.DefaultCommunityGroupID 119
Ektron.Cms.Controls.CommunityGroupMembers.DisplayMode 119
Ektron.Cms.Controls.CommunityGroupMembers.DisplayXslt 119
Ektron.Cms.Controls.CommunityGroupMembers.DynamicParameter

120

Ektron.Cms.Controls.CommunityGroupMembers.Fill 117

Ektron.Cms.Controls.CommunityGroupMembers.Link 120

Ektron.Cms.Controls.CommunityGroupMembers.LinkTarget 120

Ektron.Cms.Controls.CommunityGroupMembers.MarkupLanguage
121

Ektron.Cms.Controls.CommunityGroupMembers.MaxResults
121

Ektron.Cms.Controls.CommunityGroupMembers.Stylesheet 121

Ektron.Cms.Controls.CommunityGroupMembers.ToString 117

Ektron.Cms.Controls.CommunityGroupMembers.Wrap 117

Ektron.Cms.Controls.CommunityGroupMembers.XmlDoc 122

Ektron.Cms.Controls.CommunityGroupProfile 122

Ektron.Cms.Controls.CommunityGroupProfile.DefaultGroupID
123

Ektron.Cms.Controls.CommunityGroupProfile.DisplayXslt 124

Ektron.Cms.Controls.CommunityGroupProfile.DynamicParameter
124

Ektron.Cms.Controls.CommunityGroupProfile.EkItem 124

Ektron.Cms.Controls.CommunityGroupProfile.Fill 122

Ektron.Cms.Controls.CommunityGroupProfile.GroupAdminLink
125

Ektron.Cms.Controls.CommunityGroupProfile.GroupAdminLinkTarget
125

Ektron.Cms.Controls.CommunityGroupProfile.Stylesheet 125

Ektron.Cms.Controls.CommunityGroupProfile.TagTemplate 125

Ektron.Cms.Controls.CommunityGroupProfile.TagTemplateTarget
126

Ektron.Cms.Controls.CommunityGroupProfile.Text 126

Ektron.Cms.Controls.CommunityGroupProfile.ToString 122

Ektron.Cms.Controls.CommunityGroupProfile.Wrap 123

Ektron.Cms.Controls.CommunityGroupProfile.XmlDoc 126

Ektron.Cms.Controls.CommunitySearch 127

Ektron.Cms.Controls.CommunitySearch.CacheInterval 129

Ektron.Cms.Controls.CommunitySearch.CustomPropertyID 130

Ektron.Cms.Controls.CommunitySearch.DisplayXslt 130

Ektron.Cms.Controls.CommunitySearch.EnableFilteringResults
131

Ektron.Cms.Controls.CommunitySearch.EnableGroupResults
131

Ektron.Cms.Controls.CommunitySearch.EnableMap 131

Ektron.Cms.Controls.CommunitySearch.EnableMouseOutHidesInfo
132

Ektron.Cms.Controls.CommunitySearch.EnableMouseOver 132

Ektron.Cms.Controls.CommunitySearch.EnableUnauthorizedSearching
132

Ektron.Cms.Controls.CommunitySearch.EnableUserResults 133

Ektron.Cms.Controls.CommunitySearch.Fill 127

Ektron.Cms.Controls.CommunitySearch.FriendsOnly 133

Ektron.Cms.Controls.CommunitySearch.GroupTaxonomyID 133

Ektron.Cms.Controls.CommunitySearch.InfoBoxShowsEmail
134

Ektron.Cms.Controls.CommunitySearch.LoadingImage 134

Ektron.Cms.Controls.CommunitySearch.MapControl 134

Ektron.Cms.Controls.CommunitySearch.MarkupLanguage 135

Ektron.Cms.Controls.CommunitySearch.MaxTagCount 135

Ektron.Cms.Controls.CommunitySearch.MembersOnly 135

Ektron.Cms.Controls.CommunitySearch.PageSize 136

Ektron.Cms.Controls.CommunitySearch.ResultProfile 136

Ektron.Cms.Controls.CommunitySearch.SearchPrivate 136

Ektron.Cms.Controls.CommunitySearch.ShowNameEmail 137

Ektron.Cms.Controls.CommunitySearch.StartingTab 137

Ektron.Cms.Controls.CommunitySearch.Stylesheet 138

Ektron.Cms.Controls.CommunitySearch.TemplateGroupProfile
138

Ektron.Cms.Controls.CommunitySearch.TemplateParamName
138

Ektron.Cms.Controls.CommunitySearch.TemplateTarget 139

Ektron.Cms.Controls.CommunitySearch.TemplateUserProfile
139

Ektron.Cms.Controls.CommunitySearch.ToString 127

Ektron.Cms.Controls.CommunitySearch.UseMsAjax 139

Ektron.Cms.Controls.CommunitySearch.UserTaxonomyID 140

Ektron.Cms.Controls.CommunitySearch.Wrap 128

Ektron.Cms.Controls.CommunitySearch.XmlDoc 140

Ektron.Cms.Controls.CommunitySearch.XsltDoc 140

Ektron.Cms.Controls.ContentBlock 140

Ektron.Cms.Controls.ContentBlock.CacheInterval 143

Ektron.Cms.Controls.ContentBlock.CopyTo 141

Ektron.Cms.Controls.ContentBlock.Count 144

Ektron.Cms.Controls.ContentBlock.Current 144
Ektron.Cms.Controls.ContentBlock.DefaultContentID 144
Ektron.Cms.Controls.ContentBlock.DisplayXslt 145
Ektron.Cms.Controls.ContentBlock.DynamicParameter 145
Ektron.Cms.Controls.ContentBlock.EkItem 145
Ektron.Cms.Controls.ContentBlock.Fill 142
Ektron.Cms.Controls.ContentBlock.GetEnumerator 142
Ektron.Cms.Controls.ContentBlock.IsSynchronized 146
Ektron.Cms.Controls.ContentBlock.MoveNext 142
Ektron.Cms.Controls.ContentBlock.OverrideXslt 146
Ektron.Cms.Controls.ContentBlock.Reset 142
Ektron.Cms.Controls.ContentBlock.StartingFolderID 146
Ektron.Cms.Controls.ContentBlock.SyncRoot 147
Ektron.Cms.Controls.ContentBlock.this 142
Ektron.Cms.Controls.ContentBlock.ToString 142
Ektron.Cms.Controls.ContentBlock.Wrap 143
Ektron.Cms.Controls.ContentFlagging 147
Ektron.Cms.Controls.ContentFlagging.DefaultContentID 148
Ektron.Cms.Controls.ContentFlagging.DisplayXslt 149
Ektron.Cms.Controls.ContentFlagging.DynamicParameter 149
Ektron.Cms.Controls.ContentFlagging.Fill 147
Ektron.Cms.Controls.ContentFlagging.FlagImage 149
Ektron.Cms.Controls.ContentFlagging.FlagText 150
Ektron.Cms.Controls.ContentFlagging.Stylesheet 150
Ektron.Cms.Controls.ContentFlagging.ToString 147
Ektron.Cms.Controls.ContentFlagging.Wrap 148
Ektron.Cms.Controls.ContentFlagging.XmlDoc 151
Ektron.Cms.Controls.ContentList 151
Ektron.Cms.Controls.ContentList.CacheInterval 155
Ektron.Cms.Controls.ContentList.ContentIds 155
Ektron.Cms.Controls.ContentList.CopyTo 152
Ektron.Cms.Controls.ContentList.Count 156
Ektron.Cms.Controls.ContentList.Current 156
Ektron.Cms.Controls.ContentList.DefaultContentID 156
Ektron.Cms.Controls.ContentList.Direction 156
Ektron.Cms.Controls.ContentList.DynamicParameter 157
Ektron.Cms.Controls.ContentList.EkItems 157
Ektron.Cms.Controls.ContentList.Fill 152
Ektron.Cms.Controls.ContentList.GetEnumerator 152
Ektron.Cms.Controls.ContentList.GetHtml 157
Ektron.Cms.Controls.ContentList.GetRssFeed 152
Ektron.Cms.Controls.ContentList.IncludeIcons 158
Ektron.Cms.Controls.ContentList.IsSynchronized 158
Ektron.Cms.Controls.ContentList.LinkTarget 158
Ektron.Cms.Controls.ContentList.MarkupLanguage 159
Ektron.Cms.Controls.ContentList.MetaTag 159
Ektron.Cms.Controls.ContentList.MoveNext 152
Ektron.Cms.Controls.ContentList.OrderByDirection 159
Ektron.Cms.Controls.ContentList.OrderKey 160
Ektron.Cms.Controls.ContentList.Random 160
Ektron.Cms.Controls.ContentList.Reset 153
Ektron.Cms.Controls.ContentList.SuppressWrapperTags 160
Ektron.Cms.Controls.ContentList.SyncRoot 161
Ektron.Cms.Controls.ContentList.this 153
Ektron.Cms.Controls.ContentList.ToString 153
Ektron.Cms.Controls.ContentList.Wrap 153
Ektron.Cms.Controls.ContentList.XmlDoc 161
Ektron.Cms.Controls.ContentRating 161
Ektron.Cms.Controls.ContentRating.AlreadyRatedMessage 163
Ektron.Cms.Controls.ContentRating.BadLabel 163
Ektron.Cms.Controls.ContentRating.CharsRemainingLabel 164
Ektron.Cms.Controls.ContentRating.CompleteGraphDisplay 164
Ektron.Cms.Controls.ContentRating.CookiesRequiredMessage 164
Ektron.Cms.Controls.ContentRating.DefaultContentID 165
Ektron.Cms.Controls.ContentRating.DynamicParameter 165
Ektron.Cms.Controls.ContentRating.FeedbackHeader 165
Ektron.Cms.Controls.ContentRating.Fill 162
Ektron.Cms.Controls.ContentRating.GoodLabel 166
Ektron.Cms.Controls.ContentRating.GraphBarColor 166
Ektron.Cms.Controls.ContentRating.InitialGraphDisplay 166
Ektron.Cms.Controls.ContentRating.Padding 167
Ektron.Cms.Controls.ContentRating.RatingLevelLabel 167
Ektron.Cms.Controls.ContentRating.ThankYouMessage 168
Ektron.Cms.Controls.ContentRating.ToString 162
Ektron.Cms.Controls.ContentRating.TotalRatingsLabel 168
Ektron.Cms.Controls.ContentRating.UserCommentsHeader 168
Ektron.Cms.Controls.ContentRating.ValidationMessage 169
Ektron.Cms.Controls.ContentRating.Wrap 162

Ektron.Cms.Controls.ContentReview 169
Ektron.Cms.Controls.ContentReview.DefaultContentID 171
Ektron.Cms.Controls.ContentReview.DisplayXslt 172
Ektron.Cms.Controls.ContentReview.DynamicParameter 172
Ektron.Cms.Controls.ContentReview.DynamicUserParameter 173
Ektron.Cms.Controls.ContentReview.Fill 170
Ektron.Cms.Controls.ContentReview.GetReviews 173
Ektron.Cms.Controls.ContentReview.MaxReviews 173
Ektron.Cms.Controls.ContentReview.Moderate 174
Ektron.Cms.Controls.ContentReview.RaiseCallbackEvent 170
Ektron.Cms.Controls.ContentReview.RatingsMinimum 174
Ektron.Cms.Controls.ContentReview.ToString 170
Ektron.Cms.Controls.ContentReview.UserID 174
Ektron.Cms.Controls.ContentReview.Wrap 170
Ektron.Cms.Controls.ContentReview.XmlDoc 175
Ektron.Cms.Controls.CurrencySelect 175
Ektron.Cms.Controls.CurrencySelect.CacheInterval 176
Ektron.Cms.Controls.CurrencySelect.DisplayXslt 177
Ektron.Cms.Controls.CurrencySelect.EkItem 178
Ektron.Cms.Controls.CurrencySelect.Fill 176
Ektron.Cms.Controls.CurrencySelect.Stylesheet 178
Ektron.Cms.Controls.CurrencySelect.ToString 176
Ektron.Cms.Controls.CurrencySelect.Wrap 176
Ektron.Cms.Controls.DefaultCatalogIDUITypeEditor 178
Ektron.Cms.Controls.DesignTimeDiagnostic 178
Ektron.Cms.Controls.DesignTimeDiagnostic.Authenticated 180
Ektron.Cms.Controls.DesignTimeDiagnostic.Fill 179
Ektron.Cms.Controls.DesignTimeDiagnostic.Hide 180
Ektron.Cms.Controls.DesignTimeDiagnostic.Language 180
Ektron.Cms.Controls.DesignTimeDiagnostic.SuppressWrapperTags 180
Ektron.Cms.Controls.DesignTimeDiagnostic.WrapTag 180
Ektron.Cms.Controls.DhtmlMenu 181
Ektron.Cms.Controls.DhtmlMenu.AccessKey 183
Ektron.Cms.Controls.DhtmlMenu.Align 183
Ektron.Cms.Controls.DhtmlMenu.BackColor 183
Ektron.Cms.Controls.DhtmlMenu.BorderColor 183
Ektron.Cms.Controls.DhtmlMenu.BorderStyle 184
Ektron.Cms.Controls.DhtmlMenu.BorderWidth 184
Ektron.Cms.Controls.DhtmlMenu.CssClass 184
Ektron.Cms.Controls.DhtmlMenu.DisplayXslt 184
Ektron.Cms.Controls.DhtmlMenu.Enabled 184
Ektron.Cms.Controls.DhtmlMenu.EnableIE6FormsFix 185
Ektron.Cms.Controls.DhtmlMenu.Fill 181
Ektron.Cms.Controls.DhtmlMenu.Height 185
Ektron.Cms.Controls.DhtmlMenu.HorizontalDisplay 185
Ektron.Cms.Controls.DhtmlMenu.MenuHoverOverColor 186
Ektron.Cms.Controls.DhtmlMenu.MenuStyle 186
Ektron.Cms.Controls.DhtmlMenu.Stylesheet 186
Ektron.Cms.Controls.DhtmlMenu.SubMenuBackColor 186
Ektron.Cms.Controls.DhtmlMenu.SubMenuHoverOverColor 187
Ektron.Cms.Controls.DhtmlMenu.SubMenuItemHeight 187
Ektron.Cms.Controls.DhtmlMenu.SubMenuItemWidth 187
Ektron.Cms.Controls.DhtmlMenu.SubMenuTopOffset 188
Ektron.Cms.Controls.DhtmlMenu.TabIndex 188
Ektron.Cms.Controls.DhtmlMenu.ToolTip 188
Ektron.Cms.Controls.DhtmlMenu.Width 188
Ektron.Cms.Controls.DhtmlMenu.Wrap 181
Ektron.Cms.Controls.Directory 189
Ektron.Cms.Controls.Directory.AddItemFolderId 193
Ektron.Cms.Controls.Directory.CacheInterval 194
Ektron.Cms.Controls.Directory.CopyTo 190
Ektron.Cms.Controls.Directory.Count 194
Ektron.Cms.Controls.Directory.Current 194
Ektron.Cms.Controls.Directory.DisplayXslt 194
Ektron.Cms.Controls.Directory.DynamicParameter 195
Ektron.Cms.Controls.Directory.EkItems 195
Ektron.Cms.Controls.Directory.EnableAjax 195
Ektron.Cms.Controls.Directory.EnablePaging 196
Ektron.Cms.Controls.Directory.EnableSearch 196
Ektron.Cms.Controls.Directory.Fill 190
Ektron.Cms.Controls.Directory.GetEnumerator 190
Ektron.Cms.Controls.Directory.GetView 190
Ektron.Cms.Controls.Directory.GetViewNames 190
Ektron.Cms.Controls.Directory.IncludeIcons 196
Ektron.Cms.Controls.Directory.IsSynchronized 197
Ektron.Cms.Controls.Directory.ItemSortOrder 197

Ektron.Cms.Controls.Directory.LinkTarget 197
Ektron.Cms.Controls.Directory.MarkupLanguage 198
Ektron.Cms.Controls.Directory.MaxResults 198
Ektron.Cms.Controls.Directory.MoveNext 191
Ektron.Cms.Controls.Directory.Recursive 198
Ektron.Cms.Controls.Directory.Reset 191
Ektron.Cms.Controls.Directory.ShowAllChecked 198
Ektron.Cms.Controls.Directory.ShowCount 199
Ektron.Cms.Controls.Directory.SortDirection 199
Ektron.Cms.Controls.Directory.SuppressWrapperTags 199
Ektron.Cms.Controls.Directory.SyncRoot 200
Ektron.Cms.Controls.Directory.TaxonomyCols 200
Ektron.Cms.Controls.Directory.TaxonomyDepth 200
Ektron.Cms.Controls.Directory.TaxonomyDescription 200
Ektron.Cms.Controls.Directory.TaxonomyId 201
Ektron.Cms.Controls.Directory.TaxonomyItemCols 201
Ektron.Cms.Controls.Directory.TaxonomyPath 201
Ektron.Cms.Controls.Directory.TaxonomyTitle 201
Ektron.Cms.Controls.Directory.TaxonomyTreeData 202
Ektron.Cms.Controls.Directory.TaxonomyXml 202
Ektron.Cms.Controls.Directory.this 191
Ektron.Cms.Controls.Directory.ToString 191
Ektron.Cms.Controls.Directory.Wrap 192
Ektron.Cms.Controls.Directory.XmlDoc 202
Ektron.Cms.Controls.EktronCatalogPart 202
Ektron.Cms.Controls.EktronCatalogPart.BlogID 203
Ektron.Cms.Controls.EktronCatalogPart.BoardID 203
Ektron.Cms.Controls.EktronCatalogPart.CalendarDynamicParameter 204
Ektron.Cms.Controls.EktronCatalogPart.CollectionDynamicParameter 204
Ektron.Cms.Controls.EktronCatalogPart.ContentBlockDynamicParameter 204
Ektron.Cms.Controls.EktronCatalogPart.DefaultCalendarID 205
Ektron.Cms.Controls.EktronCatalogPart.DefaultCollectionID 205
Ektron.Cms.Controls.EktronCatalogPart.DefaultContentID 205
Ektron.Cms.Controls.EktronCatalogPart.DefaultFormID 206
Ektron.Cms.Controls.EktronCatalogPart.EnableBlogControl 206
Ektron.Cms.Controls.EktronCatalogPart.EnableCalendarControl 206
Ektron.Cms.Controls.EktronCatalogPart.EnableCollectionControl 206
Ektron.Cms.Controls.EktronCatalogPart.EnableContentBlockControl 207
Ektron.Cms.Controls.EktronCatalogPart.EnableDiscussionBoardsControl 207
Ektron.Cms.Controls.EktronCatalogPart.EnableFormBlockControl 207
Ektron.Cms.Controls.EktronCatalogPart.EnableListSummaryControl 208
Ektron.Cms.Controls.EktronCatalogPart.EnableThreadedDiscussionsControl 208
Ektron.Cms.Controls.EktronCatalogPart.FolderID 208
Ektron.Cms.Controls.EktronCatalogPart.FormBlockDynamicParameter 208
Ektron.Cms.Controls.EktronCatalogPart.Title 209
Ektron.Cms.Controls.EktronCommunityCatalogPart 209
Ektron.Cms.Controls.EktronCommunityCatalogPart.EnableBlogs 210
Ektron.Cms.Controls.EktronCommunityCatalogPart.EnableDiscussionBoards 210
Ektron.Cms.Controls.EktronCommunityCatalogPart.EnableForums 210
Ektron.Cms.Controls.EktronCommunityCatalogPart.Title 210
Ektron.Cms.Controls.EktronWebPartZone 211
Ektron.Cms.Controls.EktronWebPartZone.CmsSitePath 211
Ektron.Cms.Controls.EkWebControl 212
Ektron.Cms.Controls.EkWebControl.Authenticated 218
Ektron.Cms.Controls.EkWebControl.CmsSitePath 218
Ektron.Cms.Controls.EkWebControl.EkWebControl 212
Ektron.Cms.Controls.EkWebControl.Error 218
Ektron.Cms.Controls.EkWebControl.Fill 213
Ektron.Cms.Controls.EkWebControl.GetEnumDescription 213
Ektron.Cms.Controls.EkWebControl.GetFilenameFromPath 213
Ektron.Cms.Controls.EkWebControl.GetMessage 214

Ektron.Cms.Controls.EkWebControl.Hide 219
Ektron.Cms.Controls.EkWebControl.IsLoggedIn 219
Ektron.Cms.Controls.EkWebControl.Language 219
Ektron.Cms.Controls.EkWebControl.LanguageID 220
Ektron.Cms.Controls.EkWebControl.LocalizeString 214
Ektron.Cms.Controls.EkWebControl.loggedInUserID 220
Ektron.Cms.Controls.EkWebControl.loggedInUserName 220
Ektron.Cms.Controls.EkWebControl.LogLevel 220
Ektron.Cms.Controls.EkWebControl.RedirectFromLoginPage 215
Ektron.Cms.Controls.EkWebControl.RedirectToLoginPage 215
Ektron.Cms.Controls.EkWebControl.RegisterAllDmsMenuCssAndScripts 215
Ektron.Cms.Controls.EkWebControl.RegisterDmsMenuCSS 215
Ektron.Cms.Controls.EkWebControl.RegisterDmsMenuScript 216
Ektron.Cms.Controls.EkWebControl.RegisterJQueryScript 216
Ektron.Cms.Controls.EkWebControl.RegisterMembershipTabScript 216
Ektron.Cms.Controls.EkWebControl.RegisterThickBoxCSS 216
Ektron.Cms.Controls.EkWebControl.RegisterThickBoxScript 216
Ektron.Cms.Controls.EkWebControl.SearchHeader 216
Ektron.Cms.Controls.EkWebControl.SuppressWrapperTags 220
Ektron.Cms.Controls.EkWebControl.Text 221
Ektron.Cms.Controls.EkWebControl.ToString 217
Ektron.Cms.Controls.EkWebControl.Wrap 217
Ektron.Cms.Controls.EkWebControl.WrapTag 221
Ektron.Cms.Controls.EkWebPart 222
Ektron.Cms.Controls.EkWebPart.CatalogIconImageUrl 223
Ektron.Cms.Controls.EkWebPart.CreateEditorParts 222
Ektron.Cms.Controls.EkWebPart.Description 223
Ektron.Cms.Controls.EkWebPart.Subtitle 223
Ektron.Cms.Controls.EkWebPart.Title 223
Ektron.Cms.Controls.EkWebPart.TitleIconImageUrl 223
Ektron.Cms.Controls.EkWebPart.TitleUrl 224
Ektron.Cms.Controls.EkWebPart.WebBrowsableObject 224
Ektron.Cms.Controls.EkXsltBase 224
Ektron.Cms.Controls.EkXsltBase.CustomXml 225
Ektron.Cms.Controls.EkXsltBase.DisplayXslt 225
Ektron.Cms.Controls.EkXsltBase.EkXsltBase 224
Ektron.Cms.Controls.EkXsltBase.EnableErrorLogging 225
Ektron.Cms.Controls.EkXsltBase.XmlDoc 226
Ektron.Cms.Controls.EkXsltBase.XsltDoc 226
Ektron.Cms.Controls.ExplorerDragDrop 226
Ektron.Cms.Controls.ExplorerDragDrop.AssetID 227
Ektron.Cms.Controls.ExplorerDragDrop.ContentLanguage 227
Ektron.Cms.Controls.ExplorerDragDrop.FolderID 227
Ektron.Cms.Controls.ExplorerDragDrop.IsImage 228
Ektron.Cms.Controls.ExplorerDragDrop.OverrideExtension 228
Ektron.Cms.Controls.ExplorerDragDrop.TaxonomyId 228
Ektron.Cms.Controls.Favorites 229
Ektron.Cms.Controls.Favorites.CacheInterval 231
Ektron.Cms.Controls.Favorites.DefaultUserID 231
Ektron.Cms.Controls.Favorites.DisplayMode 231
Ektron.Cms.Controls.Favorites.DisplayXslt 232
Ektron.Cms.Controls.Favorites.DynamicParameter 232
Ektron.Cms.Controls.Favorites.FavoritesTitle 232
Ektron.Cms.Controls.Favorites.FavoritesUriTitle 233
Ektron.Cms.Controls.Favorites.Fill 229
Ektron.Cms.Controls.Favorites.Link 233
Ektron.Cms.Controls.Favorites.LinkTarget 233
Ektron.Cms.Controls.Favorites.MarkupLanguage 234
Ektron.Cms.Controls.Favorites.MaxResults 234
Ektron.Cms.Controls.Favorites.Stylesheet 234
Ektron.Cms.Controls.Favorites.ToString 229
Ektron.Cms.Controls.Favorites.Wrap 229
Ektron.Cms.Controls.Favorites.XmlDoc 235
Ektron.Cms.Controls.FlexMenu 235
Ektron.Cms.Controls.FlexMenu.AutoCollapseBranches 236
Ektron.Cms.Controls.FlexMenu.CacheInterval 237
Ektron.Cms.Controls.FlexMenu.DisplayXslt 237
Ektron.Cms.Controls.FlexMenu.EnableAjax 237
Ektron.Cms.Controls.FlexMenu.EnableMouseOverPopUp 238
Ektron.Cms.Controls.FlexMenu.EnableSmartOpen 238
Ektron.Cms.Controls.FlexMenu.EnableSubmenuLinkPathMatching 238
Ektron.Cms.Controls.FlexMenu.EnableSubmenuTemplateMatching 239

Ektron.Cms.Controls.FlexMenu.Fill 235
Ektron.Cms.Controls.FlexMenu.MasterControlId 239
Ektron.Cms.Controls.FlexMenu.MenuDepth 239
Ektron.Cms.Controls.FlexMenu.SetDirty 235
Ektron.Cms.Controls.FlexMenu.StartCollapsed 240
Ektron.Cms.Controls.FlexMenu.StartLevel 240
Ektron.Cms.Controls.FlexMenu.Stylesheet 240
Ektron.Cms.Controls.FlexMenu.UseAliasRealId 241
Ektron.Cms.Controls.FlexMenu.XmlDoc 241
Ektron.Cms.Controls.FlexMenu.XsltDoc 241
Ektron.Cms.Controls.FolderBreadcrumb 241
Ektron.Cms.Controls.FolderBreadcrumb.DefaultContentID 243
Ektron.Cms.Controls.FolderBreadcrumb.DefaultFolderID 243
Ektron.Cms.Controls.FolderBreadcrumb.DisplayStyle 244
Ektron.Cms.Controls.FolderBreadcrumb.DisplayXslt 244
Ektron.Cms.Controls.FolderBreadcrumb.DynamicParameter 244
Ektron.Cms.Controls.FolderBreadcrumb.Fill 242
Ektron.Cms.Controls.FolderBreadcrumb.LinkTarget 245
Ektron.Cms.Controls.FolderBreadcrumb.Mode 245
Ektron.Cms.Controls.FolderBreadcrumb.separator 245
Ektron.Cms.Controls.FolderBreadcrumb.XmlDoc 245
Ektron.Cms.Controls.FormBlock 246
Ektron.Cms.Controls.FormBlock.AddValidation 249
Ektron.Cms.Controls.FormBlock.CacheInterval 250
Ektron.Cms.Controls.FormBlock.CopyTo 247
Ektron.Cms.Controls.FormBlock.Count 250
Ektron.Cms.Controls.FormBlock.Current 250
Ektron.Cms.Controls.FormBlock.DefaultFormID 250
Ektron.Cms.Controls.FormBlock.DynamicParameter 251
Ektron.Cms.Controls.FormBlock.EkItem 251
Ektron.Cms.Controls.FormBlock.Fields 251
Ektron.Cms.Controls.FormBlock.Fill 247
Ektron.Cms.Controls.FormBlock.FormName 252
Ektron.Cms.Controls.FormBlock.GetEnumerator 247
Ektron.Cms.Controls.FormBlock.GetView 247
Ektron.Cms.Controls.FormBlock.GetViewNames 247
Ektron.Cms.Controls.FormBlock.IncludeTags 252
Ektron.Cms.Controls.FormBlock.IsSynchronized 252
Ektron.Cms.Controls.FormBlock.MoveNext 248
Ektron.Cms.Controls.FormBlock.Reset 248
Ektron.Cms.Controls.FormBlock.StartingFolderID 253
Ektron.Cms.Controls.FormBlock.SyncRoot 253
Ektron.Cms.Controls.FormBlock.this 248
Ektron.Cms.Controls.FormBlock.ToString 248
Ektron.Cms.Controls.FormBlock.Wrap 248
Ektron.Cms.Controls.FormFieldItem 253
Ektron.Cms.Controls.FormFieldItem.Basetype 254
Ektron.Cms.Controls.FormFieldItem.ContentType 254
Ektron.Cms.Controls.FormFieldItem.DatalistItems 254
Ektron.Cms.Controls.FormFieldItem.Datatype 255
Ektron.Cms.Controls.FormFieldItem.DisplayName 255
Ektron.Cms.Controls.FormFieldItem.FieldName 255
Ektron.Cms.Controls.FormFieldItem.Language 256
Ektron.Cms.Controls.FormFieldItem.Value 256
Ektron.Cms.Controls.Forum 256
Ektron.Cms.Controls.Forum.AllowAnonymousPost 260
Ektron.Cms.Controls.Forum.BoardID 260
Ektron.Cms.Controls.Forum.CacheInterval 260
Ektron.Cms.Controls.Forum.CustomOrderBy 261
Ektron.Cms.Controls.Forum.DynamicForumParameter 261
Ektron.Cms.Controls.Forum.DynamicGroupParameter 261
Ektron.Cms.Controls.Forum.DynamicParameter 262
Ektron.Cms.Controls.Forum.DynamicThreadParameter 262
Ektron.Cms.Controls.Forum.EditorCSS 262
Ektron.Cms.Controls.Forum.EditorToolbar 263
Ektron.Cms.Controls.Forum.EnableForumQuickSelect 263
Ektron.Cms.Controls.Forum.EnableThreadRating 263
Ektron.Cms.Controls.Forum.ExcludeList 264
Ektron.Cms.Controls.Forum.Fill 257
Ektron.Cms.Controls.Forum.FilterXslt 264
Ektron.Cms.Controls.Forum.GetCallbackResult 257
Ektron.Cms.Controls.Forum.HideUserProfile 265
Ektron.Cms.Controls.Forum.HighlightedUserRole 265
Ektron.Cms.Controls.Forum.JavascriptEditorHeight 265
Ektron.Cms.Controls.Forum.JavascriptEditorHTMLMode 266
Ektron.Cms.Controls.Forum.JavascriptEditorToolbar 266
Ektron.Cms.Controls.Forum.JavascriptEditorWidth 267
Ektron.Cms.Controls.Forum.OrderBy 267

Ektron.Cms.Controls.Forum.OrderDirection 268
Ektron.Cms.Controls.Forum.ProfileLink 268
Ektron.Cms.Controls.Forum.ProfileLinkTarget 268
Ektron.Cms.Controls.Forum.RaiseCallbackEvent 257
Ektron.Cms.Controls.Forum.ShowCategories 268
Ektron.Cms.Controls.Forum.TaxCategories 269
Ektron.Cms.Controls.Forum.TaxonomyCols 269
Ektron.Cms.Controls.Forum.TaxonomyMarkupLanguage 269
Ektron.Cms.Controls.Forum.Theme 269
Ektron.Cms.Controls.Forum.ToString 257
Ektron.Cms.Controls.Forum.Wrap 258
Ektron.Cms.Controls.Friends 270
Ektron.Cms.Controls.Friends.AvatarHeight 272
Ektron.Cms.Controls.Friends.AvatarWidth 272
Ektron.Cms.Controls.Friends.CacheInterval 273
Ektron.Cms.Controls.Friends.DefaultUserID 273
Ektron.Cms.Controls.Friends.DisplayMode 273
Ektron.Cms.Controls.Friends.DisplayXslt 274
Ektron.Cms.Controls.Friends.DynamicParameter 274
Ektron.Cms.Controls.Friends.Fill 270
Ektron.Cms.Controls.Friends.FriendLabel 275
Ektron.Cms.Controls.Friends.FriendsLabel 275
Ektron.Cms.Controls.Friends.Link 275
Ektron.Cms.Controls.Friends.LinkTarget 275
Ektron.Cms.Controls.Friends.MarkupLanguage 276
Ektron.Cms.Controls.Friends.MaxResults 276
Ektron.Cms.Controls.Friends.ShowSortingOptions 276
Ektron.Cms.Controls.Friends.Stylesheet 277
Ektron.Cms.Controls.Friends.ToString 271
Ektron.Cms.Controls.Friends.Wrap 271
Ektron.Cms.Controls.Friends.XmlDoc 277
Ektron.Cms.Controls.HtmlEditDisplay 277
Ektron.Cms.Controls.HtmlEditDisplay.BaseJavaScript 278
Ektron.Cms.Controls.HtmlEditDisplay.BeginJavaScript 278
Ektron.Cms.Controls.HtmlEditDisplay.EndJavaScript 279
Ektron.Cms.Controls.HtmlEditDisplay.MakeFieldNameFromId 279
Ektron.Cms.Controls.HtmlEditDisplay.RenderCMSFields 279
Ektron.Cms.Controls.HtmlEditDisplay.SelectedJavaScript 280
Ektron.Cms.Controls.HtmlEditor 280
Ektron.Cms.Controls.HtmlEditor.Archived 284
Ektron.Cms.Controls.HtmlEditor.Authenticated 284
Ektron.Cms.Controls.HtmlEditor.CmsSitePath 285
Ektron.Cms.Controls.HtmlEditor.ContentAction 285
Ektron.Cms.Controls.HtmlEditor.ContentID 286
Ektron.Cms.Controls.HtmlEditor.ContentTitle 286
Ektron.Cms.Controls.HtmlEditor.ContentType 287
Ektron.Cms.Controls.HtmlEditor.EditorReadyCall 287
Ektron.Cms.Controls.HtmlEditor.EnableUpload 288
Ektron.Cms.Controls.HtmlEditor.ExecuteContentAction 281
Ektron.Cms.Controls.HtmlEditor.HtmlEditor 281
Ektron.Cms.Controls.HtmlEditor.IncludeContentInfo 288
Ektron.Cms.Controls.HtmlEditor.IsLoggedIn 289
Ektron.Cms.Controls.HtmlEditor.Language 289
Ektron.Cms.Controls.HtmlEditor.loggedInUserID 290
Ektron.Cms.Controls.HtmlEditor.loggedInUserName 290
Ektron.Cms.Controls.HtmlEditor.Path 290
Ektron.Cms.Controls.HtmlEditor.PreferredType 291
Ektron.Cms.Controls.HtmlEditor.SafeEncode 291
Ektron.Cms.Controls.HtmlEditor.ShowCMSToolbar 292
Ektron.Cms.Controls.HtmlEditor.ShowTitle 292
Ektron.Cms.Controls.HtmlEditor.StyleSheet 293
Ektron.Cms.Controls.HtmlEditor.Text 293
Ektron.Cms.Controls.HtmlEditor.ToolbarLevel 294
Ektron.Cms.Controls.HtmlEditor.ToolbarResetCall 294
Ektron.Cms.Controls.HtmlEditor.ToString 282
Ektron.Cms.Controls.HtmlEditor.TranslationFile 295
Ektron.Cms.Controls.HtmlEditor.WorkareaMode 282
Ektron.Cms.Controls.ImageControl 295
Ektron.Cms.Controls.ImageControl.DefaultImageID 297
Ektron.Cms.Controls.ImageControl.DynamicParameter 297
Ektron.Cms.Controls.ImageControl.Fill 296
Ektron.Cms.Controls.ImageControl.FolderID 297
Ektron.Cms.Controls.ImageControl.TaxonomyID 298
Ektron.Cms.Controls.ImageControl.Title 298
Ektron.Cms.Controls.ImageControl.Wrap 296
Ektron.Cms.Controls.IndexSearch 298
Ektron.Cms.Controls.IndexSearch.AddParm 300
Ektron.Cms.Controls.IndexSearch.ButtonImgSrc 305

Ektron.Cms.Controls.IndexSearch.ButtonText 305
Ektron.Cms.Controls.IndexSearch.ContentParameter 306
Ektron.Cms.Controls.IndexSearch.CopyTo 301
Ektron.Cms.Controls.IndexSearch.Count 306
Ektron.Cms.Controls.IndexSearch.Current 306
Ektron.Cms.Controls.IndexSearch.CustomOrderBy 306
Ektron.Cms.Controls.IndexSearch.EkItems 307
Ektron.Cms.Controls.IndexSearch.EmptyResultMsg 307
Ektron.Cms.Controls.IndexSearch.EncodeContentHtml 307
Ektron.Cms.Controls.IndexSearch.Fill 301
Ektron.Cms.Controls.IndexSearch.FolderId 307
Ektron.Cms.Controls.IndexSearch.FormName 308
Ektron.Cms.Controls.IndexSearch.GetEnumerator 301
Ektron.Cms.Controls.IndexSearch.GetSearchParmDataType 301
Ektron.Cms.Controls.IndexSearch.GetSearchParmSearchType 302
Ektron.Cms.Controls.IndexSearch.GetXPathValues 302
Ektron.Cms.Controls.IndexSearch.IndexSearch 299
Ektron.Cms.Controls.IndexSearch.IsSynchronized 308
Ektron.Cms.Controls.IndexSearch.Labelafter 308
Ektron.Cms.Controls.IndexSearch.Labelbefore 308
Ektron.Cms.Controls.IndexSearch.LabelBetween 309
Ektron.Cms.Controls.IndexSearch.LabelContains 309
Ektron.Cms.Controls.IndexSearch.LabeldateBetween 309
Ektron.Cms.Controls.IndexSearch.LabelEqual 310
Ektron.Cms.Controls.IndexSearch.LabelExactPhrase 310
Ektron.Cms.Controls.IndexSearch.LabelGreaterThan 310
Ektron.Cms.Controls.IndexSearch.Labelinputdate 311
Ektron.Cms.Controls.IndexSearch.Labelinputnumber 311
Ektron.Cms.Controls.IndexSearch.LabelLessThan 311
Ektron.Cms.Controls.IndexSearch.Labelon 312
Ektron.Cms.Controls.IndexSearch.LabelSelect 312
Ektron.Cms.Controls.IndexSearch.LinkTarget 312
Ektron.Cms.Controls.IndexSearch.MaxResults 313
Ektron.Cms.Controls.IndexSearch.MoveNext 302
Ektron.Cms.Controls.IndexSearch.OrderBy 313
Ektron.Cms.Controls.IndexSearch.OrderByDirection 313
Ektron.Cms.Controls.IndexSearch.Recursive 314
Ektron.Cms.Controls.IndexSearch.Reset 302
Ektron.Cms.Controls.IndexSearch.Search 303
Ektron.Cms.Controls.IndexSearch.SearchParmXML 314
Ektron.Cms.Controls.IndexSearch.SearchPram 299
Ektron.Cms.Controls.IndexSearch.SearchPram structure 299
Ektron.Cms.Controls.IndexSearch.ShowSearchBoxAlways 314
Ektron.Cms.Controls.IndexSearch.SyncRoot 315
Ektron.Cms.Controls.IndexSearch.this 303
Ektron.Cms.Controls.IndexSearch.Weighted 315
Ektron.Cms.Controls.IndexSearch.Wrap 303
Ektron.Cms.Controls.IndexSearch.XmlConfigId 315
Ektron.Cms.Controls.IndexSearch.XmlDoc 315
Ektron.Cms.Controls.Invite 316
Ektron.Cms.Controls.Invite.DisplayXslt 317
Ektron.Cms.Controls.Invite.Fill 316
Ektron.Cms.Controls.Invite.FriendMessageId 317
Ektron.Cms.Controls.Invite.InviteMessageId 318
Ektron.Cms.Controls.Invite.OptionalMessageText 318
Ektron.Cms.Controls.Invite.Stylesheet 318
Ektron.Cms.Controls.Invite.ToString 316
Ektron.Cms.Controls.Invite.Wrap 316
Ektron.Cms.Controls.Invite.XmlDoc 318
Ektron.Cms.Controls.LanguageAPI 319
Ektron.Cms.Controls.LanguageAPI.CurrentLanguage 321
Ektron.Cms.Controls.LanguageAPI.CurrentLanguageID 321
Ektron.Cms.Controls.LanguageAPI.DefaultLanguage 321
Ektron.Cms.Controls.LanguageAPI.DefaultLanguageID 322
Ektron.Cms.Controls.LanguageAPI.Fill 319
Ektron.Cms.Controls.LanguageAPI.GetLanguage 320
Ektron.Cms.Controls.LanguageAPI.GetLanguageID 320
Ektron.Cms.Controls.LanguageAPI.IsValid 320
Ektron.Cms.Controls.LanguageAPI.LangaugeIdList 322
Ektron.Cms.Controls.LanguageAPI.LangaugeTitleList 322
Ektron.Cms.Controls.LanguageAPI.LanguageAPI 319
Ektron.Cms.Controls.LanguageAPI.MultiLanguageEnabled 322
Ektron.Cms.Controls.LanguageAPI.OverrideDefault 322
Ektron.Cms.Controls.LanguageAPI.SiteLanguage 323
Ektron.Cms.Controls.LanguageAPI.SiteLanguageID 323
Ektron.Cms.Controls.LanguageSelect 323
Ektron.Cms.Controls.LanguageSelect.Fill 324

Ektron.Cms.Controls.LanguageSelect.LabelName 324
Ektron.Cms.Controls.LanguageSelect.Text 325
Ektron.Cms.Controls.LanguageSelect.Wrap 324
Ektron.Cms.Controls.ListSummary 325
Ektron.Cms.Controls.ListSummary.AddText 329
Ektron.Cms.Controls.ListSummary.CacheInterval 330
Ektron.Cms.Controls.ListSummary.ContentParameter 330
Ektron.Cms.Controls.ListSummary.ContentType 331
Ektron.Cms.Controls.ListSummary.CopyTo 326
Ektron.Cms.Controls.ListSummary.Count 331
Ektron.Cms.Controls.ListSummary.Current 331
Ektron.Cms.Controls.ListSummary.Direction 332
Ektron.Cms.Controls.ListSummary.EkItems 332
Ektron.Cms.Controls.ListSummary.EnablePaging 332
Ektron.Cms.Controls.ListSummary.Fill 326
Ektron.Cms.Controls.ListSummary.FolderDescription 333
Ektron.Cms.Controls.ListSummary.FolderID 333
Ektron.Cms.Controls.ListSummary.FolderName 333
Ektron.Cms.Controls.ListSummary.FolderPath 333
Ektron.Cms.Controls.ListSummary.GetEnumerator 326
Ektron.Cms.Controls.ListSummary.GetHtml 334
Ektron.Cms.Controls.ListSummary.GetRssFeed 326
Ektron.Cms.Controls.ListSummary.GetView 327
Ektron.Cms.Controls.ListSummary.GetViewNames 327
Ektron.Cms.Controls.ListSummary.IncludeIcons 334
Ektron.Cms.Controls.ListSummary.IsSynchronized 334
Ektron.Cms.Controls.ListSummary.LinkTarget 335
Ektron.Cms.Controls.ListSummary.MarkupLanguage 335
Ektron.Cms.Controls.ListSummary.MaxResults 335
Ektron.Cms.Controls.ListSummary.MoveNext 327
Ektron.Cms.Controls.ListSummary.OrderByDirection 335
Ektron.Cms.Controls.ListSummary.OrderKey 336
Ektron.Cms.Controls.ListSummary.Random 336
Ektron.Cms.Controls.ListSummary.Recursive 336
Ektron.Cms.Controls.ListSummary.Reset 327
Ektron.Cms.Controls.ListSummary.SuppressWrapperTags 337
Ektron.Cms.Controls.ListSummary.SyncRoot 337
Ektron.Cms.Controls.ListSummary.this 328
Ektron.Cms.Controls.ListSummary.ToString 328
Ektron.Cms.Controls.ListSummary.Wrap 328
Ektron.Cms.Controls.ListSummary.XmlDoc 337
Ektron.Cms.Controls.Login 337
Ektron.Cms.Controls.Login.AutoAddType 338
Ektron.Cms.Controls.Login.AutoLogin 339
Ektron.Cms.Controls.Login.Fill 338
Ektron.Cms.Controls.Login.OnlyAllowMemberLogin 339
Ektron.Cms.Controls.Login.Password 339
Ektron.Cms.Controls.Login.PromptLogout 339
Ektron.Cms.Controls.Login.SuppressHelpButton 339
Ektron.Cms.Controls.Login.Text 340
Ektron.Cms.Controls.Login.UserName 340
Ektron.Cms.Controls.Map 340
Ektron.Cms.Controls.Map.Address 342
Ektron.Cms.Controls.Map.ContentId 343
Ektron.Cms.Controls.Map.CustomIcon 343
Ektron.Cms.Controls.Map.DateSearch 343
Ektron.Cms.Controls.Map.DisplayTextResult 343
Ektron.Cms.Controls.Map.DistanceUnit 344
Ektron.Cms.Controls.Map.EnableSearchResult 344
Ektron.Cms.Controls.Map.Fill 341
Ektron.Cms.Controls.Map.FolderId 344
Ektron.Cms.Controls.Map.ForceCategory 345
Ektron.Cms.Controls.Map.GeoControl 345
Ektron.Cms.Controls.Map.GetCallbackResult 341
Ektron.Cms.Controls.Map.Latitude 345
Ektron.Cms.Controls.Map.Longitude 346
Ektron.Cms.Controls.Map.MapCategory 346
Ektron.Cms.Controls.Map.MapProvider 346
Ektron.Cms.Controls.Map.MapStyle 346
Ektron.Cms.Controls.Map.MarkupLanguage 347
Ektron.Cms.Controls.Map.MinZoomLevel 347
Ektron.Cms.Controls.Map.PageSize 347
Ektron.Cms.Controls.Map.RaiseCallbackEvent 341
Ektron.Cms.Controls.Map.Recursive 347
Ektron.Cms.Controls.Map.SearchData 348
Ektron.Cms.Controls.Map.SelectedCategoryLogicalType 348
Ektron.Cms.Controls.Map.StartZoomLevel 348
Ektron.Cms.Controls.Map.Stylesheet 348

Ektron.Cms.Controls.Map.TemplateParamName 349
Ektron.Cms.Controls.Map.TemplateUserProfile 349
Ektron.Cms.Controls.Map.TypeControl 349
Ektron.Cms.Controls.Map.ZoomControl 349
Ektron.Cms.Controls.Membership 350
Ektron.Cms.Controls.Membership.AllowAvatarUpload 351
Ektron.Cms.Controls.Membership.AutoLogin 352
Ektron.Cms.Controls.Membership.BoardID 352
Ektron.Cms.Controls.Membership.DisplayMode 352
Ektron.Cms.Controls.Membership.DynamicParameter 353
Ektron.Cms.Controls.Membership.Fill 350
Ektron.Cms.Controls.Membership.JavascriptEditorHeight 353
Ektron.Cms.Controls.Membership.JavascriptEditorWidth 353
Ektron.Cms.Controls.Membership.RedirectFailedURL 354
Ektron.Cms.Controls.Membership.RedirectSuccessURL 354
Ektron.Cms.Controls.Membership.RegisterButtonImg 354
Ektron.Cms.Controls.Membership.RegisterButtonText 354
Ektron.Cms.Controls.Membership.ResetButtonImg 355
Ektron.Cms.Controls.Membership.ResetButtonText 355
Ektron.Cms.Controls.Membership.ResetPasswordMessageId 355
Ektron.Cms.Controls.Membership.SaveButtonImg 355
Ektron.Cms.Controls.Membership.SaveButtonText 356
Ektron.Cms.Controls.Membership.ShowExtended 356
Ektron.Cms.Controls.Membership.ShowTerms 356
Ektron.Cms.Controls.Membership.TaxonomyId 356
Ektron.Cms.Controls.Membership.Text 357
Ektron.Cms.Controls.Membership.UserExistsMessage 357
Ektron.Cms.Controls.Membership.UserID 357
Ektron.Cms.Controls.Membership.UserSuccessMessage 357
Ektron.Cms.Controls.Membership.UserUpdateSuccessMessage 358
Ektron.Cms.Controls.Menu 358
Ektron.Cms.Controls.Menu.CacheInterval 359
Ektron.Cms.Controls.Menu.DefaultMenuID 360
Ektron.Cms.Controls.Menu.DisplayXslt 360
Ektron.Cms.Controls.Menu.DynamicParameter 360
Ektron.Cms.Controls.Menu.Fill 358
Ektron.Cms.Controls.Menu.Stylesheet 361
Ektron.Cms.Controls.Menu.SuppressAddEdit 361
Ektron.Cms.Controls.Menu.Wrap 359
Ektron.Cms.Controls.MessageBoard 361
Ektron.Cms.Controls.MessageBoard.CacheInterval 363
Ektron.Cms.Controls.MessageBoard.DefaultObjectID 364
Ektron.Cms.Controls.MessageBoard.DisplayXslt 364
Ektron.Cms.Controls.MessageBoard.DynamicObjectParameter 364
Ektron.Cms.Controls.MessageBoard.EnablePaging 365
Ektron.Cms.Controls.MessageBoard.Fill 362
Ektron.Cms.Controls.MessageBoard.MarkupLanguage 365
Ektron.Cms.Controls.MessageBoard.MaxResults 366
Ektron.Cms.Controls.MessageBoard.Moderate 366
Ektron.Cms.Controls.MessageBoard.ObjectType 366
Ektron.Cms.Controls.MessageBoard.ProfileParamName 367
Ektron.Cms.Controls.MessageBoard.RaiseCallbackEvent 362
Ektron.Cms.Controls.MessageBoard.ShowMaxCharacter 367
Ektron.Cms.Controls.MessageBoard.Stylesheet 367
Ektron.Cms.Controls.MessageBoard.TemplateUserProfile 368
Ektron.Cms.Controls.MessageBoard.ToString 362
Ektron.Cms.Controls.MessageBoard.Wrap 362
Ektron.Cms.Controls.MessageBoard.XmlDoc 368
Ektron.Cms.Controls.Messaging 368
Ektron.Cms.Controls.Messaging.DisplayXslt 370
Ektron.Cms.Controls.Messaging.EnablePreSearch 370
Ektron.Cms.Controls.Messaging.Fill 369
Ektron.Cms.Controls.Messaging.FriendsOnly 371
Ektron.Cms.Controls.Messaging.GroupParamName 371
Ektron.Cms.Controls.Messaging.JavascriptEditorHeight 371
Ektron.Cms.Controls.Messaging.JavascriptEditorHTMLMode 372
Ektron.Cms.Controls.Messaging.JavascriptEditorToolbar 372
Ektron.Cms.Controls.Messaging.JavascriptEditorWidth 373
Ektron.Cms.Controls.Messaging.ProfileParamName 373
Ektron.Cms.Controls.Messaging.ProfileUrl 373
Ektron.Cms.Controls.Messaging.RecipientParamName 374
Ektron.Cms.Controls.Messaging.ShowSearchOptions 374
Ektron.Cms.Controls.Messaging.Stylesheet 374
Ektron.Cms.Controls.Messaging.ToString 369
Ektron.Cms.Controls.Messaging.UserSearchDisplayXslt 375

Ektron.Cms.Controls.Messaging.Wrap 369
Ektron.Cms.Controls.Messaging.XmlDoc 375
Ektron.Cms.Controls.MetaData 375
Ektron.Cms.Controls.MetaData.CacheInterval 376
Ektron.Cms.Controls.MetaData.DefaultContentID 377
Ektron.Cms.Controls.MetaData.DefaultItemList 377
Ektron.Cms.Controls.MetaData.DynamicParameter 377
Ektron.Cms.Controls.MetaData.Fill 376
Ektron.Cms.Controls.MetaData.GenerateDublinCore 378
Ektron.Cms.Controls.MetaDataList 378
Ektron.Cms.Controls.MetaDataList.CacheInterval 382
Ektron.Cms.Controls.MetaDataList.ContentType 383
Ektron.Cms.Controls.MetaDataList.CopyTo 379
Ektron.Cms.Controls.MetaDataList.Count 383
Ektron.Cms.Controls.MetaDataList.Current 383
Ektron.Cms.Controls.MetaDataList.EkItems 383
Ektron.Cms.Controls.MetaDataList.EnablePaging 384
Ektron.Cms.Controls.MetaDataList.ExactPhrase 384
Ektron.Cms.Controls.MetaDataList.Fill 379
Ektron.Cms.Controls.MetaDataList.FolderID 385
Ektron.Cms.Controls.MetaDataList.GetEnumerator 379
Ektron.Cms.Controls.MetaDataList.GetHtml 385
Ektron.Cms.Controls.MetaDataList.IncludeIcons 385
Ektron.Cms.Controls.MetaDataList.IsSynchronized 386
Ektron.Cms.Controls.MetaDataList.KeyWordName 386
Ektron.Cms.Controls.MetaDataList.KeyWordValue 386
Ektron.Cms.Controls.MetaDataList.KeyWordValueMatchAll 387
Ektron.Cms.Controls.MetaDataList.KeyWordValueSeparator 387
Ektron.Cms.Controls.MetaDataList.LinkTarget 387
Ektron.Cms.Controls.MetaDataList.MarkupLanguage 388
Ektron.Cms.Controls.MetaDataList.MaxNumber 388
Ektron.Cms.Controls.MetaDataList.MoveNext 380
Ektron.Cms.Controls.MetaDataList.OrderBy 388
Ektron.Cms.Controls.MetaDataList.Recursive 389
Ektron.Cms.Controls.MetaDataList.Reset 380
Ektron.Cms.Controls.MetaDataList.SortOrder 389
Ektron.Cms.Controls.MetaDataList.SuppressWrapperTags 390
Ektron.Cms.Controls.MetaDataList.SyncRoot 390
Ektron.Cms.Controls.MetaDataList.this 380
Ektron.Cms.Controls.MetaDataList.Wrap 380
Ektron.Cms.Controls.MetaDataList.XmlDoc 390
Ektron.Cms.Controls.MyAccount 390
Ektron.Cms.Controls.MyAccount.CustomPropertyID 392
Ektron.Cms.Controls.MyAccount.DefaultCountryId 392
Ektron.Cms.Controls.MyAccount.DisplayXslt 393
Ektron.Cms.Controls.MyAccount.EkItem 393
Ektron.Cms.Controls.MyAccount.Fill 391
Ektron.Cms.Controls.MyAccount.LoadingImage 393
Ektron.Cms.Controls.MyAccount.RedirectUrl 394
Ektron.Cms.Controls.MyAccount.Stylesheet 394
Ektron.Cms.Controls.MyAccount.ToString 391
Ektron.Cms.Controls.MyAccount.UseMsAjax 394
Ektron.Cms.Controls.MyAccount.Wrap 391
Ektron.Cms.Controls.OrderList 395
Ektron.Cms.Controls.OrderList.CacheInterval 396
Ektron.Cms.Controls.OrderList.DefaultUserID 397
Ektron.Cms.Controls.OrderList.DisplayXslt 397
Ektron.Cms.Controls.OrderList.DynamicOrderParameter 398
Ektron.Cms.Controls.OrderList.DynamicProductParameter 398
Ektron.Cms.Controls.OrderList.EkItem 398
Ektron.Cms.Controls.OrderList.Fill 395
Ektron.Cms.Controls.OrderList.PageSize 399
Ektron.Cms.Controls.OrderList.Stylesheet 399
Ektron.Cms.Controls.OrderList.TemplateProduct 399
Ektron.Cms.Controls.OrderList.ToString 396
Ektron.Cms.Controls.OrderList.Wrap 396
Ektron.Cms.Controls.PersonalizationManager 400
Ektron.Cms.Controls.PersonalizationManager.AddContentExitText 401
Ektron.Cms.Controls.PersonalizationManager.AddContentText 401
Ektron.Cms.Controls.PersonalizationManager.CurrentUserButtonText 402
Ektron.Cms.Controls.PersonalizationManager.Display 402
Ektron.Cms.Controls.PersonalizationManager.Fill 400
Ektron.Cms.Controls.PersonalizationManager.HeaderBackColor 402
Ektron.Cms.Controls.PersonalizationManager.HeaderText 403

Ektron.Cms.Controls.PersonalizationManager.PersonalizeExitText 403
Ektron.Cms.Controls.PersonalizationManager.PersonalizeScopeText 403
Ektron.Cms.Controls.PersonalizationManager.PersonalizeStartText 404
Ektron.Cms.Controls.PersonalizationManager.PublicUserButtonText 404
Ektron.Cms.Controls.PersonalizationManager.ResetLinkText 404
Ektron.Cms.Controls.PersonalizationManager.ResetLinkToolTip 405
Ektron.Cms.Controls.PersonalizationManager.UrlCompareMode 405
Ektron.Cms.Controls.PhotoGallery 406
Ektron.Cms.Controls.PhotoGallery.Fill 406
Ektron.Cms.Controls.Poll 406
Ektron.Cms.Controls.Poll.EnableAjax 408
Ektron.Cms.Controls.Poll.Fill 407
Ektron.Cms.Controls.Poll.PollID 408
Ektron.Cms.Controls.Poll.WrapPoll 407
Ektron.Cms.Controls.PostHistory 409
Ektron.Cms.Controls.PostHistory.BoardID 410
Ektron.Cms.Controls.PostHistory.CacheInterval 410
Ektron.Cms.Controls.PostHistory.Fill 409
Ektron.Cms.Controls.PostHistory.MaxNumber 411
Ektron.Cms.Controls.PostHistory.ToString 409
Ektron.Cms.Controls.PostHistory.URLPath 411
Ektron.Cms.Controls.PostHistory.UserID 411
Ektron.Cms.Controls.PostHistory.Wrap 410
Ektron.Cms.Controls.PostLoginEventArgs 412
Ektron.Cms.Controls.Product 412
Ektron.Cms.Controls.Product.CacheInterval 414
Ektron.Cms.Controls.Product.DefaultProductID 415
Ektron.Cms.Controls.Product.DisplayXslt 415
Ektron.Cms.Controls.Product.DynamicParameter 415
Ektron.Cms.Controls.Product.EkItems 416
Ektron.Cms.Controls.Product.Fill 413
Ektron.Cms.Controls.Product.ImageGallery 416
Ektron.Cms.Controls.Product.OverrideXslt 417
Ektron.Cms.Controls.Product.RedirectToBasketPage 413
Ektron.Cms.Controls.Product.ShowAddToCart 417
Ektron.Cms.Controls.Product.Stylesheet 417
Ektron.Cms.Controls.Product.TemplateCart 418
Ektron.Cms.Controls.Product.ToString 413
Ektron.Cms.Controls.Product.Wrap 413
Ektron.Cms.Controls.ProductList 418
Ektron.Cms.Controls.ProductList.DisplayXslt 420
Ektron.Cms.Controls.ProductList.DynamicParameter 421
Ektron.Cms.Controls.ProductList.EkItems 421
Ektron.Cms.Controls.ProductList.Fill 419
Ektron.Cms.Controls.ProductList.IdList 421
Ektron.Cms.Controls.ProductList.LoadingImage 422
Ektron.Cms.Controls.ProductList.PageSize 422
Ektron.Cms.Controls.ProductList.PageSpan 422
Ektron.Cms.Controls.ProductList.Recursive 423
Ektron.Cms.Controls.ProductList.SortMode 423
Ektron.Cms.Controls.ProductList.SourceId 424
Ektron.Cms.Controls.ProductList.SourceType 424
Ektron.Cms.Controls.ProductList.Stylesheet 424
Ektron.Cms.Controls.ProductList.TaxonomyDepth 425
Ektron.Cms.Controls.ProductList.TemplateProduct 425
Ektron.Cms.Controls.ProductList.ToString 419
Ektron.Cms.Controls.ProductList.UseMsAjax 426
Ektron.Cms.Controls.ProductList.Wrap 419
Ektron.Cms.Controls.ProductSearch 426
Ektron.Cms.Controls.ProductSearch.ButtonImgSrc 430
Ektron.Cms.Controls.ProductSearch.ButtonText 430
Ektron.Cms.Controls.ProductSearch.CacheInterval 431
Ektron.Cms.Controls.ProductSearch.CatalogId 431
Ektron.Cms.Controls.ProductSearch.CustomOrderBy 431
Ektron.Cms.Controls.ProductSearch.DisplayXslt 432
Ektron.Cms.Controls.ProductSearch.DynamicProductParameter 432
Ektron.Cms.Controls.ProductSearch.EnableAdvancedLink 432
Ektron.Cms.Controls.ProductSearch.EndCount 433
Ektron.Cms.Controls.ProductSearch.Fill 427
Ektron.Cms.Controls.ProductSearch.GetCallbackResult 427

Ektron.Cms.Controls.ProductSearch.IsInWorkArea 433
Ektron.Cms.Controls.ProductSearch.LinkTarget 433
Ektron.Cms.Controls.ProductSearch.LoadingImage 434
Ektron.Cms.Controls.ProductSearch.MaxCharacters 434
Ektron.Cms.Controls.ProductSearch.MaxTeaserLength 434
Ektron.Cms.Controls.ProductSearch.OrderBy 435
Ektron.Cms.Controls.ProductSearch.OrderDirection 435
Ektron.Cms.Controls.ProductSearch.ProductSearch 426
Ektron.Cms.Controls.ProductSearch.RaiseCallbackEvent 427
Ektron.Cms.Controls.ProductSearch.Recursive 436
Ektron.Cms.Controls.ProductSearch.ResultCount 436
Ektron.Cms.Controls.ProductSearch.ResultDataTable 436
Ektron.Cms.Controls.ProductSearch.ResultsPageSize 437
Ektron.Cms.Controls.ProductSearch.ResultTagId 437
Ektron.Cms.Controls.ProductSearch.ResultXml 438
Ektron.Cms.Controls.ProductSearch.SearchText 438
Ektron.Cms.Controls.ProductSearch.ShowCategories 438
Ektron.Cms.Controls.ProductSearch.ShowCustomSummary 438
Ektron.Cms.Controls.ProductSearch.ShowSearchBoxAlways 439
Ektron.Cms.Controls.ProductSearch.StartCount 439
Ektron.Cms.Controls.ProductSearch.Stylesheet 439
Ektron.Cms.Controls.ProductSearch.TaxCategories 440
Ektron.Cms.Controls.ProductSearch.TaxonomyOperator 440
Ektron.Cms.Controls.ProductSearch.TemplateCart 441
Ektron.Cms.Controls.ProductSearch.TemplateProduct 441
Ektron.Cms.Controls.ProductSearch.TextBoxSize 441
Ektron.Cms.Controls.ProductSearch.TimeTaken 442
Ektron.Cms.Controls.ProductSearch.Wrap 427
Ektron.Cms.Controls.ProductSearch.XmlDoc 442
Ektron.Cms.Controls.Recommendation 442
Ektron.Cms.Controls.Recommendation.CacheInterval 444
Ektron.Cms.Controls.Recommendation.DefaultProductID 445
Ektron.Cms.Controls.Recommendation.DisplayXslt 445
Ektron.Cms.Controls.Recommendation.DynamicCartParameter 445
Ektron.Cms.Controls.Recommendation.DynamicProductParameter 446
Ektron.Cms.Controls.Recommendation.EkItem 446
Ektron.Cms.Controls.Recommendation.Fill 443
Ektron.Cms.Controls.Recommendation.PageSize 447
Ektron.Cms.Controls.Recommendation.RecommendationType 447
Ektron.Cms.Controls.Recommendation.Stylesheet 447
Ektron.Cms.Controls.Recommendation.TemplateCart 448
Ektron.Cms.Controls.Recommendation.TemplateProduct 448
Ektron.Cms.Controls.Recommendation.ToString 443
Ektron.Cms.Controls.Recommendation.Wrap 443
Ektron.Cms.Controls.RssAggregator 448
Ektron.Cms.Controls.RssAggregator.CopyTo 449
Ektron.Cms.Controls.RssAggregator.Count 451
Ektron.Cms.Controls.RssAggregator.Current 452
Ektron.Cms.Controls.RssAggregator.EkItems 452
Ektron.Cms.Controls.RssAggregator.Fill 449
Ektron.Cms.Controls.RssAggregator.GetEnumerator 450
Ektron.Cms.Controls.RssAggregator.GetView 450
Ektron.Cms.Controls.RssAggregator.GetViewNames 450
Ektron.Cms.Controls.RssAggregator.IsSynchronized 452
Ektron.Cms.Controls.RssAggregator.LinkTarget 452
Ektron.Cms.Controls.RssAggregator.MaxResults 453
Ektron.Cms.Controls.RssAggregator.MoveNext 450
Ektron.Cms.Controls.RssAggregator.Reset 451
Ektron.Cms.Controls.RssAggregator.SyncRoot 453
Ektron.Cms.Controls.RssAggregator.this 451
Ektron.Cms.Controls.RssAggregator.URL 453
Ektron.Cms.Controls.RssItemBase 453
Ektron.Cms.Controls.RssItemBase.DatePosted 454
Ektron.Cms.Controls.RssItemBase.Description 454
Ektron.Cms.Controls.RssItemBase.Link 454
Ektron.Cms.Controls.RssItemBase.Title 454
Ektron.Cms.Controls.Search 455
Ektron.Cms.Controls.Search.ButtonImgSrc 460
Ektron.Cms.Controls.Search.ButtonText 460
Ektron.Cms.Controls.Search.ContentFieldsetLegend 461
Ektron.Cms.Controls.Search.CopyTo 456
Ektron.Cms.Controls.Search.Count 461
Ektron.Cms.Controls.Search.Current 461
Ektron.Cms.Controls.Search.Display 461
Ektron.Cms.Controls.Search.EkItems 462
Ektron.Cms.Controls.Search.EmptyResultMsg 462

Ektron.Cms.Controls.Search.EnableAdvancedLink 462
Ektron.Cms.Controls.Search.EnableAjax 462
Ektron.Cms.Controls.Search.EnableArchived 463
Ektron.Cms.Controls.Search.EnableArchivedBtn 463
Ektron.Cms.Controls.Search.EnableContent 463
Ektron.Cms.Controls.Search.EnableContentBtn 464
Ektron.Cms.Controls.Search.EnableContentFieldset 464
Ektron.Cms.Controls.Search.EnableDMSAssets 464
Ektron.Cms.Controls.Search.EnableDMSAssetsBtn 465
Ektron.Cms.Controls.Search.EnableFolderFieldset 465
Ektron.Cms.Controls.Search.EnableForms 465
Ektron.Cms.Controls.Search.EnableFormsBtn 465
Ektron.Cms.Controls.Search.EnableLibFiles 466
Ektron.Cms.Controls.Search.EnableLibFilesBtn 466
Ektron.Cms.Controls.Search.EnableLibHyperlinks 466
Ektron.Cms.Controls.Search.EnableLibHyperlinksBtn 467
Ektron.Cms.Controls.Search.EnableLibImages 467
Ektron.Cms.Controls.Search.EnableLibImagesBtn 467
Ektron.Cms.Controls.Search.EnableLibraryFieldset 468
Ektron.Cms.Controls.Search.Fill 456
Ektron.Cms.Controls.Search.FolderFieldsetLegend 468
Ektron.Cms.Controls.Search.FolderID 468
Ektron.Cms.Controls.Search.FolderPath 469
Ektron.Cms.Controls.Search.GetEnumerator 456
Ektron.Cms.Controls.Search.IncludeIcons 469
Ektron.Cms.Controls.Search.IsSynchronized 469
Ektron.Cms.Controls.Search.LibraryFieldsetLegend 469
Ektron.Cms.Controls.Search.LinkTarget 470
Ektron.Cms.Controls.Search.LoadSearch 456
Ektron.Cms.Controls.Search.MaxCharacters 470
Ektron.Cms.Controls.Search.MaxResults 470
Ektron.Cms.Controls.Search.MaxTeaserLength 471
Ektron.Cms.Controls.Search.MoveNext 457
Ektron.Cms.Controls.Search.OrderBy 471
Ektron.Cms.Controls.Search.OrderDirection 472
Ektron.Cms.Controls.Search.PostResultsAt 472
Ektron.Cms.Controls.Search.Recursive 472
Ektron.Cms.Controls.Search.RemoveTeaserHtml 472
Ektron.Cms.Controls.Search.Reset 457
Ektron.Cms.Controls.Search.ResultTagId 473
Ektron.Cms.Controls.Search.Search 455
Ektron.Cms.Controls.Search.SearchDisplayUI 457
Ektron.Cms.Controls.Search.SearchText 473
Ektron.Cms.Controls.Search.SearchType 473
Ektron.Cms.Controls.Search.ShowDate 473
Ektron.Cms.Controls.Search.ShowExtendedSearch 474
Ektron.Cms.Controls.Search.ShowSearchBoxAlways 474
Ektron.Cms.Controls.Search.ShowSearchOptions 475
Ektron.Cms.Controls.Search.Stylesheet 475
Ektron.Cms.Controls.Search.SyncRoot 475
Ektron.Cms.Controls.Search.TargetFolderDescriptionList 475
Ektron.Cms.Controls.Search.TextBoxSize 476
Ektron.Cms.Controls.Search.this 457
Ektron.Cms.Controls.Search.Wrap 457
Ektron.Cms.Controls.Search.XmlDoc 476
Ektron.Cms.Controls.Sitemap 476
Ektron.Cms.Controls.Sitemap.CacheInterval 477
Ektron.Cms.Controls.Sitemap.ClassName 478
Ektron.Cms.Controls.Sitemap.DisplayType 478
Ektron.Cms.Controls.Sitemap.DisplayXslt 479
Ektron.Cms.Controls.Sitemap.Fill 477
Ektron.Cms.Controls.Sitemap.FlatTopLevel 479
Ektron.Cms.Controls.Sitemap.FolderID 479
Ektron.Cms.Controls.Sitemap.MaxLevel 480
Ektron.Cms.Controls.Sitemap.StartingLevel 480
Ektron.Cms.Controls.Sitemap.Stylesheet 480
Ektron.Cms.Controls.Sitemap.XmlDoc 480
Ektron.Cms.Controls.SmartMenu 481
Ektron.Cms.Controls.SmartMenu.AutoCollapseBranches 482
Ektron.Cms.Controls.SmartMenu.DisplayXslt 482
Ektron.Cms.Controls.SmartMenu.Enable508Compliance 482
Ektron.Cms.Controls.SmartMenu.Enable508NavigationHeadings 483
Ektron.Cms.Controls.SmartMenu.EnableMouseOverPopUp 483
Ektron.Cms.Controls.SmartMenu.EnableSmartOpen 483
Ektron.Cms.Controls.SmartMenu.EnableSubmenuLinkPathMatching 484
Ektron.Cms.Controls.SmartMenu.EnableSubmenuTemplateMat

ching
484
Ektron.Cms.Controls.SmartMenu.LaunchLinksInNewWindow
484
Ektron.Cms.Controls.SmartMenu.ShowRootFolder 485
Ektron.Cms.Controls.SmartMenu.SingleEditButton 485
Ektron.Cms.Controls.SmartMenu.StartCollapsed 486
Ektron.Cms.Controls.SmartMenu.StartWithRootCollapsed 486
Ektron.Cms.Controls.SmartMenu.Stylesheet 486
Ektron.Cms.Controls.SocialBar 487
Ektron.Cms.Controls.SocialBar.CacheInterval 489
Ektron.Cms.Controls.SocialBar.DefaultObjectID 489
Ektron.Cms.Controls.SocialBar.DisplayXslt 489
Ektron.Cms.Controls.SocialBar.DynamicObjectParameter 490
Ektron.Cms.Controls.SocialBar.EkItems 490
Ektron.Cms.Controls.SocialBar.Fill 487
Ektron.Cms.Controls.SocialBar.GetCallbackResult 487
Ektron.Cms.Controls.SocialBar.IncludeIcons 490
Ektron.Cms.Controls.SocialBar.InviteURL 491
Ektron.Cms.Controls.SocialBar.Items 491
Ektron.Cms.Controls.SocialBar.MarkupLanguage 491
Ektron.Cms.Controls.SocialBar.MessagingURL 492
Ektron.Cms.Controls.SocialBar.ObjectType 492
Ektron.Cms.Controls.SocialBar.Stylesheet 492
Ektron.Cms.Controls.SocialBar.SuppressWrapperTags 493
Ektron.Cms.Controls.SocialBar.ToString 488
Ektron.Cms.Controls.SocialBar.Wrap 488
Ektron.Cms.Controls.SocialBar.XmlDoc 493
Ektron.Cms.Controls.TagCloud 493
Ektron.Cms.Controls.TagCloud.CacheInterval 495
Ektron.Cms.Controls.TagCloud.DisplayXslt 496
Ektron.Cms.Controls.TagCloud.DynamicParameter 496
Ektron.Cms.Controls.TagCloud.Fill 494
Ektron.Cms.Controls.TagCloud.MarkupLanguage 496
Ektron.Cms.Controls.TagCloud.MaxTagsDisplayed 496
Ektron.Cms.Controls.TagCloud.OrderBy 497
Ektron.Cms.Controls.TagCloud.OrderByDirection 497
Ektron.Cms.Controls.TagCloud.Stylesheet 498
Ektron.Cms.Controls.TagCloud.TagTemplate 498
Ektron.Cms.Controls.TagCloud.TagTemplateTarget 499
Ektron.Cms.Controls.TagCloud.TagType 499
Ektron.Cms.Controls.TagCloud.ToString 494
Ektron.Cms.Controls.TagCloud.Wrap 494
Ektron.Cms.Controls.TagCloud.XmlDoc 499
Ektron.Cms.Controls.UserProfile 499
Ektron.Cms.Controls.UserProfile.AvatarHeight 501
Ektron.Cms.Controls.UserProfile.AvatarWidth 501
Ektron.Cms.Controls.UserProfile.CacheInterval 502
Ektron.Cms.Controls.UserProfile.DefaultUserID 502
Ektron.Cms.Controls.UserProfile.DisplayXslt 502
Ektron.Cms.Controls.UserProfile.DynamicParameter 503
Ektron.Cms.Controls.UserProfile.EnableEmailDisplay 503
Ektron.Cms.Controls.UserProfile.Fill 500
Ektron.Cms.Controls.UserProfile.Stylesheet 503
Ektron.Cms.Controls.UserProfile.TagTemplate 504
Ektron.Cms.Controls.UserProfile.TagTemplateTarget 504
Ektron.Cms.Controls.UserProfile.TaxonomyId 504
Ektron.Cms.Controls.UserProfile.ToString 500
Ektron.Cms.Controls.UserProfile.Wrap 500
Ektron.Cms.Controls.UserProfile.XmlDoc 505
Ektron.Cms.Controls.WebSearch 505
Ektron.Cms.Controls.WebSearch.ButtonImgSrc 510
Ektron.Cms.Controls.WebSearch.ButtonText 510
Ektron.Cms.Controls.WebSearch.CustomOrderBy 510
Ektron.Cms.Controls.WebSearch.CustomSearch 511
Ektron.Cms.Controls.WebSearch.DisableForumSearch 511
Ektron.Cms.Controls.WebSearch.DisplayXslt 511
Ektron.Cms.Controls.WebSearch.DynamicContentTemplate 512
Ektron.Cms.Controls.WebSearch.EnableAdvancedLink 512
Ektron.Cms.Controls.WebSearch.EndCount 512
Ektron.Cms.Controls.WebSearch.Fill 506
Ektron.Cms.Controls.WebSearch.FolderID 512
Ektron.Cms.Controls.WebSearch.GetCallbackResult 506
Ektron.Cms.Controls.WebSearch.IsInWorkArea 513
Ektron.Cms.Controls.WebSearch.LinkTarget 513
Ektron.Cms.Controls.WebSearch.LoadSearch 506
Ektron.Cms.Controls.WebSearch.MarkupLanguage 514
Ektron.Cms.Controls.WebSearch.MaxCharacters 514
Ektron.Cms.Controls.WebSearch.MaxTeaserLength 514

Ektron.Cms.Controls.WebSearch.MultipleFolders 515
Ektron.Cms.Controls.WebSearch.OrderBy 515
Ektron.Cms.Controls.WebSearch.OrderDirection 515
Ektron.Cms.Controls.WebSearch.RaiseCallbackEvent 506
Ektron.Cms.Controls.WebSearch.Recursive 516
Ektron.Cms.Controls.WebSearch.RemoveTeaserHtml 516
Ektron.Cms.Controls.WebSearch.ResultCount 516
Ektron.Cms.Controls.WebSearch.ResultDataTable 517
Ektron.Cms.Controls.WebSearch.ResultsPageSize 517
Ektron.Cms.Controls.WebSearch.ResultTagId 518
Ektron.Cms.Controls.WebSearch.ResultXml 518
Ektron.Cms.Controls.WebSearch.SearchFor 518
Ektron.Cms.Controls.WebSearch.SearchSynonyms 519
Ektron.Cms.Controls.WebSearch.SearchText 519
Ektron.Cms.Controls.WebSearch.ShowCategories 520
Ektron.Cms.Controls.WebSearch.ShowCustomSummary 520
Ektron.Cms.Controls.WebSearch.ShowSearchBoxAlways 520
Ektron.Cms.Controls.WebSearch.ShowSearchOptions 521
Ektron.Cms.Controls.WebSearch.ShowSuggestedResults 521
Ektron.Cms.Controls.WebSearch.StartCount 522
Ektron.Cms.Controls.WebSearch.Stylesheet 522
Ektron.Cms.Controls.WebSearch.TaxCategories 522
Ektron.Cms.Controls.WebSearch.TaxonomyOperator 522
Ektron.Cms.Controls.WebSearch.TextBoxSize 523
Ektron.Cms.Controls.WebSearch.TimeTaken 523
Ektron.Cms.Controls.WebSearch.WebSearch 505
Ektron.Cms.Controls.WebSearch.Wrap 507
Ektron.Cms.Controls.WebSearch.XmlDoc 523
Ektron.Cms.Extensibility 524
Ektron.Cms.Extensibility namespace 524
Classes 525
Ektron.Cms.Extensibility.ActionRecipientInfo 526
Ektron.Cms.Extensibility.ActionRecipientInfo.ActionRecipientInfo 526
Ektron.Cms.Extensibility.ActionRecipientInfo.DisplayName 526
Ektron.Cms.Extensibility.ActionRecipientInfo.Email 527
Ektron.Cms.Extensibility.ActionRecipientInfo.FirstName 527
Ektron.Cms.Extensibility.ActionRecipientInfo.IsUser 527
Ektron.Cms.Extensibility.ActionRecipientInfo.LastName 527
Ektron.Cms.Extensibility.ActionRecipientInfo.UserId 527
Ektron.Cms.Extensibility.ActionRequestInfo 527
Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestInfo 528
Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestType 528
Ektron.Cms.Extensibility.ActionRequestInfo.ActionRequestType enumeration 528
Ektron.Cms.Extensibility.ActionRequestInfo.ErrorMessage 529
Ektron.Cms.Extensibility.ActionRequestInfo.Id 529
Ektron.Cms.Extensibility.ActionRequestInfo.Recipients 529
Ektron.Cms.Extensibility.ActionRequestInfo.SenderId 529
Ektron.Cms.Extensibility.ActionRequestInfo.Type 529
Ektron.Cms.Extensibility.ContentInfo 530
Ektron.Cms.Extensibility.ContentInfo.AssetID 530
Ektron.Cms.Extensibility.ContentInfo.Comments 531
Ektron.Cms.Extensibility.ContentInfo.ErrorMessage 531
Ektron.Cms.Extensibility.ContentInfo.FileAndPath 531
Ektron.Cms.Extensibility.ContentInfo.FileExtension 531
Ektron.Cms.Extensibility.ContentInfo.FileName 531
Ektron.Cms.Extensibility.ContentInfo.FolderID 531
Ektron.Cms.Extensibility.ContentInfo.Html 532
Ektron.Cms.Extensibility.ContentInfo.ID 532
Ektron.Cms.Extensibility.ContentInfo.Language 532
Ektron.Cms.Extensibility.ContentInfo.MimeType 532
Ektron.Cms.Extensibility.ContentInfo.Status 532
Ektron.Cms.Extensibility.ContentInfo.Title 532
Ektron.Cms.Extensibility.ContentInfo.Type 533
Ektron.Cms.Extensibility.ContentInfo.UserID 533
Ektron.Cms.Extensibility.ExtensionAuthor 533
Ektron.Cms.Extensibility.ExtensionAuthor.Author 534
Ektron.Cms.Extensibility.ExtensionAuthor.ExtensionAuthor 533
Ektron.Cms.Extensibility.ExtensionAuthor.ToString 534
Ektron.Cms.Extensibility.ExtensionDescription 534
Ektron.Cms.Extensibility.ExtensionDescription.Desc 535
Ektron.Cms.Extensibility.ExtensionDescription.ExtensionDescription 535
Ektron.Cms.Extensibility.ExtensionDescription.ToString 535
Ektron.Cms.Extensibility.ExtensionEvent 535
Ektron.Cms.Extensibility.ExtensionEvent.Content 552

Ektron.Cms.Extensibility.ExtensionEvent.Folder 553
Ektron.Cms.Extensibility.ExtensionEvent.Form 553
Ektron.Cms.Extensibility.ExtensionEvent.GetActiveDirectoryUserProxy 538
Ektron.Cms.Extensibility.ExtensionEvent.GetAssetProxy 538
Ektron.Cms.Extensibility.ExtensionEvent.GetBlogProxy 538
Ektron.Cms.Extensibility.ExtensionEvent.GetBusinessRulesProxy 539
Ektron.Cms.Extensibility.ExtensionEvent.GetCalendarEventTypeProxy 539
Ektron.Cms.Extensibility.ExtensionEvent.GetCalendarProxy 539
Ektron.Cms.Extensibility.ExtensionEvent.GetCommunityGroupProxy 539
Ektron.Cms.Extensibility.ExtensionEvent.GetContentProxy 539
Ektron.Cms.Extensibility.ExtensionEvent.GetContentRatingProxy 540
Ektron.Cms.Extensibility.ExtensionEvent.GetCustomFieldsProxy 540
Ektron.Cms.Extensibility.ExtensionEvent.GetFavoritesProxy 540
Ektron.Cms.Extensibility.ExtensionEvent.GetFlagProxy 540
Ektron.Cms.Extensibility.ExtensionEvent.GetFolderProxy 541
Ektron.Cms.Extensibility.ExtensionEvent.GetFontProxy 541
Ektron.Cms.Extensibility.ExtensionEvent.GetFormProxy 541
Ektron.Cms.Extensibility.ExtensionEvent.GetFriendsProxy 541
Ektron.Cms.Extensibility.ExtensionEvent.GetLibraryProxy 542
Ektron.Cms.Extensibility.ExtensionEvent.GetMessageBoardProxy 542
Ektron.Cms.Extensibility.ExtensionEvent.GetMetadataProxy 542
Ektron.Cms.Extensibility.ExtensionEvent.GetObjectStateAsXml 542
Ektron.Cms.Extensibility.ExtensionEvent.GetPermissionsProxy 542
Ektron.Cms.Extensibility.ExtensionEvent.GetSearchManagerProxy 543
Ektron.Cms.Extensibility.ExtensionEvent.GetSiteMapProxy 543
Ektron.Cms.Extensibility.ExtensionEvent.GetSiteProxy 543
Ektron.Cms.Extensibility.ExtensionEvent.GetTagsProxy 543
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskCategoryProxy 544
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskCategoryTypeProxy 544
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskProxy 544
Ektron.Cms.Extensibility.ExtensionEvent.GetTaxonomyProxy 544
Ektron.Cms.Extensibility.ExtensionEvent.GetThreadedDiscussionProxy 545
Ektron.Cms.Extensibility.ExtensionEvent.GetUserProxy 545
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterActionRequestSent 545
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentAdd 545
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentDelete 546
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentEdit 546
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentFlagged 546
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderAdd 546
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderDelete 546
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderEdit 547
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFormSubmit 547
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterPublish 547
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterReviewRejected 547
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserAdd 548
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserDelete 548
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserEdit 548
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserLogin 548
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeActionRequestSent 549
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentAdd 549
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentDelete 549

Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentEdit 549
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentFlagged 550
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderAdd 550
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderDelete 550
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderEdit 550
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFormSubmit 550
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforePublish 551
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeReviewRejected 551
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserAdd 551
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserDelete 551
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserEdit 552
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserLogin 552
Ektron.Cms.Extensibility.ExtensionEvent.Site 553
Ektron.Cms.Extensibility.ExtensionEvent.User 553
Ektron.Cms.Extensibility.FlagInfo 553
Ektron.Cms.Extensibility.FlagInfo.ContentHtml 554
Ektron.Cms.Extensibility.FlagInfo.ContentID 554
Ektron.Cms.Extensibility.FlagInfo.ContentLanguage 555
Ektron.Cms.Extensibility.FlagInfo.ContentTitle 555
Ektron.Cms.Extensibility.FlagInfo.EntryId 555
Ektron.Cms.Extensibility.FlagInfo.ErrorMessage 555
Ektron.Cms.Extensibility.FlagInfo.FlagComment 555
Ektron.Cms.Extensibility.FlagInfo.FlagDate 555
Ektron.Cms.Extensibility.FlagInfo.FlagId 556
Ektron.Cms.Extensibility.FlagInfo.FlagName 556
Ektron.Cms.Extensibility.FlagInfo.UserAvatar 556
Ektron.Cms.Extensibility.FlagInfo.UserDisplayName 556
Ektron.Cms.Extensibility.FlagInfo.UserFirstName 556
Ektron.Cms.Extensibility.FlagInfo.UserID 556
Ektron.Cms.Extensibility.FlagInfo.UserLastName 557
Ektron.Cms.Extensibility.FlagInfo.UserName 557
Ektron.Cms.Extensibility.FlagInfo.VisitorID 557
Ektron.Cms.Extensibility.FolderInfo 557
Ektron.Cms.Extensibility.FolderInfo.Description 558
Ektron.Cms.Extensibility.FolderInfo.ErrorMessage 558
Ektron.Cms.Extensibility.FolderInfo.ID 558
Ektron.Cms.Extensibility.FolderInfo.Name 558
Ektron.Cms.Extensibility.FolderInfo.ParentID 558
Ektron.Cms.Extensibility.FolderInfo.ReplicationState 558
Ektron.Cms.Extensibility.FolderInfo.StyleSheet 559
Ektron.Cms.Extensibility.FolderInfo.Type 559
Ektron.Cms.Extensibility.FormField 559
Ektron.Cms.Extensibility.FormField.FieldName 560
Ektron.Cms.Extensibility.FormField.FieldValue 560
Ektron.Cms.Extensibility.FormField.FormField 559
Ektron.Cms.Extensibility.FormInfo 560
Ektron.Cms.Extensibility.FormInfo.ErrorMessage 561
Ektron.Cms.Extensibility.FormInfo.FormDescription 561
Ektron.Cms.Extensibility.FormInfo.FormFields 561
Ektron.Cms.Extensibility.FormInfo.FormTitle 561
Ektron.Cms.Extensibility.FormInfo.this 560
Ektron.Cms.Extensibility.MessageBoardInfo 562
Ektron.Cms.Extensibility.MessageBoardInfo.ErrorMessage 562
Ektron.Cms.Extensibility.MessageBoardInfo.MessageBoardInfo 562
Ektron.Cms.Extensibility.RequestInfo 563
Ektron.Cms.Extensibility.ReviewInfo 563
Ektron.Cms.Extensibility.ReviewInfo.AverageRating 564
Ektron.Cms.Extensibility.ReviewInfo.ContentHtml 564
Ektron.Cms.Extensibility.ReviewInfo.ContentID 564
Ektron.Cms.Extensibility.ReviewInfo.ContentLanguage 564
Ektron.Cms.Extensibility.ReviewInfo.ContentTitle 564
Ektron.Cms.Extensibility.ReviewInfo.ErrorMessage 565
Ektron.Cms.Extensibility.ReviewInfo.ID 565
Ektron.Cms.Extensibility.ReviewInfo.NumberofRatings 565
Ektron.Cms.Extensibility.ReviewInfo.Rating 565
Ektron.Cms.Extensibility.ReviewInfo.RatingDate 565
Ektron.Cms.Extensibility.ReviewInfo.State 565
Ektron.Cms.Extensibility.ReviewInfo.UserAvatar 566
Ektron.Cms.Extensibility.ReviewInfo.UserComments 566
Ektron.Cms.Extensibility.ReviewInfo.UserDisplayName 566
Ektron.Cms.Extensibility.ReviewInfo.UserFirstName 566

Ektron.Cms.Extensibility.ReviewInfo.UserID 566
Ektron.Cms.Extensibility.ReviewInfo.UserLastName 566
Ektron.Cms.Extensibility.ReviewInfo.UserName 567
Ektron.Cms.Extensibility.ReviewInfo.VisitorID 567
Ektron.Cms.Extensibility.SiteInfo 567
Ektron.Cms.Extensibility.SiteInfo.AppPath 567
Ektron.Cms.Extensibility.SiteInfo.IsSSL 568
Ektron.Cms.Extensibility.SiteInfo.Name 568
Ektron.Cms.Extensibility.SiteInfo.PhysicalSitePath 568
Ektron.Cms.Extensibility.SiteInfo.Port 568
Ektron.Cms.Extensibility.SiteInfo.Url 568
Ektron.Cms.Extensibility.UserInfo 568
Ektron.Cms.Extensibility.UserInfo.Addres 569
Ektron.Cms.Extensibility.UserInfo.DisplayName 569
Ektron.Cms.Extensibility.UserInfo.Editor 570
Ektron.Cms.Extensibility.UserInfo.EmailAddress 570
Ektron.Cms.Extensibility.UserInfo.ErrorMessage 570
Ektron.Cms.Extensibility.UserInfo.FirstName 570
Ektron.Cms.Extensibility.UserInfo.ID 570
Ektron.Cms.Extensibility.UserInfo.Language 570
Ektron.Cms.Extensibility.UserInfo.LastName 571
Ektron.Cms.Extensibility.UserInfo.Latitude 571
Ektron.Cms.Extensibility.UserInfo.Longitude 571
Ektron.Cms.Extensibility.UserInfo.Notifications 571
Ektron.Cms.Extensibility.UserInfo.Type 571
Ektron.Cms.Extensibility.UserInfo.UserName 571
Ektron.Cms.FormData 1394
Ektron.Cms.FormData.Autofill 1395
Ektron.Cms.FormData.Description 1396
Ektron.Cms.FormData.MailCc 1396
Ektron.Cms.FormData.MailFrom 1396
Ektron.Cms.FormData.MailPreamble 1396
Ektron.Cms.FormData.MailSubject 1397
Ektron.Cms.FormData.MailTo 1397
Ektron.Cms.FormData.PollRevision 1397
Ektron.Cms.FormData.SendXmlPacket 1397
Ektron.Cms.FormData.StoreDataTo 1398
Ektron.Cms.FormData.Validation 1398
Ektron.Cms.FormFieldDataItem 1399
Ektron.Cms.FormFieldDataItem.DataValue 1399
Ektron.Cms.FormFieldDataItem.FieldName 1399
Ektron.Cms.FormFieldDatalist 1400
Ektron.Cms.FormFieldDatalist.CaptionXPath 1400
Ektron.Cms.FormFieldDatalist.DatalistItems 1401
Ektron.Cms.FormFieldDatalist.DataNamespaces 1401
Ektron.Cms.FormFieldDatalist.DataSelect 1401
Ektron.Cms.FormFieldDatalist.DataSrc 1401
Ektron.Cms.FormFieldDatalist.Language 1402
Ektron.Cms.FormFieldDatalist.Name 1402
Ektron.Cms.FormFieldDatalist.ValueXPath 1402
Ektron.Cms.FormFieldDatalistItem 1402
Ektron.Cms.FormFieldDatalistItem.DataValue 1403
Ektron.Cms.FormFieldDatalistItem.DisplayValue 1403
Ektron.Cms.FormFieldDatalistItem.IsDefault 1404
Ektron.Cms.FormFieldDatalistItem.IsEnabled 1404
Ektron.Cms.FormFieldDefinition 1404
Ektron.Cms.FormFieldDefinition.Basetype 1405
Ektron.Cms.FormFieldDefinition.ContentType 1405
Ektron.Cms.FormFieldDefinition.DatalistName 1406
Ektron.Cms.FormFieldDefinition.Datatype 1406
Ektron.Cms.FormFieldDefinition.DisplayName 1406
Ektron.Cms.FormFieldDefinition.FieldName 1407
Ektron.Cms.FormFieldDefinition.Language 1407
Ektron.Cms.FormFieldDefinition.XPath 1407
Ektron.Cms.FormFieldList 1407
Ektron.Cms.FormFieldList.Datalists 1408
Ektron.Cms.FormFieldList.Fields 1408
Ektron.Cms.FormRequestData 1409
Ektron.Cms.FormRequestData.AutoFill 1410
Ektron.Cms.FormRequestData.FieldList 1410
Ektron.Cms.FormRequestData.Id 1410
Ektron.Cms.FormRequestData.IsPoll 1410
Ektron.Cms.FormRequestData.Language 1411
Ektron.Cms.FormRequestData.PostBackData 1411
Ektron.Cms.FormRequestData.PreFillFielddata 1411
Ektron.Cms.FormRequestData.TagRequired 1412
Ektron.Cms.FormSubmittedData 1412
Ektron.Cms.FormSubmittedData.DataItems 1413

Ektron.Cms.FormSubmittedData.DateSubmitted 1413
Ektron.Cms.FormSubmittedData.FormDataID 1413
Ektron.Cms.FormSubmittedData.UserId 1413
Ektron.Cms.FormSubmittedData.UserInfo 1414
Ektron.Cms.ImageMediaData 1283
Ektron.Cms.ImageMediaData.Thumbnails 1284
Ektron.Cms.MediaData 1288
Ektron.Cms.MediaData.Alt 1289
Ektron.Cms.MediaData.FileName 1289
Ektron.Cms.MediaData.FilePath 1289
Ektron.Cms.MediaData.Height 1289
Ektron.Cms.MediaData.Id 1289
Ektron.Cms.MediaData.MimeType 1290
Ektron.Cms.MediaData.Sequence 1290
Ektron.Cms.MediaData.Title 1290
Ektron.Cms.MediaData.Width 1290
Ektron.Cms.MediaGalleryData 1291
Ektron.Cms.MediaGalleryData.Images 1291
Ektron.Cms.MediaGalleryData.ObjectId 1291
Ektron.Cms.TaxonomyRequest 1414
Ektron.Cms.TaxonomyRequest.Depth 1415
Ektron.Cms.TaxonomyRequest.FolderId 1415
Ektron.Cms.TaxonomyRequest.IncludeItems 1415
Ektron.Cms.TaxonomyRequest.Preview 1416
Ektron.Cms.TaxonomyRequest.ReadCount 1416
Ektron.Cms.TaxonomyRequest.TaxonomyIdList 1416
Ektron.Cms.TaxonomyRequest.TaxonomyItemType 1417
Ektron.Cms.TaxonomyRequest.TaxonomyPath 1417
Ektron.Cms.ThumbnailData 1343
Ektron.Cms.ThumbnailData.Height 1344
Ektron.Cms.ThumbnailData.Width 1344
Ektron.Cms.ThumbnailDefaultData 1344
Ektron.Cms.ThumbnailDefaultData.FileNameFormat 1345
Ektron.Cms.ThumbnailDefaultData.Height 1345
Ektron.Cms.ThumbnailDefaultData.Id 1345
Ektron.Cms.ThumbnailDefaultData.MimeType 1345
Ektron.Cms.ThumbnailDefaultData.Title 1345
Ektron.Cms.ThumbnailDefaultData.Width 1346
Ektron.Cms.UriFavorite 1417
Ektron.Cms.UriFavorite.FavoriteDesc 1418
Ektron.Cms.UriFavorite.FavoriteId 1418
Ektron.Cms.UriFavorite.FavoriteName 1418
Ektron.Cms.UriFavorite.FavoriteUri 1419
Ektron.Cms.UriFavorite.UserId 1419
Ektron.Cms.UserData 1419
Ektron.Cms.UserData.AcceptedTerms 1420
Ektron.Cms.UserData.AdUserName 1421
Ektron.Cms.UserData.DateCreated 1421
Ektron.Cms.UserData.GroupId 1421
Ektron.Cms.UserData.GroupName 1421
Ektron.Cms.UserData.IsMemberShip 1422
Ektron.Cms.UserData.LanguageId 1422
Ektron.Cms.UserData.LanguageName 1422
Ektron.Cms.UserData.LastLoginDate 1423
Ektron.Cms.UserData.LoginAttempts 1423
Ektron.Cms.UserData.Rank 1423
Ektron.Cms.UserData.Signature 1423
Ektron.Cms.UserData.UserPreference 1424
Ektron.Cms.UserRequestData 1424
Ektron.Cms.UserRequestData.Group 1425
Ektron.Cms.UserRequestData.OrderBy 1425
Ektron.Cms.UserRequestData.OrderDirection 1426
Ektron.Cms.UserRequestData.RequiredFlag 1426
Ektron.Cms.UserRequestData.SearchText 1426
Ektron.Cms.UserRequestData.Type 1427
Ektron.Services 955
Ektron.Services namespace 955
Classes 973
Ektron.Services.ActiveDirectoryUser 974
Ektron.Services.ActiveDirectoryUser.AddADGroupToCMS 975
Ektron.Services.ActiveDirectoryUser.DeSynchUserGroups 975
Ektron.Services.ActiveDirectoryUser.DeSynchUsers 975
Ektron.Services.ActiveDirectoryUser.GetADMapping 976
Ektron.Services.ActiveDirectoryUser.GetADStatus 976
Ektron.Services.ActiveDirectoryUser.GetAvailableADGroups 977
Ektron.Services.ActiveDirectoryUser.GetAvailableADUsers 977
Ektron.Services.ActiveDirectoryUser.GetCMSGroupsToSync 978

Ektron.Services.ActiveDirectoryUser.GetCMSRelationshipsToSync 978

Ektron.Services.ActiveDirectoryUser.GetCMSUsersToSync 979

Ektron.Services.ActiveDirectoryUser.GetDomains 979

Ektron.Services.ActiveDirectoryUser.MapCMSUserGroupToAD 980

Ektron.Services.ActiveDirectoryUser.MapCMSUserToAD 980

Ektron.Services.ActiveDirectoryUser.UpdateADMapping 981

Ektron.Services.ActiveDirectoryUser.UpdateADUsersGroups 981

Ektron.Services.ActiveDirectoryUser.UpdateUsersInfoFromAD 981

Ektron.Services.Asset 982

Ektron.Services.Asset.GetAssetMgtConfigInfo 983

Ektron.Services.Asset.GetAssetMgtInfo 983

Ektron.Services.Asset.GetAssetSupertypes 983

Ektron.Services.Asset.GetEditSnippet 984

Ektron.Services.Asset.GetMyAssetMgtID 984

Ektron.Services.Asset.MakeCopyDMSAsset 984

Ektron.Services.Asset.SetAssetMgtConfigInfo 985

Ektron.Services.Blog 985

Ektron.Services.Blog.AddBlog 986

Ektron.Services.Blog.AddCommentForPost 987

Ektron.Services.Blog.AddPost 987

Ektron.Services.Blog.DeleteBlog 988

Ektron.Services.Blog.DeleteBlogPostComment 988

Ektron.Services.Blog.DeletePost 988

Ektron.Services.Blog.GetBlankBlogPostData 989

Ektron.Services.Blog.GetBlog 990, 991

Ektron.Services.Blog.GetBlogData 992, 993, 994

Ektron.Services.Blog.GetBlogDataForPost 995

Ektron.Services.Blog.GetBlogPostData 995

Ektron.Services.Blog.GetBlogRoll 996

Ektron.Services.Blog.GetPost 997

Ektron.Services.Blog.GetPostbyID 997

Ektron.Services.Blog.GetUserBlog 998

Ektron.Services.Blog.UpdateBlogProperties 998

Ektron.Services.BusinessRules 999

Ektron.Services.BusinessRules.GetBusinessRules 1000

Ektron.Services.CommunityGroup 1001

Ektron.Services.CommunityGroup.AddCommunityGroup 1002

Ektron.Services.CommunityGroup.AddUserToCommunityGroup 1003

Ektron.Services.CommunityGroup.ApprovePendingGroupUser 1004

Ektron.Services.CommunityGroup.CancelJoinRequestForCommunityGroup 1004

Ektron.Services.CommunityGroup.DeleteCommunityGroupByID 1004

Ektron.Services.CommunityGroup.DeletePendingGroupUser 1005

Ektron.Services.CommunityGroup.GetAllCommunityGroups 1005

Ektron.Services.CommunityGroup.GetAllCommunityGroupsInDirectory 1006

Ektron.Services.CommunityGroup.GetAllUnassignedCommunityGroupUsers 1006

Ektron.Services.CommunityGroup.GetCommunityGroupBlogIds 1006

Ektron.Services.CommunityGroup.GetCommunityGroupByID 1007

Ektron.Services.CommunityGroup.GetCommunityGroupDiscussionBoardIds 1007

Ektron.Services.CommunityGroup.GetCommunityGroups 1008

Ektron.Services.CommunityGroup.GetCommunityGroupUserIDs 1008

Ektron.Services.CommunityGroup.GetCommunityGroupUsers 1009

Ektron.Services.CommunityGroup.GetFolderIdForCommunityGroup 1010

Ektron.Services.CommunityGroup.GetGroupMemberStatus 1010

Ektron.Services.CommunityGroup.GetInvitedCommunityGroups 1011

Ektron.Services.CommunityGroup.GetMostPopularCommunityGroups 1012

Ektron.Services.CommunityGroup.GetPendingCommunityGroups 1013

Ektron.Services.CommunityGroup.GetPendingCommunityGroupUsers 1013

Ektron.Services.CommunityGroup.InviteUsers 1014

Ektron.Services.CommunityGroup.IsCommunityGroupAdmin 1014

Ektron.Services.CommunityGroup.IsUserInCommunityGroup 1014

Ektron.Services.CommunityGroup.RemoveUserFromCommunityGroup 1015

Ektron.Services.CommunityGroup.UpdateCommunityGroup 1015

Ektron.Services.Content 1016

Ektron.Services.Content.AddContent 1017, 1018, 1019

Ektron.Services.Content.CheckOutContent 1019

Ektron.Services.Content.CopyContentToFolder 1020

Ektron.Services.Content.DeleteContentItem 1020

Ektron.Services.Content.GetAddViewLanguage 1021

Ektron.Services.Content.GetAllComments 1021

Ektron.Services.Content.GetAllTemplates 1021

Ektron.Services.Content.GetAllUnassignedItemApprovals 1022

Ektron.Services.Content.GetChildContent 1022, 1023

Ektron.Services.Content.GetContent 1024

Ektron.Services.Content.GetContentByHistory 1024

Ektron.Services.Content.GetContentForEditing 1025

Ektron.Services.Content.GetContentState 1025

Ektron.Services.Content.GetContentStatus 1025

Ektron.Services.Content.GetDomain 1026

Ektron.Services.Content.MoveContentToFolder 1026

Ektron.Services.Content.PublishContent 1026

Ektron.Services.Content.RenameContent 1027

Ektron.Services.Content.SaveContent 1027

Ektron.Services.Content.SubmitForDelete 1028

Ektron.Services.Content.UndoCheckout 1028

Ektron.Services.Content.UpdateContentMetaData 1029

Ektron.Services.ContentRating 1029

Ektron.Services.ContentRating.AddContentRating 1030

Ektron.Services.ContentRating.GetContentRatingResults 1030

Ektron.Services.ContentRating.PurgeContentRatings 1030

Ektron.Services.CustomFields 1031

Ektron.Services.CustomFields.AddNewEditableCustomFieldAssignments 1032

Ektron.Services.CustomFields.BoolToYesNo 1033

Ektron.Services.CustomFields.GetEditableCustomFieldAssignments 1033

Ektron.Services.CustomFields.GetEnhancedMetadataArea 1034

Ektron.Services.CustomFields.GetEnhancedMetadataScript 1034

Ektron.Services.CustomFields.GetFieldsByFolder 1035

Ektron.Services.CustomFields.GetSearchProperties 1035

Ektron.Services.CustomFields.ProcessCustomFields 1035

Ektron.Services.CustomFields.SearchAllAssets 1036

Ektron.Services.CustomFields.WriteFilteredMetadataForEdit 1036

Ektron.Services.CustomFields.WriteFilteredMetadataForView 1037

Ektron.Services.CustomFields.WriteLibrarySearchExtended 1037

Ektron.Services.CustomFields.WriteMetadataDefaultForEdit 1037

Ektron.Services.CustomFields.WriteMetadataTypeForView 1038

Ektron.Services.CustomFields.WritePublicSearch 1038

Ektron.Services.Favorites 1039

Ektron.Services.Favorites.AddContentFavorite 1040

Ektron.Services.Favorites.DeleteContentFavorite 1041

Ektron.Services.Favorites.GetContentFavorites 1041, 1042

Ektron.Services.Favorites.GetFavoritesDirectory 1043

Ektron.Services.Favorites.IsContentFavorite 1043

Ektron.Services.Favorites.MoveContentFavorite 1044

Ektron.Services.Flag 1044

Ektron.Services.Flag.AddFlagObject 1045

Ektron.Services.Flag.AssignFlagToContent 1045

Ektron.Services.Flag.GetAllFlagEntries 1046

Ektron.Services.Flag.GetContentFlagData 1046

Ektron.Services.Flag.GetFlagDataById 1047

Ektron.Services.Flag.GetFlaggedRank 1047

Ektron.Services.Flag.PurgeFlagEntries 1047

Ektron.Services.Flag.RemoveObjectObject 1048

Ektron.Services.Flag.UpdateFlagObject 1048

Ektron.Services.Folder 1048

Ektron.Services.Folder.AddFolder 1049

Ektron.Services.Folder.CreateFolder 1050

Ektron.Services.Folder.DeleteFolder 1050

Ektron.Services.Folder.GetChildFolders 1051

Ektron.Services.Folder.GetFolder 1052
Ektron.Services.Folder.GetFolderID 1053
Ektron.Services.Folder.GetPath 1054
Ektron.Services.Folder.RenameFolder 1054
Ektron.Services.Folder.UpdateFolder 1055
Ektron.Services.Font 1055
Ektron.Services.Font.AddFont 1056
Ektron.Services.Font.DeleteFont 1056
Ektron.Services.Font.GetAllFonts 1057
Ektron.Services.Font.GetFont 1057
Ektron.Services.Font.GetFontConfigList 1057
Ektron.Services.Font.UpdateFont 1058
Ektron.Services.Form 1058
Ektron.Services.Form.AssignFormTask 1059
Ektron.Services.Form.EmailFormFieldData 1059
Ektron.Services.Form.GetAllForms 1060, 1061
Ektron.Services.Form.GetForm 1061
Ektron.Services.Form.GetFormDataHistogram 1062
Ektron.Services.Form.GetFormFieldData 1062, 1063
Ektron.Services.Form.GetFormFieldList 1063
Ektron.Services.Form.GetFormFieldQuestions 1064
Ektron.Services.Form.GetFormTitle 1064
Ektron.Services.Form.SubmitFormFieldData 1065
Ektron.Services.Form.UpdateFormFieldData 1065
Ektron.Services.Friends 1066
Ektron.Services.Friends.AcceptPendingFriend 1067
Ektron.Services.Friends.AddFriend 1067, 1068
Ektron.Services.Friends.AddPendingFriend 1068
Ektron.Services.Friends.AddSelectedFriend 1069
Ektron.Services.Friends.DeleteFriend 1069
Ektron.Services.Friends.DeletePendingFriendRequest 1070
Ektron.Services.Friends.DeleteSelectedFriend 1071
Ektron.Services.Friends.GetFriends 1072, 1073
Ektron.Services.Friends.GetFriendsDirectory 1073
Ektron.Services.Friends.GetFriendStatus 1074
Ektron.Services.Friends.GetInvitedUsers 1074
Ektron.Services.Friends.GetPendingFriends 1075
Ektron.Services.Friends.GetSelectedFriends 1075
Ektron.Services.Friends.GetSentFriendInvitations 1076
Ektron.Services.Friends.InviteUsers 1076
Ektron.Services.Friends.IsFriend 1077
Ektron.Services.Friends.IsSelectedFriend 1077
Ektron.Services.Friends.MoveFriend 1077
Ektron.Services.Library 1078
Ektron.Services.Library.AddLoadBalanceItem 1079
Ektron.Services.Library.DeleteLibraryItem 1079
Ektron.Services.Library.DeleteLibraryItemById 1079
Ektron.Services.Library.DeleteLoadBalanceItem 1080
Ektron.Services.Library.GetAllChildLibItems 1080
Ektron.Services.Library.GetAllLoadBalancePaths 1081, 1082
Ektron.Services.Library.GetLibraryItem 1083
Ektron.Services.Library.GetLibrarySettings 1083
Ektron.Services.Library.GetLibraryTypes 1084
Ektron.Services.Library.GetLoadBalancePath 1084
Ektron.Services.Library.InternalLibrarySearch 1085
Ektron.Services.Library.UpdateLibrary 1085
Ektron.Services.Library.UpdateLibrarySettings 1085
Ektron.Services.Library.UpdateLoadBalanceSettings 1086
Ektron.Services.Library.UpdateQlinkTemplates 1086
Ektron.Services.MessageBoard 1086
Ektron.Services.MessageBoard.AddMessageBoardEntry 1087
Ektron.Services.MessageBoard.ApproveMessage 1088, 1089
Ektron.Services.MessageBoard.DeleteMessageBoardEntries 1090, 1091
Ektron.Services.MessageBoard.DeleteMessageBoardEntriesByUser 1092, 1093
Ektron.Services.MessageBoard.DeleteMessageBoardEntry 1094, 1095
Ektron.Services.MessageBoard.GetAllMessageBoardEntries 1097, 1098
Ektron.Services.MessageBoard.GetAllMessageBoardEntriesByUser 1099, 1100
Ektron.Services.MessageBoard.GetMessageBoardEntry 1101, 1102
Ektron.Services.MessageBoard.IsUserMessageBoardAdmin 1103
Ektron.Services.MessageBoard.UpdateMessageText 1104, 1105
Ektron.Services.Metadata 1106
Ektron.Services.Metadata.AddMetaDataType 1106

Ektron.Services.Metadata.DeleteMetadataType 1107
Ektron.Services.Metadata.GetContentMetadataList 1107, 1108
Ektron.Services.Metadata.GetMetadataType 1108
Ektron.Services.Metadata.GetMetaDataTypes 1109
Ektron.Services.Metadata.UpdateMetadataType 1109
Ektron.Services.Permissions 1110
Ektron.Services.Permissions.AddItemPermission 1110
Ektron.Services.Permissions.DeleteItemPermission 1111
Ektron.Services.Permissions.DisableContentInheritance 1111
Ektron.Services.Permissions.DisableFolderInheritance 1112
Ektron.Services.Permissions.EnableContentInheritance 1112
Ektron.Services.Permissions.EnableFolderInheritance 1113
Ektron.Services.Permissions.GetUserPermissions 1113, 1114
Ektron.Services.Permissions.LoadPermissions 1114
Ektron.Services.Permissions.UpdateItemPermission 1115
Ektron.Services.SearchManager 1115
Ektron.Services.SearchManager.ExecuteSearch 1116, 1117
Ektron.Services.SearchManager.Search 1118, 1119
Ektron.Services.SearchManager.SearchForCategories 1119
Ektron.Services.SearchManager.SearchTable 1119
Ektron.Services.Site 1120
Ektron.Services.Site.BreakTaskPermissionInherit 1121
Ektron.Services.Site.GetAllActiveLanguages 1121
Ektron.Services.Site.GetAllLanguages 1122
Ektron.Services.Site.GetClientDayNames 1123
Ektron.Services.Site.GetClientMonthNames 1123
Ektron.Services.Site.GetCookieValue 1123
Ektron.Services.Site.GetLanguage 1124
Ektron.Services.Site.GetLanguages 1124
Ektron.Services.Site.GetSiteVariables 1125
Ektron.Services.Site.GetTaskPermission 1125
Ektron.Services.Site.IsTaskPermissionInherited 1126
Ektron.Services.Site.SetCookieValue 1126
Ektron.Services.Site.SetTaskPermission 1126
Ektron.Services.Site.UpdateSiteEnabledLanguages 1127
Ektron.Services.SiteMap 1127
Ektron.Services.SiteMap.GetFolderBreadcrumbPath 1128
Ektron.Services.SiteMap.GetWebsiteMap 1128
Ektron.Services.SiteMap.RenderFolderBreadcrumbHtml 1129
Ektron.Services.SiteMap.RenderWebsiteMap 1129
Ektron.Services.Tags 1130
Ektron.Services.Tags.AddTag 1131
Ektron.Services.Tags.AddTagToCommunityGroup 1132, 1133
Ektron.Services.Tags.AddTagToContent 1134, 1135
Ektron.Services.Tags.AddTagToObject 1136
Ektron.Services.Tags.AddTagToUser 1137, 1138
Ektron.Services.Tags.DeleteTagById 1139
Ektron.Services.Tags.DeleteTagOnObject 1139
Ektron.Services.Tags.DeleteTagsByUser 1140
Ektron.Services.Tags.DeleteTagsOnCommunityGroup 1140
Ektron.Services.Tags.DeleteTagsOnContent 1141
Ektron.Services.Tags.DeleteTagsOnObject 1141
Ektron.Services.Tags.DeleteTagsOnUser 1142
Ektron.Services.Tags.EditTag 1142
Ektron.Services.Tags.GetAllTagRelatedContent 1142
Ektron.Services.Tags.GetAllTagRelatedUsers 1143
Ektron.Services.Tags.GetAllTags 1144
Ektron.Services.Tags.GetDefaultTags 1145, 1146
Ektron.Services.Tags.GetTagByID 1146
Ektron.Services.Tags.GetTagByName 1146
Ektron.Services.Tags.GetTagCloud 1147
Ektron.Services.Tags.GetTagsForContent 1147
Ektron.Services.Tags.GetTagsForObject 1148, 1149
Ektron.Services.Tags.GetTagsForUser 1150
Ektron.Services.Tags.RemoveTagAsDefault 1150
Ektron.Services.Tags.SaveTagAsDefault 1151
Ektron.Services.Task 956
Ektron.Services.Task namespace 956
Classes 956
Ektron.Services.Task.Task 957
Ektron.Services.Task.Task.AddTask 957
Ektron.Services.Task.Task.AssignTaskToUser 958, 959
Ektron.Services.Task.Task.AssignTaskToUsergroup 959, 960
Ektron.Services.Task.Task.DeleteTask 960, 961
Ektron.Services.Task.Task.DeleteTasks 961, 962
Ektron.Services.Task.Task.GetTask 962
Ektron.Services.Task.Task.GetTasks 963
Ektron.Services.Task.Task.GetTasksAssignedByUser 963, 964

Ektron.Services.Task.Task.GetTasksAssignedToUser 965
Ektron.Services.Task.Task.GetTasksAssignedToUsergroup 966
Ektron.Services.Task.Task.GetTaskState 967
Ektron.Services.Task.Task.SetStateForTask 967
Ektron.Services.Task.Task.UpdateTask 967
Ektron.Services.Task.TaskCategory 968
Ektron.Services.Task.TaskCategory.AddTaskCategory 969
Ektron.Services.Task.TaskCategory.DeleteTaskCategory 969
Ektron.Services.Task.TaskCategory.GetAllCategories 969
Ektron.Services.Task.TaskCategory.GetTaskCategory 970
Ektron.Services.Task.TaskCategory.UpdateTaskCategory 970
Ektron.Services.Task.TaskCategoryType 971
Ektron.Services.Task.TaskCategoryType.AddTaskType 971
Ektron.Services.Task.TaskCategoryType.DeleteTaskType 972
Ektron.Services.Task.TaskCategoryType.GetTaskType 972
Ektron.Services.Task.TaskCategoryType.UpdateTaskType 972
Ektron.Services.Taxonomy 1151
Ektron.Services.Taxonomy.AddTaxonomyItem 1153
Ektron.Services.Taxonomy.AddTaxonomySyncFolder 1153
Ektron.Services.Taxonomy.CreateTaxonomy 1154
Ektron.Services.Taxonomy.DeleteTaxonomy 1154
Ektron.Services.Taxonomy.GetTaxonomyIdByPath 1154
Ektron.Services.Taxonomy.LoadTaxonomy 1155
Ektron.Services.Taxonomy.ReadAllAssignedCategory 1155
Ektron.Services.Taxonomy.ReadTaxonomy 1156
Ektron.Services.Taxonomy.RemoveTaxonomyFolder 1156
Ektron.Services.Taxonomy.RemoveTaxonomyItem 1156
Ektron.Services.Taxonomy.ReOrderTaxonomyItems 1157
Ektron.Services.Taxonomy.UpdateTaxonomy 1157
Ektron.Services.ThreadedDiscussion 1157
Ektron.Services.ThreadedDiscussion.AddBoard 1158
Ektron.Services.ThreadedDiscussion.AddForum 1159
Ektron.Services.ThreadedDiscussion.AddTopic 1160
Ektron.Services.ThreadedDiscussion.DeleteBoard 1161
Ektron.Services.ThreadedDiscussion.DeleteForum 1161
Ektron.Services.ThreadedDiscussion.DeleteTopic 1162
Ektron.Services.ThreadedDiscussion.GetBoardCategories 1162
Ektron.Services.ThreadedDiscussion.GetBoardForums 1162
Ektron.Services.ThreadedDiscussion.GetDiscussionBoard 1163, 1164
Ektron.Services.ThreadedDiscussion.GetDiscussionBoardString 1164
Ektron.Services.ThreadedDiscussion.GetForum 1165
Ektron.Services.ThreadedDiscussion.GetRepliesForTopics 1165
Ektron.Services.ThreadedDiscussion.GetTopic 1166
Ektron.Services.ThreadedDiscussion.SearchReplies 1166
Ektron.Services.User 1167
Ektron.Services.User.ActivateUserAccount 1169
Ektron.Services.User.AddMembershipUser 1170
Ektron.Services.User.AddNewUser 1170
Ektron.Services.User.AddRolePermission 1171
Ektron.Services.User.AddUser 1171
Ektron.Services.User.AddUserGroup 1172
Ektron.Services.User.AddUserToGroup 1172
Ektron.Services.User.AutoLogInUser 1172
Ektron.Services.User.CreateBlankUserObject 1173, 1174
Ektron.Services.User.DeleteUserByID 1174
Ektron.Services.User.DeleteUserByIds 1174
Ektron.Services.User.DeleteUserFromGroup 1175
Ektron.Services.User.DeleteUserGroup 1175
Ektron.Services.User.EditUserCustomProperties 1176
Ektron.Services.User.GetActiveUser 1176
Ektron.Services.User.GetADMapping 1176
Ektron.Services.User.GetAllCmsUsers 1177
Ektron.Services.User.GetAllCustomProperties 1178
Ektron.Services.User.GetAllUserByCustomProperty 1178
Ektron.Services.User.GetAllUserGroups 1179, 1180
Ektron.Services.User.GetAllUsers 1181
Ektron.Services.User.GetGroupsUserIsIn 1181
Ektron.Services.User.GetMembershipUser 1182
Ektron.Services.User.GetUser 1182
Ektron.Services.User.GetUserbyUsername 1183
Ektron.Services.User.GetUserCustomProperties 1183
Ektron.Services.User.GetUserEmailInfo 1184
Ektron.Services.User.GetUserForFolderAdmin 1184
Ektron.Services.User.GetUserGroup 1185
Ektron.Services.User.GetUserGroupForFolderAdmin 1185
Ektron.Services.User.GetUserPermissions 1185
Ektron.Services.User.GetUserPreference 1186

Ektron.Services.User.GetUsers 1187, 1188
Ektron.Services.User.GetUsersForTask 1188
Ektron.Services.User.GetUsersNotInGroup 1189
Ektron.Services.User.IsAdmin 1189
Ektron.Services.User.IsARoleMember 1191, 1192
Ektron.Services.User.IsARoleMemberForContent 1193
Ektron.Services.User.IsARoleMemberForFolder 1193
Ektron.Services.User.LogInUser 1194
Ektron.Services.User.ResetMembershipUserPassword 1195
Ektron.Services.User.UnsubscribeUser 1195
Ektron.Services.User.UpdateCustomPropertiesItemOrder 1196
Ektron.Services.User.UpdateMapCoordinate 1196
Ektron.Services.User.UpdateUser 1197, 1198
Ektron.Services.User.UpDateUserGroup 1198
Ektron.Services.User.UpdateUserPreferences 1198
Ektron.Services.User.UserObject 1199, 1200
EktronCatalogPart class 202
about EktronCatalogPart class 202
BlogID 203
BoardID 203
CalendarDynamicParameter 204
CollectionDynamicParameter 204
ContentBlockDynamicParameter 204
DefaultCalendarID 205
DefaultCollectionID 205
DefaultContentID 205
DefaultFormID 206
EktronCatalogPart properties 202
EnableBlogControl 206
EnableCalendarControl 206
EnableCollectionControl 206
EnableContentBlockControl 207
EnableDiscussionBoardsControl 207
EnableFormBlockControl 207
EnableListSummaryControl 208
EnableThreadedDiscussionsControl 208
FolderID 208
FormBlockDynamicParameter 208
Title 209
EktronCommunityCatalogPart class 209
about EktronCommunityCatalogPart class 209
EktronCommunityCatalogPart properties 209
EnableBlogs 210
EnableDiscussionBoards 210
EnableForums 210
Title 210
EktronWebPartZone class 211
about EktronWebPartZone class 211
CmsSitePath 211
EktronWebPartZone properties 211
EkWebControl class 212
about EkWebControl class 212
Authenticated 218
CmsSitePath 218
EkWebControl 212
EkWebControl methods 212
EkWebControl properties 217
Error 218
Fill 213
GetEnumDescription 213
GetFilenameFromPath 213
GetMessage 214
GetMessage method 214
Hide 219
IsLoggedIn 219
Language 219
LanguageID 220
LocalizeString 214
loggedInUserID 220
loggedInUserName 220
LogLevel 220
RedirectFromLoginPage 215
RedirectToLoginPage 215
RegisterAllDmsMenuCssAndScripts 215
RegisterDmsMenuCSS 215
RegisterDmsMenuScript 216
RegisterJQueryScript 216
RegisterMembershipTabScript 216

RegisterThickBoxCSS 216
RegisterThickBoxScript 216
SearchHeader 216
SuppressWrapperTags 220
Text 221
ToString 217
Wrap 217
WrapTag 221
EkWebPart class 222
about EkWebPart class 222
CatalogIconImageUrl 223
CreateEditorParts 222
Description 223
EkWebPart methods 222
EkWebPart properties 222
Subtitle 223
Title 223
TitleIconImageUrl 223
TitleUrl 224
WebBrowsableObject 224
EkXsltBase class 224
about EkXsltBase class 224
CustomXml 225
DisplayXslt 225
EkXsltBase 224
EkXsltBase properties 225
EnableErrorLogging 225
XmlDoc 226
XsltDoc 226
EntryAttributeData class 1262
about EntryAttributeData class 1262
AttributeTypeId 1264
CurrentValue 1264
EntryAttributeData 1263
EntryAttributeData constructor 1262
EntryAttributeData properties 1264
VersionId 1264
EntryData class 1264
about EntryData class 1264
Attributes 1267
Comment 1267
ContentStatus 1267
CurrencyId 1267
DateModified 1268
Dimensions 1268
DisableInventoryManagement 1268
EndDate 1268
EndDateAction 1268
EntryData 1265
EntryData constructor 1265
EntryData properties 1266
EntryType 1269
FolderId 1269
GoLive 1269
Html 1269
Id 1269
Image 1270
ImageThumbnail 1270
IsArchived 1270
IsBuyable 1270
IsMarkedForDeletion 1271
IsPublished 1271
IsTangible 1271
LanguageId 1271
LastEditorFirstName 1271
LastEditorId 1272
LastEditorLastName 1272
ListPrice 1272
Media 1272
Metadata 1273
NumberSold 1273
Pricing 1273
ProductType 1273
QuantityMultiple 1273
Quicklink 1274
SalePrice 1274
Sku 1274
Status 1274

StatusLanguage 1275
Summary 1275
TaxClassId 1275
TemplateId 1275
Title 1275
Url 1276
VersionId 1276
Weight 1276
EntryGroupData class 1276
about EntryGroupData class 1276
Description 1278
DisplayOrder 1278
EntryGroupData 1277
EntryGroupData constructor 1277
EntryGroupData properties 1277
EntryId 1278
GroupType 1278
Id 1278
Name 1278
EntryGroupItemData class 1279
about EntryGroupItemData class 1279
DisplayOrder 1280
EntryGroupItemData 1279
EntryGroupItemData constructor 1279
EntryGroupItemData properties 1280
EntryId 1280
GroupId 1280
Id 1281
ExchangeRateApi class 605
about ExchangeRateApi class 605
Add 606
ConvertPrice 606
Delete 606
ExchangeRateApi methods 605
GetCurrentExchangeRate 606
GetCurrentList 607
GetList 607
Update 607
ExchangeRateData class 1281
about ExchangeRateData class 1281
BaseCurrencyId 1282
ConvertPrice 1281
EffectiveDate 1282
ExchangeCurrencyId 1283
ExchangeRateData 1281
ExchangeRateData methods 1281
ExchangeRateData properties 1282
Rate 1283
Validate 1282
ExplorerDragDrop class 226
about ExplorerDragDrop class 226
AssetID 227
ContentLanguage 227
ExplorerDragDrop properties 227
FolderID 227
IsImage 228
OverrideExtension 228
TaxonomyId 228
ExtensionAuthor class 533
about ExtensionAuthor class 533
Author 534
ExtensionAuthor 533
ExtensionAuthor methods 533
ExtensionAuthor properties 534
ToString 534
ExtensionDescription class 534
about ExtensionDescription class 534
Desc 535
ExtensionDescription 535
ExtensionDescription methods 535
ExtensionDescription properties 535
ToString 535
ExtensionEvent class 535
about ExtensionEvent class 535
Content 552
ExtensionEvent methods 536
ExtensionEvent properties 552
Folder 553

Form 553
GetActiveDirectoryUserProxy 538
GetAssetProxy 538
GetBlogProxy 538
GetBusinessRulesProxy 539
GetCalendarEventTypeProxy 539
GetCalendarProxy 539
GetCommunityGroupProxy 539
GetContentProxy 539
GetContentRatingProxy 540
GetCustomFieldsProxy 540
GetFavoritesProxy 540
GetFlagProxy 540
GetFolderProxy 541
GetFontProxy 541
GetFormProxy 541
GetFriendsProxy 541
GetLibraryProxy 542
GetMessageBoardProxy 542
GetMetadataProxy 542
GetObjectStateAsXml 542
GetPermissionsProxy 542
GetSearchManagerProxy 543
GetSiteMapProxy 543
GetSiteProxy 543
GetTagsProxy 543
GetTaskCategoryProxy 544
GetTaskCategoryTypeProxy 544
GetTaskProxy 544
GetTaxonomyProxy 544
GetThreadedDiscussionProxy 545
GetUserProxy 545
OnAfterActionRequestSent 545
OnAfterContentAdd 545
OnAfterContentDelete 546
OnAfterContentEdit 546
OnAfterContentFlagged 546
OnAfterFolderAdd 546
OnAfterFolderDelete 546
OnAfterFolderEdit 547
OnAfterFormSubmit 547
OnAfterPublish 547
OnAfterReviewRejected 547
OnAfterUserAdd 548
OnAfterUserDelete 548
OnAfterUserEdit 548
OnAfterUserLogin 548
OnBeforeActionRequestSent 549
OnBeforeContentAdd 549
OnBeforeContentDelete 549
OnBeforeContentEdit 549
OnBeforeContentFlagged 550
OnBeforeFolderAdd 550
OnBeforeFolderDelete 550
OnBeforeFolderEdit 550
OnBeforeFormSubmit 550
OnBeforePublish 551
OnBeforeReviewRejected 551
OnBeforeUserAdd 551
OnBeforeUserDelete 551
OnBeforeUserEdit 552
OnBeforeUserLogin 552
Site 553
User 553

## F

Favorites class 229, 731, 1039
about Favorites class 229, 731, 1039
AddContentFavorite 732, 733, 1040
AddContentFavorite method 732, 1039
AddFavoriteLink 733
CacheInterval 231
DefaultUserID 231
DeleteContentFavorite 733, 1041
DeleteFavoriteLink 734
DisplayMode 231
DisplayXslt 232
DynamicParameter 232

Favorites methods 229, 732, 1039
Favorites properties 230
FavoritesTitle 232
FavoritesUriTitle 233
Fill 229
GetAllFavoriteLink 734, 735
GetAllFavoriteLink method 734
GetContentFavorites 736, 737, 1041, 1042
GetContentFavorites method 735, 1041
GetFavoriteId 737
GetFavoriteLink 738
GetFavoritesDirectory 738, 1043
IsContentFavorite 738, 1043
Link 233
LinkTarget 233
MarkupLanguage 234
MaxResults 234
MoveContentFavorite 739, 1044
MoveFavoriteLink 739
Stylesheet 234
ToString 229
UpdateFavoriteLink 739
Wrap 229
XmlDoc 235
Flag class 1044
about Flag class 1044
AddFlagObject 1045
AssignFlagToContent 1045
Flag methods 1045
GetAllFlagEntries 1046
GetContentFlagData 1046
GetFlagDataById 1047
GetFlaggedRank 1047
PurgeFlagEntries 1047
RemoveObjectObject 1048
UpdateFlagObject 1048
Flagging class 739
about Flagging class 739
AddFlagObject 740
AssignFlagToContent 741
Flagging methods 740
GetAllFlagEntries 741, 742
GetAllFlagEntries method 741
GetContentFlagData 743
GetContentFlagData method 742
GetDefaultFolderFlagDef 743
GetFlagDataById 744
GetFlaggedRank 744
PurgeFlagEntries 744
RemoveObjectFlag 744
UpdateFlagObject 745
FlagInfo class 553
about FlagInfo class 553
ContentHtml 554
ContentID 554
ContentLanguage 555
ContentTitle 555
EntryId 555
ErrorMessage 555
FlagComment 555
FlagDate 555
FlagId 556
FlagInfo properties 554
FlagName 556
UserAvatar 556
UserDisplayName 556
UserFirstName 556
UserID 556
UserLastName 557
UserName 557
VisitorID 557
FlexMenu class 235
about FlexMenu class 235
AutoCollapseBranches 236
CacheInterval 237
DisplayXslt 237
EnableAjax 237
EnableMouseOverPopUp 238

EnableSmartOpen 238
EnableSubmenuLinkPathMatching 238
EnableSubmenuTemplateMatching 239
Fill 235
FlexMenu methods 235
FlexMenu properties 236
MasterControlId 239
MenuDepth 239
SetDirty 235
StartCollapsed 240
StartLevel 240
Stylesheet 240
UseAliasRealId 241
XmlDoc 241
XsltDoc 241
Folder class 652, 1048
about Folder class 652, 1048
AddCommunityFolder 653
AddFolder 653, 1049
CreateFolder 654, 1050
DeleteFolder 1050
DeleteFolderById 655
Folder methods 652, 1049
GetChildFolders 656, 658, 1051
GetChildFolders method 656, 1050
GetFolder 659, 660, 1052
GetFolder method 659, 1051
GetFolderDataWithPermission 661
GetFolderId 661
GetFolderID 1053
GetFolderId method 661
GetFolderID method 1053
GetFolderInfo 662
GetPath 662, 1054
IsFolderHidden 662
RenameFolder 663, 1054
UpdateFolder 663, 664, 1055
UpdateFolder method 663, 1054
UpdateFolderApprovalMethod 664
FolderBreadcrumb class 241
about FolderBreadcrumb class 241
DefaultContentID 243
DefaultFolderID 243
DisplayStyle 244
DisplayXslt 244
DynamicParameter 244
Fill 242
FolderBreadcrumb methods 242
FolderBreadcrumb properties 242
LinkTarget 245
Mode 245
separator 245
XmlDoc 245
FolderInfo class 557
about FolderInfo class 557
Description 558
ErrorMessage 558
FolderInfo properties 557
ID 558
Name 558
ParentID 558
ReplicationState 558
StyleSheet 559
Type 559
Font class 1055
about Font class 1055
AddFont 1056
DeleteFont 1056
Font methods 1056
GetAllFonts 1057
GetFont 1057
GetFontConfigList 1057
UpdateFont 1058
Form class 836, 1058
about Form class 836, 1058
AssignFormTask 837, 1059
EmailFormFieldData 837, 1059
Form methods 836, 1058

GetAllForms 838, 1060, 1061
GetAllForms method 838, 1060
GetForm 839, 1061
GetFormDataHistogram 1062
GetFormFieldData 839, 840, 1062, 1063
GetFormFieldData method 839, 1062
GetFormFieldList 840, 1063
GetFormFieldQuestions 840, 1064
GetFormTitle 841, 1064
SubmitFormFieldData 841, 1065
UpdateFormFieldData 841, 1065
FormBlock class 246
about FormBlock class 246
AddValidation 249
CacheInterval 250
CopyTo 247
Count 250
Current 250
DefaultFormID 250
DynamicParameter 251
EkItem 251
Fields 251
Fill 247
FormBlock methods 246
FormBlock properties 249
FormName 252
GetEnumerator 247
GetView 247
GetViewNames 247
IncludeTags 252
IsSynchronized 252
MoveNext 248
Reset 248
StartingFolderID 253
SyncRoot 253
this 248
ToString 248
Wrap 248
FormData class 1394
about FormData class 1394
Autofill 1395
Description 1396
FormData properties 1395
MailCc 1396
MailFrom 1396
MailPreamble 1396
MailSubject 1397
MailTo 1397
PollRevision 1397
SendXmlPacket 1397
StoreDataTo 1398
Validation 1398
FormField class 559
about FormField class 559
FieldName 560
FieldValue 560
FormField 559
FormField properties 560
FormFieldDataItem class 1399
about FormFieldDataItem class 1399
DataValue 1399
FieldName 1399
FormFieldDataItem properties 1399
FormFieldDatalist class 1400
about FormFieldDatalist class 1400
CaptionXPath 1400
DatalistItems 1401
DataNamespaces 1401
DataSelect 1401
DataSrc 1401
FormFieldDatalist properties 1400
Language 1402
Name 1402
ValueXPath 1402
FormFieldDatalistItem class 1402
about FormFieldDatalistItem class 1402
DataValue 1403
DisplayValue 1403

FormFieldDatalistItem properties 1403
IsDefault 1404
IsEnabled 1404
FormFieldDefinition class 1404
about FormFieldDefinition class 1404
Basetype 1405
ContentType 1405
DatalistName 1406
Datatype 1406
DisplayName 1406
FieldName 1407
FormFieldDefinition properties 1405
Language 1407
XPath 1407
FormFieldItem class 253
about FormFieldItem class 253
Basetype 254
ContentType 254
DatalistItems 254
Datatype 255
DisplayName 255
FieldName 255
FormFieldItem properties 254
Language 256
Value 256
FormFieldList class 1407
about FormFieldList class 1407
Datalists 1408
Fields 1408
FormFieldList properties 1408
FormInfo class 560
about FormInfo class 560
ErrorMessage 561
FormDescription 561
FormFields 561
FormInfo methods 560
FormInfo properties 561
FormTitle 561
this 560
FormRequestData class 1409
about FormRequestData class 1409
AutoFill 1410
FieldList 1410
FormRequestData properties 1409
Id 1410
IsPoll 1410
Language 1411
PostBackData 1411
PreFillFielddata 1411
TagRequired 1412
FormSubmittedData class 1412
about FormSubmittedData class 1412
DataItems 1413
DateSubmitted 1413
FormDataID 1413
FormSubmittedData properties 1412
UserId 1413
UserInfo 1414
Forum class 256
about Forum class 256
AllowAnonymousPost 260
BoardID 260
CacheInterval 260
CustomOrderBy 261
DynamicForumParameter 261
DynamicGroupParameter 261
DynamicParameter 262
DynamicThreadParameter 262
EditorCSS 262
EditorToolbar 263
EnableForumQuickSelect 263
EnableThreadRating 263
ExcludeList 264
Fill 257
FilterXslt 264
Forum methods 257
Forum properties 258
GetCallbackResult 257

HideUserProfile 265
HighlightedUserRole 265
JavascriptEditorHeight 265
JavascriptEditorHTMLMode 266
JavascriptEditorToolbar 266
JavascriptEditorWidth 267
OrderBy 267
OrderDirection 268
ProfileLink 268
ProfileLinkTarget 268
RaiseCallbackEvent 257
ShowCategories 268
TaxCategories 269
TaxonomyCols 269
TaxonomyMarkupLanguage 269
Theme 269
ToString 257
Wrap 258
Friend enumeration member 528
Friends class 270, 745, 1066
about Friends class 270, 745, 1066
AcceptFriendInvite 746
AcceptPendingFriend 746, 1067
AddFriend 747, 1067, 1068
AddFriend method 746, 1067
AddPendingFriend 747, 1068
AddSelectedFriend 748, 1069
AvatarHeight 272
AvatarWidth 272
CacheInterval 273
DefaultUserID 273
DeleteFriend 748, 1069
DeletePendingFriendRequest 749, 1070
DeletePendingFriendRequest method 748, 1069
DeleteSelectedFriend 749, 1071
DisplayMode 273
DisplayXslt 274
DynamicParameter 274
Fill 270
FriendLabel 275
Friends methods 270, 745, 1066
Friends properties 271
FriendsLabel 275
GetFriends 750, 751, 1072, 1073
GetFriends method 749, 1071
GetFriendsDirectory 751, 1073
GetFriendStatus 752, 1074
GetInvitedUsers 752, 1074
GetPendingFriends 752, 1075
GetSelectedFriends 753, 1075
GetSentFriendInvitations 753, 1076
InviteUsers 753, 1076
IsFriend 754, 1077
IsSelectedFriend 754, 1077
Link 275
LinkTarget 275
MarkupLanguage 276
MaxResults 276
MoveFriend 754, 1077
ShowSortingOptions 276
Stylesheet 277
ToString 271
Wrap 271
XmlDoc 277

## H

How Do I... 2
HtmlEditDisplay class 277
about HtmlEditDisplay class 277
BaseJavaScript 278
BeginJavaScript 278
EndJavaScript 279
HtmlEditDisplay methods 277
MakeFieldNameFromId 279
RenderCMSFields 279
SelectedJavaScript 280
HtmlEditor class 280
about HtmlEditor class 280

Archived 284
Authenticated 284
CmsSitePath 285
ContentAction 285
ContentID 286
ContentTitle 286
ContentType 287
EditorReadyCall 287
EnableUpload 288
ExecuteContentAction 281
HtmlEditor 281
HtmlEditor methods 281
HtmlEditor properties 282
IncludeContentInfo 288
IsLoggedIn 289
Language 289
loggedInUserID 290
loggedInUserName 290
Path 290
PreferredType 291
SafeEncode 291
ShowCMSToolbar 292
ShowTitle 292
StyleSheet 293
Text 293
ToolbarLevel 294
ToolbarResetCall 294
ToString 282
TranslationFile 295
WorkareaMode 282

ImageControl class 295
about ImageControl class 295
DefaultImageID 297
DynamicParameter 297
Fill 296
FolderID 297
ImageControl methods 296
ImageControl properties 296
TaxonomyID 298
Title 298
Wrap 296
ImageMediaData class 1283
about ImageMediaData class 1283
ImageMediaData properties 1283
Thumbnails 1284
IndexSearch class 298
about IndexSearch class 298
AddParm 300
AddParm method 300
ButtonImgSrc 305
ButtonText 305
ContentParameter 306
CopyTo 301
Count 306
Current 306
CustomOrderBy 306
EkItems 307
EmptyResultMsg 307
EncodeContentHtml 307
Fill 301
FolderId 307
FormName 308
GetEnumerator 301
GetSearchParmDataType 301
GetSearchParmSearchType 302
GetXPathValues 302
IndexSearch 299
IndexSearch methods 300
IndexSearch properties 304
IndexSearch structures 299
IsSynchronized 308
Labelafter 308
Labelbefore 308
LabelBetween 309
LabelContains 309
LabeldateBetween 309

LabelEqual 310
LabelExactPhrase 310
LabelGreaterThan 310
Labelinputdate 311
Labelinputnumber 311
LabelLessThan 311
Labelon 312
LabelSelect 312
LinkTarget 312
MaxResults 313
MoveNext 302
OrderBy 313
OrderByDirection 313
Recursive 314
Reset 302
Search 303
Search method 303
SearchParmXML 314
ShowSearchBoxAlways 314
SyncRoot 315
this 303
Weighted 315
Wrap 303
XmlConfigId 315
XmlDoc 315
InventoryApi class 608
about InventoryApi class 608
InventoryApi methods 608
SaveInventory 608
InventoryData class 1284
about InventoryData class 1284
EntryId 1284
InventoryData properties 1284
ReorderLevel 1285
UnitsInStock 1285
UnitsOnOrder 1285
Invite class 316
about Invite class 316
DisplayXslt 317
Fill 316
FriendMessageId 317
Invite methods 316
Invite properties 317
InviteMessageId 318
OptionalMessageText 318
Stylesheet 318
ToString 316
Wrap 316
XmlDoc 318


## J


JoinGroup enumeration member 528


## K


KitConfigData class 1285
about KitConfigData class 1285
BasketItemId 1286
GetGroupIndex 1286
Groups 1286
Id 1287
KitConfigData 1285
KitConfigData methods 1286
KitConfigData properties 1286
PriceModifier 1287
KitData class 1287
about KitData class 1287
KitData 1287
KitData properties 1288
OptionGroups 1288


## L


LanguageAPI class 319
about LanguageAPI class 319
CurrentLanguage 321
CurrentLanguageID 321
DefaultLanguage 321
DefaultLanguageID 322
Fill 319

GetLanguage 320
GetLanguageID 320
IsValid 320
IsValid method 320
LangaugeIdList 322
LangaugeTitleList 322
LanguageAPI 319
LanguageAPI methods 319
LanguageAPI properties 320
MultiLanguageEnabled 322
OverrideDefault 322
SiteLanguage 323
SiteLanguageID 323
LanguageSelect class 323
about LanguageSelect class 323
Fill 324
LabelName 324
LanguageSelect methods 324
LanguageSelect properties 324
Text 325
Wrap 324
Library class 664, 1078
about Library class 664, 1078
AddLibraryItem 665
AddLoadBalanceItem 665, 1079
DeleteLibraryItem 666, 1079
DeleteLibraryItemById 666, 1079
DeleteLoadBalanceItem 666, 1080
GetAllChildLibItems 666, 1080
GetAllLoadBalancePaths 669, 1081, 1082
GetAllLoadBalancePaths method 668, 1081
GetLibraryItem 670, 1083
GetLibraryItem method 669, 1082
GetLibrarySettings 670, 1083
GetLibraryTypes 671, 1084
GetLoadBalancePath 671, 1084
InternalLibrarySearch 671, 1085
Library methods 664, 1078
UpdateLibrary 672, 1085
UpdateLibrarySettings 672, 1085
UpdateLoadBalanceSettings 672, 1086
UpdateQlinkTemplates 672, 1086
List of Developer Resources 4
ListSummary class 325
about ListSummary class 325
AddText 329
CacheInterval 330
ContentParameter 330
ContentType 331
CopyTo 326
Count 331
Current 331
Direction 332
EkItems 332
EnablePaging 332
Fill 326
FolderDescription 333
FolderID 333
FolderName 333
FolderPath 333
GetEnumerator 326
GetHtml 334
GetRssFeed 326
GetView 327
GetViewNames 327
IncludeIcons 334
IsSynchronized 334
LinkTarget 335
ListSummary methods 325
ListSummary properties 328
MarkupLanguage 335
MaxResults 335
MoveNext 327
OrderByDirection 335
OrderKey 336
Random 336
Recursive 336
Reset 327

SuppressWrapperTags 337
SyncRoot 337
this 328
ToString 328
Wrap 328
XmlDoc 337
Localization class 673
about Localization class 673
GetAddViewLanguage 674
GetAddViewLanguageForAllContent 674
GetAddViewLanguageForMenus 674
GetContentItems 675
GetFile 675
GetFiles 675
GetFilesInZipFile 676
GetFlagUrl 676, 677
GetFlagUrl method 676
GetInterchangeFile 678
GetInterchangeFiles 678
GetInterchangeFilesInZipFile 678
GetJobs 679
GetJobs method 679
GetLocalizationUrl 680
GetTranslationUploadDirectory 680
Localization methods 673
StartExportContentForTranslation 680
StartExportFolderForTranslation 681
StartExportMenusForTranslation 681
StartImportTranslation 682
Login class 337
about Login class 337
AutoAddType 338
AutoLogin 339
Fill 338
Login methods 338
Login properties 338
OnlyAllowMemberLogin 339
Password 339
PromptLogout 339
SuppressHelpButton 339
Text 340
UserName 340

## M

Map class 340
about Map class 340
Address 342
ContentId 343
CustomIcon 343
DateSearch 343
DisplayTextResult 343
DistanceUnit 344
EnableSearchResult 344
Fill 341
FolderId 344
ForceCategory 345
GeoControl 345
GetCallbackResult 341
Latitude 345
Longitude 346
Map methods 340
Map properties 341
MapCategory 346
MapProvider 346
MapStyle 346
MarkupLanguage 347
MinZoomLevel 347
PageSize 347
RaiseCallbackEvent 341
Recursive 347
SearchData 348
SelectedCategoryLogicalType 348
StartZoomLevel 348
Stylesheet 348
TemplateParamName 349
TemplateUserProfile 349
TypeControl 349
ZoomControl 349

MediaData class 1288
about MediaData class 1288
Alt 1289
FileName 1289
FilePath 1289
Height 1289
Id 1289
MediaData properties 1288
MimeType 1290
Sequence 1290
Title 1290
Width 1290
MediaGalleryData class 1291
about MediaGalleryData class 1291
Images 1291
MediaGalleryData properties 1291
ObjectId 1291
Membership class 350
about Membership class 350
AllowAvatarUpload 351
AutoLogin 352
BoardID 352
DisplayMode 352
DynamicParameter 353
Fill 350
JavascriptEditorHeight 353
JavascriptEditorWidth 353
Membership methods 350
Membership properties 350
RedirectFailedURL 354
RedirectSuccessURL 354
RegisterButtonImg 354
RegisterButtonText 354
ResetButtonImg 355
ResetButtonText 355
ResetPasswordMessageId 355
SaveButtonImg 355
SaveButtonText 356
ShowExtended 356
ShowTerms 356
TaxonomyId 356
Text 357
UserExistsMessage 357
UserID 357
UserSuccessMessage 357
UserUpdateSuccessMessage 358
Menu class 358
about Menu class 358
CacheInterval 359
DefaultMenuID 360
DisplayXslt 360
DynamicParameter 360
Fill 358
Menu methods 358
Menu properties 359
Stylesheet 361
SuppressAddEdit 361
Wrap 359
MessageBoard class 361, 755, 1086
about MessageBoard class 361, 755, 1086
AddMessageBoardEntry 755, 1087
ApproveMessage 756, 757, 1088, 1089
ApproveMessage method 756, 1088
CacheInterval 363
DefaultObjectID 364
DeleteMessageBoardEntries 758, 759, 1090, 1091
DeleteMessageBoardEntries method 758, 1090
DeleteMessageBoardEntriesByUser 760, 761, 1092, 1093
DeleteMessageBoardEntriesByUser method 759, 1092
DeleteMessageBoardEntry 762, 1094, 1095
DeleteMessageBoardEntry method 761, 1094
DisplayXslt 364
DynamicObjectParameter 364
EnablePaging 365
Fill 362
GetAllMessageBoardEntries 764, 1097, 1098
GetAllMessageBoardEntries method 763, 1096
GetAllMessageBoardEntriesByUser 765, 766, 1099, 1100

GetAllMessageBoardEntriesByUser method 765, 1098
GetMessageBoardEntry 767, 768, 1101, 1102
GetMessageBoardEntry method 767, 1101
IsUserMessageBoardAdmin 769, 1103
MarkupLanguage 365
MaxResults 366
MessageBoard methods 362, 755, 1087
MessageBoard properties 363
Moderate 366
ObjectType 366
ProfileParamName 367
RaiseCallbackEvent 362
ShowMaxCharacter 367
Stylesheet 367
TemplateUserProfile 368
ToString 362
UpdateMessageText 770, 1104, 1105
UpdateMessageText method 769, 1103
Wrap 362
XmlDoc 368
MessageBoardInfo class 562
about MessageBoardInfo class 562
ErrorMessage 562
MessageBoardInfo 562
MessageBoardInfo properties 562
Messaging class 368, 682
about Messaging class 368, 682
DisplayXslt 370
EnablePreSearch 370
Fill 369
FriendsOnly 371
GetUnreadMessageCountForUser 682
GroupParamName 371
JavascriptEditorHeight 371
JavascriptEditorHTMLMode 372
JavascriptEditorToolbar 372
JavascriptEditorWidth 373
Messaging methods 369, 682
Messaging properties 369
ProfileParamName 373
ProfileUrl 373
RecipientParamName 374
ShowSearchOptions 374
Stylesheet 374
ToString 369
UserSearchDisplayXslt 375
Wrap 369
XmlDoc 375
Metadata class 683, 1106
about Metadata class 683, 1106
AddMetaDataType 684, 1106
AddMetaDataType method 683
DeleteMetadataType 686, 1107
GetContentMetadataList 686, 688, 1107, 1108
GetContentMetadataList method 686, 1107
GetIntegratedSearchFolder 688
GetIntegratedSearchFolders 688
GetMetadataByFolder 689
GetMetadataType 689, 1108
GetMetaDataTypes 689, 1109
Metadata methods 683, 1106
UpdateMetadataType 690, 1109
MetaData class 375
about MetaData class 375
CacheInterval 376
DefaultContentID 377
DefaultItemList 377
DynamicParameter 377
Fill 376
GenerateDublinCore 378
MetaData methods 376
MetaData properties 376
MetaDataList class 378
about MetaDataList class 378
CacheInterval 382
ContentType 383
CopyTo 379
Count 383

Current 383
EkItems 383
EnablePaging 384
ExactPhrase 384
Fill 379
FolderID 385
GetEnumerator 379
GetHtml 385
IncludeIcons 385
IsSynchronized 386
KeyWordName 386
KeyWordValue 386
KeyWordValueMatchAll 387
KeyWordValueSeparator 387
LinkTarget 387
MarkupLanguage 388
MaxNumber 388
MetaDataList methods 379
MetaDataList properties 380
MoveNext 380
OrderBy 388
Recursive 389
Reset 380
SortOrder 389
SuppressWrapperTags 390
SyncRoot 390
this 380
Wrap 380
XmlDoc 390
MyAccount class 390
about MyAccount class 390
CustomPropertyID 392
DefaultCountryId 392
DisplayXslt 393
EkItem 393
Fill 391
LoadingImage 393
MyAccount methods 391
MyAccount properties 391
RedirectUrl 394
Stylesheet 394
ToString 391
UseMsAjax 394
Wrap 391

## O

OptionGroupData class 1292
about OptionGroupData class 1292
Image 1293
OptionGroupData 1292, 1293
OptionGroupData constructor 1292
OptionGroupData properties 1293
Options 1293
OptionGroupItemCollection class 1293
about OptionGroupItemCollection class 1293
AddRange 1294
Find 1294
FindAll 1295
FindIndex 1295
OptionGroupItemCollection 1294
OptionGroupItemCollection methods 1294
OptionGroupItemData class 1295
about OptionGroupItemData class 1295
CurrencyId 1297
ExtraText 1297
Name 1297
OptionGroupItemData 1296
OptionGroupItemData constructor 1295
OptionGroupItemData properties 1296
PriceModification 1297
PriceModificationType 1297
OrderApi class 608
about OrderApi class 608
Capture 609, 610
Capture method 609
GetCustomerOrderList 610
GetItem 610
GetList 611

GetList method 611
GetOnHoldOrderList 612
GetOnHoldOrderList method 611
GetOrderMessage 613
GetOrderMessage method 612
GetOrderPayment 613
GetOrderStatus 614
GetProductOrderList 614
GetProductOrderList method 614
GetReport 615
ItemsInStock 615
MarkAsFraud 615
OrderApi methods 609
ProcessOrderInventory 615
RollbackOrderInventory 616
SetOrderStatus 616
SetOrderToHold 616
SetTrackingNumber 616
OrderData class 1298
about OrderData class 1298
BillingAddressId 1299
ClientIP 1299
CouponTotal 1299
CouponUsed 1299
Currency 1299
Customer 1300
CustomerType 1300
DateCompleted 1300
DateCreated 1300
DateRequired 1300
HandlingTotal 1301
HttpHost 1301
Id 1301
OrderData properties 1298
OrderTotal 1301
Parts 1301
ShippingTotal 1301
SpecialInstructions 1302
StageId 1302
StageName 1302
Status 1302
Subtotal 1302
TaxTotal 1303
VisitorId 1303
OrderLineData class 1303
about OrderLineData class 1303
Configuration 1304
ItemType 1304
LineItemId 1304
OrderLineData properties 1303
OrderPartId 1304
PriceEach 1304
PriceSubTotal 1305
PriceTotal 1305
ProductId 1305
ProductLanguage 1305
Quantity 1305
OrderList class 395
about OrderList class 395
CacheInterval 396
DefaultUserID 397
DisplayXslt 397
DynamicOrderParameter 398
DynamicProductParameter 398
EkItem 398
Fill 395
OrderList methods 395
OrderList properties 396
PageSize 399
Stylesheet 399
TemplateProduct 399
ToString 396
Wrap 396
OrderPartData class 1306
about OrderPartData class 1306
DateShipped 1306
HandlingTotal 1307
Lines 1307

OrderId 1307
OrderPartData properties 1306
PartId 1307
PartTotal 1307
ShippingAddressId 1308
ShippingTotal 1308
Status 1308
Subtotal 1308
SupplierId 1308
TaxTotal 1309
TrackingNumber 1309

## P

PackageApi class 617
about PackageApi class 617
Add 617
Delete 618
GetItem 618
GetList 618
PackageApi methods 617
Update 618
PackageData class 1309
about PackageData class 1309
Dimensions 1311
Id 1311
MaxWeight 1311
Name 1311
PackageData 1310
PackageData constructor 1309
PackageData properties 1310
PaymentAuthorization class 1311
about PaymentAuthorization class 1311
AuthorizedOn 1312
CapturedOn 1312
CreditedOn 1312
Gateway 1312
Message 1313
PaymentAuthorization properties 1312
RecurringTransactionId 1313
TransactionId 1313
VoidedOn 1313
Permissions class 690, 1110
about Permissions class 690, 1110
AddItemPermission 691, 1110
DeleteItemPermission 691, 1111
DisableContentInheritance 691, 1111
DisableFolderInheritance 692, 1112
EnableContentInheritance 692, 1112
EnableFolderInheritance 692, 1113
GetUserPermissions 693, 1113, 1114
GetUserPermissions method 692, 1113
LoadPermissions 694, 1114
Permissions methods 690, 1110
UpdateItemPermission 694, 1115
PersonalizationManager class 400
about PersonalizationManager class 400
AddContentExitText 401
AddContentText 401
CurrentUserButtonText 402
Display 402
Fill 400
HeaderBackColor 402
HeaderText 403
PersonalizationManager methods 400
PersonalizationManager properties 401
PersonalizeExitText 403
PersonalizeScopeText 403
PersonalizeStartText 404
PublicUserButtonText 404
ResetLinkText 404
ResetLinkToolTip 405
UrlCompareMode 405
PhotoGallery class 406
about PhotoGallery class 406
Fill 406
PhotoGallery methods 406
PipelineStageData class 1313
about PipelineStageData class 1313

Name 1314
PipelineStageData properties 1314
Sequence 1314
StatusId 1314
Plug-in Extension API 524
Poll class 406
about Poll class 406
EnableAjax 408
Fill 407
Poll methods 407
Poll properties 408
PollID 408
WrapPoll 407
PostalCodeTaxRateData class 1314
about PostalCodeTaxRateData class 1314
PostalCode 1316
PostalCodeTaxRateData 1315, 1316
PostalCodeTaxRateData constructor 1315
PostalCodeTaxRateData properties 1316
RegionId 1316
PostHistory class 409
about PostHistory class 409
BoardID 410
CacheInterval 410
Fill 409
MaxNumber 411
PostHistory methods 409
PostHistory properties 410
ToString 409
URLPath 411
UserID 411
Wrap 410
PostLoginEventArgs class 412
about PostLoginEventArgs class 412
PricingData class 1316
about PricingData class 1316
CurrencyPricelist 1320
GetCurrencyById 1318
GetCurrencyByIso 1318
GetCurrencyByIsoNumber 1319
GetListPrice 1319
GetListPriceByCurrencyId 1319
GetListPriceByIsoNumber 1319
GetSalePrice 1319
GetSalePriceByCurrencyId 1320
GetSalePriceByIsoNumber 1320
IsRecurringPrice 1321
PricingData 1317
PricingData constructor 1317
PricingData methods 1318
PricingData properties 1320
Recurrence 1321
Validate 1320
Product class 412
about Product class 412
CacheInterval 414
DefaultProductID 415
DisplayXslt 415
DynamicParameter 415
EkItems 416
Fill 413
ImageGallery 416
OverrideXslt 417
Product methods 412
Product properties 413
RedirectToBasketPage 413
ShowAddToCart 417
Stylesheet 417
TemplateCart 418
ToString 413
Wrap 413
ProductCouponEntryData class 1349
about ProductCouponEntryData class 1349
EntryType 1351
ProductCouponEntryData 1350
ProductCouponEntryData constructor 1350
ProductCouponEntryData properties 1350
ProductData class 1321

about ProductData class 1321
ProductData 1321, 1322
ProductData constructor 1321
ProductData properties 1322
Variants 1322
ProductList class 418
about ProductList class 418
DisplayXslt 420
DynamicParameter 421
EkItems 421
Fill 419
IdList 421
LoadingImage 422
PageSize 422
PageSpan 422
ProductList methods 418
ProductList properties 419
Recursive 423
SortMode 423
SourceId 424
SourceType 424
Stylesheet 424
TaxonomyDepth 425
TemplateProduct 425
ToString 419
UseMsAjax 426
Wrap 419
ProductSearch class 426
about ProductSearch class 426
ButtonImgSrc 430
ButtonText 430
CacheInterval 431
CatalogId 431
CustomOrderBy 431
DisplayXslt 432
DynamicProductParameter 432
EnableAdvancedLink 432
EndCount 433
Fill 427
GetCallbackResult 427
IsInWorkArea 433
LinkTarget 433
LoadingImage 434
MaxCharacters 434
MaxTeaserLength 434
OrderBy 435
OrderDirection 435
ProductSearch 426
ProductSearch methods 427
ProductSearch properties 428
RaiseCallbackEvent 427
Recursive 436
ResultCount 436
ResultDataTable 436
ResultsPageSize 437
ResultTagId 437
ResultXml 438
SearchText 438
ShowCategories 438
ShowCustomSummary 438
ShowSearchBoxAlways 439
StartCount 439
Stylesheet 439
TaxCategories 440
TaxonomyOperator 440
TemplateCart 441
TemplateProduct 441
TextBoxSize 441
TimeTaken 442
Wrap 427
XmlDoc 442
ProductTypeApi class 619
about ProductTypeApi class 619
Add 619
AddProductAttribute 620
Delete 620
DeleteProductAttribute 620, 621
DeleteProductAttribute method 620

GetFolderProductTypeList 621
GetItem 621
GetItem method 621
GetProductAttribute 622
GetProductAttributeList 622
ProductTypeApi methods 619
Update 622
UpdateProductAttribute 622
ProductTypeAttributeData class 1322
about ProductTypeAttributeData class 1322
ActiveStatus 1324
DataType 1324
DefaultValue 1324
DisplayOrder 1324
Id 1324
Name 1325
ProductTypeAttributeData 1323
ProductTypeAttributeData constructor 1323
ProductTypeAttributeData properties 1323
ProductTypeData class 1325
about ProductTypeData class 1325
Attributes 1325
DefaultThumbnails 1326
EntryClass 1326
ProductTypeData properties 1325

## R

Recommendation class 442
about Recommendation class 442
CacheInterval 444
DefaultProductID 445
DisplayXslt 445
DynamicCartParameter 445
DynamicProductParameter 446
EkItem 446
Fill 443
PageSize 447
Recommendation methods 443
Recommendation properties 443
RecommendationType 447
Stylesheet 447
TemplateCart 448
TemplateProduct 448
ToString 443
Wrap 443
RecommendationApi class 622
about RecommendationApi class 622
Add 623
Delete 623
DeleteRecommendationByEntry 624
GetItem 624
RecommendationApi methods 623
Update 624
UpdateCrossSell 624
UpdateUpSell 625
RecommendationItemData class 1326
about RecommendationItemData class 1326
EntryLanguage 1328
EntryType 1328
Html 1328
Image 1328
ImageThumbnail 1329
Pricing 1329
Quicklink 1329
RecommendationItemData 1327
RecommendationItemData constructor 1326
RecommendationItemData properties 1328
Summary 1329
Title 1329
Type 1330
RegionApi class 625
about RegionApi class 625
Add 625
CanDelete 626
Delete 626
GetItem 626
GetList 626
RegionApi methods 625

Update 627
RegionData class 1330
about RegionData class 1330
Code 1332
CountryId 1332
Enabled 1332
Id 1332
Name 1332
RegionData 1330, 1331
RegionData constructor 1330
RegionData methods 1331
RegionData properties 1331
ToString 1331
Validate 1331
RegionTaxRateData class 1333
about RegionTaxRateData class 1333
RegionData 1334
RegionTaxRateData 1333, 1334
RegionTaxRateData constructor 1333
RegionTaxRateData properties 1334
RequestInfo class 563
about RequestInfo class 563
ReviewInfo class 563
about ReviewInfo class 563
AverageRating 564
ContentHtml 564
ContentID 564
ContentLanguage 564
ContentTitle 564
ErrorMessage 565
ID 565
NumberofRatings 565
Rating 565
RatingDate 565
ReviewInfo properties 563
State 565
UserAvatar 566
UserComments 566
UserDisplayName 566
UserFirstName 566
UserID 566
UserLastName 566
UserName 567
VisitorID 567
RssAggregator class 448
about RssAggregator class 448
CopyTo 449
Count 451
Current 452
EkItems 452
Fill 449
GetEnumerator 450
GetView 450
GetViewNames 450
IsSynchronized 452
LinkTarget 452
MaxResults 453
MoveNext 450
Reset 451
RssAggregator methods 449
RssAggregator properties 451
SyncRoot 453
this 451
URL 453
RssItemBase class 453
about RssItemBase class 453
DatePosted 454
Description 454
Link 454
RssItemBase fields 454
Title 454

## S

Search class 455
about Search class 455
ButtonImgSrc 460
ButtonText 460
ContentFieldsetLegend 461

CopyTo 456
Count 461
Current 461
Display 461
EkItems 462
EmptyResultMsg 462
EnableAdvancedLink 462
EnableAjax 462
EnableArchived 463
EnableArchivedBtn 463
EnableContent 463
EnableContentBtn 464
EnableContentFieldset 464
EnableDMSAssets 464
EnableDMSAssetsBtn 465
EnableFolderFieldset 465
EnableForms 465
EnableFormsBtn 465
EnableLibFiles 466
EnableLibFilesBtn 466
EnableLibHyperlinks 466
EnableLibHyperlinksBtn 467
EnableLibImages 467
EnableLibImagesBtn 467
EnableLibraryFieldset 468
Fill 456
FolderFieldsetLegend 468
FolderID 468
FolderPath 469
GetEnumerator 456
IncludeIcons 469
IsSynchronized 469
LibraryFieldsetLegend 469
LinkTarget 470
LoadSearch 456
MaxCharacters 470
MaxResults 470
MaxTeaserLength 471
MoveNext 457
OrderBy 471
OrderDirection 472
PostResultsAt 472
Recursive 472
RemoveTeaserHtml 472
Reset 457
ResultTagId 473
Search 455
Search methods 455
Search properties 458
SearchDisplayUI 457
SearchText 473
SearchType 473
ShowDate 473
ShowExtendedSearch 474
ShowSearchBoxAlways 474
ShowSearchOptions 475
Stylesheet 475
SyncRoot 475
TargetFolderDescriptionList 475
TextBoxSize 476
this 457
Wrap 457
XmlDoc 476
SearchManager class 865, 1115
about SearchManager class 865, 1115
Execute 866, 867
Execute method 866
ExecuteSearch 1116, 1117
ExecuteSearch method 1116
Search 868, 869, 1118, 1119
Search method 867, 1117
SearchForCategories 870, 1119
SearchManager methods 865, 1116
SearchTable 870, 1119
Server Controls API 7
ShippingBaseData class 1334
about ShippingBaseData class 1334
Id 1335

Name 1335
ProviderService 1335
ShippingBaseData 1334
ShippingBaseData properties 1335
ShippingMethodApi class 627
about ShippingMethodApi class 627
Add 627
ShippingMethodApi methods 627
Update 628
ShippingMethodData class 1355
about ShippingMethodData class 1355
DisplayOrder 1356
IsActive 1356
ShippingMethodData 1356
ShippingMethodData properties 1356
ShippingOptionData class 1335
about ShippingOptionData class 1335
CurrencyId 1336
ShippingFee 1336
ShippingOptionData 1336
ShippingOptionData properties 1336
Site class 694, 1120
about Site class 694, 1120
BreakTaskPermissionInherit 695, 1121
DeleteCookie 696
GetAllActiveLanguages 696, 1121
GetAllLanguages 696, 697, 1122
GetAllLanguages method 696, 1122
GetClientDayNames 697, 1123
GetClientMonthNames 697, 1123
GetCookieValue 697, 1123
GetLanguage 698, 1124
GetLanguages 698, 1124
GetSiteVariables 698, 1125
GetTaskPermission 699, 1125
IsTaskPermissionInherited 699, 1126
SetCookieValue 699, 1126
SetTaskPermission 700, 1126
Site methods 695, 1120
UpdateSiteEnabledLanguages 700, 1127
UpdateSiteVariables 700
SiteInfo class 567
about SiteInfo class 567
AppPath 567
IsSSL 568
Name 568
PhysicalSitePath 568
Port 568
SiteInfo properties 567
Url 568
Sitemap class 476
about Sitemap class 476
CacheInterval 477
ClassName 478
DisplayType 478
DisplayXslt 479
Fill 477
FlatTopLevel 479
FolderID 479
MaxLevel 480
Sitemap methods 477
Sitemap properties 477
StartingLevel 480
Stylesheet 480
XmlDoc 480
SiteMap class 701, 1127
about SiteMap class 701, 1127
GetFolderBreadcrumbPath 701, 1128
GetWebsiteMap 703, 1128
RenderFolderBreadcrumbHtml 704, 1129
RenderWebsiteMap 705, 1129
SiteMap methods 701, 1128
SmartMenu class 481
about SmartMenu class 481
AutoCollapseBranches 482
DisplayXslt 482
Enable508Compliance 482
Enable508NavigationHeadings 483

EnableMouseOverPopUp 483
EnableSmartOpen 483
EnableSubmenuLinkPathMatching 484
EnableSubmenuTemplateMatching 484
LaunchLinksInNewWindow 484
ShowRootFolder 485
SingleEditButton 485
SmartMenu properties 481
StartCollapsed 486
StartWithRootCollapsed 486
Stylesheet 486
SocialBar class 487
about SocialBar class 487
CacheInterval 489
DefaultObjectID 489
DisplayXslt 489
DynamicObjectParameter 490
EkItems 490
Fill 487
GetCallbackResult 487
IncludeIcons 490
InviteURL 491
Items 491
MarkupLanguage 491
MessagingURL 492
ObjectType 492
SocialBar methods 487
SocialBar properties 488
Stylesheet 492
SuppressWrapperTags 493
ToString 488
Wrap 488
XmlDoc 493
Subscription class 705
about Subscription class 705
AddEmailFrom 706
AddSubscription 707
AddSubscriptionMessage 707
DeleteEmailFrom 707
DeleteSubscription 707
DeleteSubscriptionMessage 708
GetAllActiveSubscriptions 708
GetAllEmailFrom 708
GetAllSubscriptionMessages 708
GetAllSubscriptions 709
GetFolderInheritedFrom 709
GetSubscription 709
GetSubscriptionMessage 709
GetSubscriptionMessagesForType 710
GetSubscriptionPropertiesForContent 710
GetSubscriptionPropertiesForFolder 710
GetSubscriptionsForContent 711
GetSubscriptionsForFolder 711
SendWebAlert 711
Subscription methods 706
UpdateEmailFrom 712
UpdateSubscription 712
UpdateSubscriptionMessage 712
UpdateSubscriptionPropertiesForContent 712
UpdateSubscriptionPropertiesForFolder 713


## T


TagCloud class 493
about TagCloud class 493
CacheInterval 495
DisplayXslt 496
DynamicParameter 496
Fill 494
MarkupLanguage 496
MaxTagsDisplayed 496
OrderBy 497
OrderByDirection 497
Stylesheet 498
TagCloud methods 493
TagCloud properties 494
TagTemplate 498
TagTemplateTarget 499
TagType 499

ToString 494
Wrap 494
XmlDoc 499
Tags class 771, 1130
about Tags class 771, 1130
AddTag 772, 1131
AddTagToCommunityGroup 773, 774, 1132, 1133
AddTagToCommunityGroup method 772, 1132
AddTagToContent 775, 1134, 1135
AddTagToContent method 774, 1134
AddTagToObject 776, 777, 1136
AddTagToObject method 775, 1135
AddTagToUser 778, 1137, 1138
AddTagToUser method 777, 1137
DeleteTagById 779, 1139
DeleteTagOnObject 779, 1139
DeleteTagsByUser 779, 1140
DeleteTagsOnCommunityGroup 780, 1140
DeleteTagsOnContent 780, 1141
DeleteTagsOnObject 780, 1141
DeleteTagsOnUser 781, 1142
EditTag 781, 1142
GetAllTagRelatedContent 781, 1142
GetAllTagRelatedUsers 781, 1143
GetAllTags 782, 783, 1144
GetAllTags method 782, 1143
GetDefaultTags 783, 784, 1145, 1146
GetDefaultTags method 783, 1145
GetTagByID 784, 1146
GetTagByName 784, 1146
GetTagCloud 785, 1147
GetTagsForContent 785, 1147
GetTagsForObject 786, 1148, 1149
GetTagsForObject method 785, 1148
GetTagsForUser 787, 1150
GetTagsForUser method 786, 1149
RemoveTagAsDefault 788, 1150
SaveTagAsDefault 788, 1151
Tags methods 771, 1130
Task class 871, 957
about Task class 871, 957
AddTask 872, 957
AssignTaskToUser 872, 873, 958, 959
AssignTaskToUser method 872, 958
AssignTaskToUsergroup 873, 874, 959, 960
AssignTaskToUsergroup method 873, 959
DeleteTask 874, 875, 960, 961
DeleteTask method 874, 960
DeleteTasks 875, 876, 961, 962
DeleteTasks method 875, 961
GetTask 876, 962
GetTasks 876, 877, 963
GetTasks method 876
GetTasksAssignedByUser 877, 878, 963, 964
GetTasksAssignedByUser method 877, 963
GetTasksAssignedToUser 878, 879, 965
GetTasksAssignedToUser method 878, 964
GetTasksAssignedToUsergroup 880, 966
GetTasksAssignedToUsergroup method 879, 965
GetTaskState 881, 967
SetStateForTask 881, 967
Task methods 871, 957
UpdateTask 881, 967
TaskCategory class 882, 968
about TaskCategory class 882, 968
AddTaskCategory 882, 969
DeleteTaskCategory 882, 969
GetAllCategories 883, 969
GetTaskCategory 883, 970
TaskCategory methods 882, 968
UpdateTaskCategory 883, 970
TaskCategoryType class 884, 971
about TaskCategoryType class 884, 971
AddTaskType 884, 971
DeleteTaskType 885, 972
GetTaskType 885, 972
TaskCategoryType methods 884, 971
UpdateTaskType 885, 972

TaxApi class 628
about TaxApi class 628
Add 629
Delete 629
GetApplicableTaxRate 629
GetApplicableTaxRateList 629
GetItem 630
GetItem method 630
GetItemByCountryId 630
GetItemByPostalCode 631
GetItemByPostalCode method 631
GetItemByPostalId 632
GetItemByRegionId 632
GetList 632
TaxApi methods 628
Update 632
TaxClassApi class 633
about TaxClassApi class 633
Add 633
Delete 633
GetItem 634
GetList 634
TaxClassApi methods 633
Update 634
TaxClassData class 1336
about TaxClassData class 1336
Id 1338
Name 1338
TaxClassData 1337, 1338
TaxClassData constructor 1337
TaxClassData properties 1338
Taxonomy class 842, 1151
about Taxonomy class 842, 1151
AddTaxonomyItem 843, 1153
AddTaxonomyItem method 843, 1152
AddTaxonomySyncFolder 843, 1153
CloneTaxonomy 844
CreateTaxonomy 844, 1154
DeleteTaxonomy 844, 1154
GetTaxonomyIdByPath 845, 1154
ImportTaxonomy 845, 846
ImportTaxonomy method 845
LoadTaxonomy 846, 1155
ReadAllAssignedCategory 846, 1155
ReadTaxonomy 847, 1156
RemoveTaxonomyFolder 847, 1156
RemoveTaxonomyItem 847, 1156
ReOrderTaxonomyItems 847, 1157
Taxonomy methods 842, 1152
UpdateTaxonomy 848, 1157
TaxonomyRequest class 1414
about TaxonomyRequest class 1414
Depth 1415
FolderId 1415
IncludeItems 1415
Preview 1416
ReadCount 1416
TaxonomyIdList 1416
TaxonomyItemType 1417
TaxonomyPath 1417
TaxonomyRequest properties 1414
TaxRateData class 1338
about TaxRateData class 1338
Id 1340
Rate 1340
TaxClassId 1340
TaxRateData 1339
TaxRateData constructor 1339
TaxRateData properties 1340
TypeId 1340
TypeItemId 1341
TaxTypeApi class 635
about TaxTypeApi class 635
Add 635
Delete 635
GetItem 636
GetList 636
TaxTypeApi methods 635

Update 636
TaxTypeData class 1341
about TaxTypeData class 1341
Id 1342
Name 1343
Precedence 1343
TaxTypeData 1341, 1342
TaxTypeData constructor 1341
TaxTypeData properties 1342
ThreadedDiscussion class 848, 1157
about ThreadedDiscussion class 848, 1157
AddBoard 849, 1158
AddForum 850, 1159
AddReply 852, 853
AddReply method 852
AddTopic 854, 1160
AddTopic method 853, 1160
DeleteBoard 854, 1161
DeleteForum 855, 1161
DeleteTopic 855, 1162
GetActiveTopics 855, 856
GetActiveTopics method 855
GetActiveTopicsForForum 856
GetActiveUsers 856
GetBoardCategories 857, 1162
GetBoardForums 857, 1162
GetDiscussionBoard 858, 1163, 1164
GetDiscussionBoard method 857, 1163
GetDiscussionBoardString 858, 1164
GetForum 859, 1165
GetRepliesForTopics 860, 1165
GetTopic 861, 1166
SearchReplies 861, 1166
SubscribeToBoard 862
SubscribeToForum 862
SubscribeToTopic 862
ThreadedDiscussion methods 848, 1158
UnSubscribeToBoard 863
UnSubscribeToForum 863
UnSubscribeToTopic 864
ThumbnailData class 1343
about ThumbnailData class 1343
Height 1344
ThumbnailData properties 1343
Width 1344
ThumbnailDefaultData class 1344
about ThumbnailDefaultData class 1344
FileNameFormat 1345
Height 1345
Id 1345
MimeType 1345
ThumbnailDefaultData properties 1344
Title 1345
Width 1346
TierPriceData class 1346
about TierPriceData class 1346
Clone 1347
CompareTo 1348
DataState 1348
Id 1348
Quantity 1349
RecurrenceId 1349
SalePrice 1349
TierPriceData 1347
TierPriceData constructor 1346
TierPriceData methods 1347
TierPriceData properties 1348
Validate 1348

## U

UriFavorite class 1417
about UriFavorite class 1417
FavoriteDesc 1418
FavoriteId 1418
FavoriteName 1418
FavoriteUri 1419
UriFavorite properties 1418
UserId 1419

UrlAliasAuto class 886
about UrlAliasAuto class 886
Add 887
ClearCache 887
Delete 888
GetAutoAliasNameTypes 888
GetAutoAliasTypes 888
GetFolderItem 888
GetFolderList 889
GetFolderSource 889
GetItem 889
GetItemForContent 890
GetList 890
GetListForContent 891
GetTarget 892
GetTarget method 891
GetTaxonomyItem 893
GetTaxonomyList 893
GetTaxonomySource 893
Update 893
UrlAliasAuto methods 887
UrlAliasAutoData class 1365
about UrlAliasAutoData class 1365
AliasName 1366
AutoAliasType 1366
AutoId 1366
Example 1366
ExcludedPath 1367
FileExtension 1367
FormattedSelectedPath 1367
HostName 1367
IsEnabled 1367
LanguageId 1368
PageNameType 1368
ReplacementCharacter 1368
SelectedPath 1368
SiteId 1369
SourceId 1369
SourceName 1369
Target 1369
UrlAliasAutoData properties 1365
UrlAliasCommon class 894
about UrlAliasCommon class 894
GetAliasForContent 894
GetTarget 895
GetTarget method 894
UrlAliasCommon methods 894
UrlAliasManual class 895
about UrlAliasManual class 895
Add 896
ClearCache 896
Delete 896
GetDefaultAlias 897
GetFileExtensions 897
GetItem 897
GetList 898, 899, 900, 901
GetList method 898
GetSiteList 902
GetTemplateList 902
Update 902
UrlAliasManual methods 896
ValidateName 903
UrlAliasManualData class 1370
about UrlAliasManualData class 1370
AliasId 1370
AliasName 1371
AliasPageName 1371
ContentId 1371
ContentLanguage 1371
ContentTitle 1372
DisplayAlias 1372
FileExtension 1372
HostName 1372
IsDefault 1373
IsEnabled 1373
LibraryId 1373
QueryString 1373
QueryStringAction 1373

SiteId 1374
Target 1374
UrlAliasManualData properties 1370
UrlAliasRegEx class 903
about UrlAliasRegEx class 903
Add 904
ClearCache 904
Delete 904
EvaluateExpression 904
GetItem 905
GetList 905
GetPriorityList 906
GetTarget 906
Update 906
UrlAliasRegEx methods 903
UrlAliasRegExData class 1374
about UrlAliasRegExData class 1374
Expression 1375
ExpressionMap 1375
ExpressionName 1375
IsEnabled 1376
Priority 1376
RegExId 1376
SiteId 1377
TransformedUrl 1377
UrlAliasRegExData properties 1375
UrlAliasSettings class 907
about UrlAliasSettings class 907
AutoCacheDuration 909
AutoCacheSize 909
DefaultPage 909
DisableLanguageAwareness 909
EnableAutoAliasing 909
EnableAutoCaching 910
EnabledTracing 910
EnableManualAliasing 910
EnableManualCaching 910
EnableRegExAliasing 910
EnableRegExCaching 911
IsAutoAliasEnabled 911
IsManualAliasEnabled 911
IsRegExAliasEnabled 911
ManualCacheDuration 912
ManualCacheSize 912
OverrideTemplate 912
PageExtensions 912
QueryStringAction 912
RefreshSettingsCache 907
RegExCacheDuration 912
RegExCacheSize 913
UrlAliasSettings methods 907
UrlAliasSettings properties 907
User class 920, 1167
about User class 920, 1167
ActivateUserAccount 923, 1169
ActivateUserAccount method 922, 1169
AddMembershipUser 923, 1170
AddNewUser 925, 1170
AddRolePermission 926, 1171
AddUser 926, 927, 1171
AddUser method 926
AddUserGroup 927, 1172
AddUserToGroup 927, 1172
AutoLogInUser 928, 1172
CreateBlankUserObject 928, 929, 1173, 1174
CreateBlankUserObject method 928, 1173
DeleteUserByID 929, 1174
DeleteUserByIds 929, 1174
DeleteUserFromGroup 929, 1175
DeleteUserGroup 930, 1175
EditUserCustomProperties 930, 1176
GetActiveUser 930, 1176
GetADMapping 931, 1176
GetAllCmsUsers 931, 1177
GetAllCustomProperties 931, 1178
GetAllUserByCustomProperty 932, 1178
GetAllUserGroups 932, 933, 1179, 1180
GetAllUserGroups method 932, 1179

GetAllUsers 933, 934, 1181
GetAllUsers method 933, 1180
GetGroupsUserIsIn 934, 1181
GetMembershipUser 934, 1182
GetUser 935, 1182
GetUserbyUsername 1183
GetUserByUsername 935
GetUserCustomProperties 937, 1183
GetUserEmailInfo 937, 1184
GetUserForFolderAdmin 937, 1184
GetUserGroup 937, 1185
GetUserGroupByName 938
GetUserGroupForFolderAdmin 938, 1185
GetUserPermissions 938, 1185
GetUserPreference 939, 1186
GetUsers 940, 1187, 1188
GetUsers method 939, 1187
GetUsersForTask 940, 1188
GetUsersNotInGroup 941, 1189
IsAdmin 941, 1189
IsAGroupMember 942
IsARoleMember 943, 944, 1191, 1192
IsARoleMember method 942, 1190
IsARoleMemberForContent 945, 1193
IsARoleMemberForContent_FolderUserAdmin 945
IsARoleMemberForFolder 946, 1193
IsARoleMemberForFolder_FolderUserAdmin 946
IsARoleMemberForFolder_MetadataAdmin 946
LogInUser 947, 948, 1194
LogInUser method 947
ReadCustomProperties 950
ResetMembershipUserPassword 950, 951, 1195
ResetMembershipUserPassword method 950, 1194
SetAuthenticationCookie 951
UnsubscribeUser 951, 1195
UpdateCustomPropertiesItemOrder 952, 1196
UpdateMapCoordinate 1196
UpdateUser 952, 953, 1197, 1198
UpdateUser method 952, 1197
UpDateUserGroup 953, 1198
UpDateUserGroup method 953
UpdateUserPreferences 954, 1198
User methods 921, 1167
User properties 955
UserIdAuthenticated 955
UserObject 954, 1199, 1200
UserObject method 954, 1199
UserData class 1419
about UserData class 1419
AcceptedTerms 1420
AdUserName 1421
DateCreated 1421
GroupId 1421
GroupName 1421
IsMemberShip 1422
LanguageId 1422
LanguageName 1422
LastLoginDate 1423
LoginAttempts 1423
Rank 1423
Signature 1423
UserData properties 1420
UserPreference 1424
UserInfo class 568
about UserInfo class 568
Addres 569
DisplayName 569
Editor 570
EmailAddress 570
ErrorMessage 570
FirstName 570
ID 570
Language 570
LastName 571
Latitude 571
Longitude 571
Notifications 571
Type 571

UserInfo properties 569
UserName 571
UserProfile class 499
about UserProfile class 499
AvatarHeight 501
AvatarWidth 501
CacheInterval 502
DefaultUserID 502
DisplayXslt 502
DynamicParameter 503
EnableEmailDisplay 503
Fill 500
Stylesheet 503
TagTemplate 504
TagTemplateTarget 504
TaxonomyId 504
ToString 500
UserProfile methods 500
UserProfile properties 501
Wrap 500
XmlDoc 505
UserRequestData class 1424
about UserRequestData class 1424
Group 1425
OrderBy 1425
OrderDirection 1426
RequiredFlag 1426
SearchText 1426
Type 1427
UserRequestData properties 1424
Utilities class 713
about Utilities class 713
LoadPermissions 714
Utilities methods 713


## W


WarehouseApi class 636
about WarehouseApi class 636
Add 637
Delete 637
GetDefaultWarehouse 637
GetItem 638
GetList 638
MarkAsDefault 638
Update 638
WarehouseApi methods 637
WarehouseData class 1351
about WarehouseData class 1351
Address 1352
Id 1352
IsDefaultWarehouse 1352
Name 1353
WarehouseData 1351, 1352
WarehouseData constructor 1351
WarehouseData properties 1352
Web Services API 955
WebSearch class 505
about WebSearch class 505
ButtonImgSrc 510
ButtonText 510
CustomOrderBy 510
CustomSearch 511
DisableForumSearch 511
DisplayXslt 511
DynamicContentTemplate 512
EnableAdvancedLink 512
EndCount 512
Fill 506
FolderID 512
GetCallbackResult 506
IsInWorkArea 513
LinkTarget 513
LoadSearch 506
MarkupLanguage 514
MaxCharacters 514
MaxTeaserLength 514
MultipleFolders 515
OrderBy 515

OrderDirection 515
RaiseCallbackEvent 506
Recursive 516
RemoveTeaserHtml 516
ResultCount 516
ResultDataTable 517
ResultsPageSize 517
ResultTagId 518
ResultXml 518
SearchFor 518
SearchSynonyms 519
SearchText 519
ShowCategories 520
ShowCustomSummary 520
ShowSearchBoxAlways 520
ShowSearchOptions 521
ShowSuggestedResults 521
StartCount 522
Stylesheet 522
TaxCategories 522
TaxonomyOperator 522
TextBoxSize 523
TimeTaken 523
WebSearch 505
WebSearch methods 505
WebSearch properties 507
Wrap 507
XmlDoc 523
Weight class 1353
about Weight class 1353
Amount 1355
Clone 1354
CompareTo 1354
ConvertUnits 1354
Units 1355
Weight 1353, 1354
Weight constructor 1353
Weight methods 1354
Weight properties 1355
Welcome 1