ektron

SDK

ektron CMS 400

API Reference Documentation

Copyright 2007 Ektron, Inc. All rights reserved www.ektron.com
EKTRON is a registered trademark of Ektron, Inc.

Table of Contents

Welcome 1

How Do I... 2

List of Developer Resources 4

Ektron Namespace 7

Server Controls API **7**
- Ektron.Cms Namespace 7
  - Ektron.Cms.Controls Namespace 7
    - Ektron.Cms.Controls.Personalization Namespace 10
    - Classes 11

Plug-in Extension API **325**
- Ektron.Cms.Extensibility Namespace 325
  - Classes 326
    - ContentInfo Class 327
    - ExtensionAuthor Class 330
    - ExtensionDescription Class 332
    - ExtensionEvent Class 333
    - FolderInfo Class 348
    - FormField Class 350
    - FormInfo Class 351
    - RequestInfo Class 353
    - ReviewInfo Class 353
    - SiteInfo Class 358
    - UserInfo Class 359

.NET Assemblies **362**
- API Namespace 363
  - Classes 364
    - BusinessRule Class 365
    - CustomFields Class 367
    - DateTimeSelector Class 377
    - Folder Class 383
    - Font Class 392

Library Class 395
Localization Class 403
Metadata Class 412
Permissions Class 419
Site Class 424
SiteMap Class 430
Subscription Class 435
Utilities Class 441
API.Analytics Namespace 442
Classes 443
API.Calendar Namespace 447
Classes 448
API.Content Namespace 449
Classes 449
API.Search Namespace 514
Classes 514
API.Task Namespace 519
Classes 520
API.User Namespace 534
Classes 534

Web Services API 575
Ektron.Services Namespace 575
Ektron.Services.Task Namespace 576
Classes 576
Classes 592
ActiveDirectoryUser Class 593
Asset Class 601
Blog Class 607
BusinessRules Class 619
Calendar Class 621
CalendarEvent Class 621
CalendarEventType Class 621
Content Class 622
ContentRating Class 634
CustomFields Class 636
Folder Class 644
Font Class 648
Form Class 651
Library Class 658

Metadata Class 667
Permissions Class 671
SearchManager Class 676
Site Class 681
SiteMap Class 688
Taxonomy Class 691
ThreadedDiscussion Class 697
User Class 706

Data Classes 738

ContentData Class 738
ContentData Properties 739
ContentData.ApprovalMethod Property 741
ContentData.Approver Property 742
ContentData.AssetData Property 742
ContentData.Comment Property 742
ContentData.ContType Property 742
ContentData.DateCreated Property 743
ContentData.DisplayDateCreated Property 743
ContentData.DisplayEndDate Property 743
ContentData.DisplayGoLive Property 744
ContentData.DisplayLastEditDate Property 744
ContentData.EditorFirstName Property 744
ContentData.EditorLastName Property 745
ContentData.EditorUserNames Property 745
ContentData.EndDate Property 745
ContentData.EndDateAction Property 745
ContentData.FolderId Property 746
ContentData.FolderName Property 746
ContentData.GoLive Property 746
ContentData.HistoryId Property 746
ContentData.Html Property 747
ContentData.HyperLink Property 747
ContentData.Id Property 747
ContentData.Image Property 748
ContentData.ImageThumbnail Property 748
ContentData.InheritedFrom Property 748
ContentData.IsInherited Property 748

ContentData.IsPrivate Property 749
ContentData.IsPublished Property 749
ContentData.IsSearchable Property 749
ContentData.IsXmlInherited Property 750
ContentData.LanguageId Property 750
ContentData.LastEditDate Property 750
ContentData.ManualAlias Property 750
ContentData.ManualAliasId Property 751
ContentData.MetaData Property 751
ContentData.Path Property 751
ContentData.Permissions Property 751
ContentData.Quicklink Property 752
ContentData.Status Property 752
ContentData.StyleSheet Property 752
ContentData.Teaser Property 753
ContentData.TemplateConfiguration Property 753
ContentData.Title Property 753
ContentData.UserId Property 754
ContentData.UserName Property 754
ContentData.XmlConfiguration Property 754
ContentData.XmlInheritedFrom Property 754

FolderData Class 755

FolderData Properties 755
FolderData.ApprovalMethod Property 757
FolderData.CategoryRequired Property 758
FolderData.ChildFolders Property 758
FolderData.Description Property 758
FolderData.DomainProduction Property 759
FolderData.DomainStaging Property 759
FolderData.FolderTaxonomy Property 759
FolderData.FolderType Property 760
FolderData.HasChildren Property 760
FolderData.Id Property 760
FolderData.Inherited Property 761
FolderData.InheritedFrom Property 761
FolderData.IsCommunityFolder Property 761
FolderData.IsDomainFolder Property 762
FolderData.MetaInherited Property 762
FolderData.MetaInheritedFrom Property 762

FolderData.Name Property 762
FolderData.NameWithPath Property 763
FolderData.ParentId Property 763
FolderData.Permissions Property 763
FolderData.PrivateContent Property 764
FolderData.PublishHtmlActive Property 764
FolderData.PublishPdfActive Property 764
FolderData.PublishPdfEnabled Property 765
FolderData.ReplicationMethod Property 765
FolderData.SitemapInherited Property 765
FolderData.SitemapInheritedFrom Property 766
FolderData.SitemapPath Property 766
FolderData.StyleSheet Property 766
FolderData.StyleSheetInherited Property 767
FolderData.TaxonomyInherited Property 767
FolderData.TaxonomyInheritedFrom Property 767
FolderData.TemplateFileName Property 768
FolderData.TemplateId Property 768
FolderData.TemplateInherited Property 768
FolderData.TotalContent Property 768
FolderData.UpdatedContent Property 769
FolderData.XmlConfiguration Property 769
FolderData.XmlInherited Property 769
FolderData.XmlInheritedFrom Property 770

FolderRequest Class 770

FolderRequest Properties 770
FolderRequest.BreakInheritButton Property 772
FolderRequest.CategoryRequired Property 772
FolderRequest.ContentSubAssignments Property 773
FolderRequest.DomainProduction Property 773
FolderRequest.DomainStaging Property 773
FolderRequest.EnableReplication Property 773
FolderRequest.FileDirectory Property 774
FolderRequest.FolderCfldAssignments Property 774
FolderRequest.FolderDescription Property 774
FolderRequest.FolderId Property 775
FolderRequest.FolderName Property 775
FolderRequest.FolderType Property 775
FolderRequest.ImageDirectory Property 776

FolderRequest.IsDomainFolder Property 776
FolderRequest.MetaInherited Property 776
FolderRequest.MetaInheritedFrom Property 777
FolderRequest.ParentId Property 777
FolderRequest.PublishActive Property 777
FolderRequest.SiteMapPath Property 777
FolderRequest.SiteMapPathInherit Property 778
FolderRequest.StyleSheet Property 778
FolderRequest.SubscriptionProperties Property 778
FolderRequest.suppressNotification Property 779
FolderRequest.TaxonomyIdList Property 779
FolderRequest.TaxonomyInherited Property 779
FolderRequest.TaxonomyInheritedFrom Property 780
FolderRequest.TemplateFileName Property 780
FolderRequest.XmlConfiguration Property 780
FolderRequest.XmlInherited Property 781

FormData Class 781
FormData Properties 781
FormData.Autofill Property 782
FormData.Description Property 782
FormData.MailCc Property 783
FormData.MailFrom Property 783
FormData.MailPreamble Property 783
FormData.MailSubject Property 783
FormData.MailTo Property 784
FormData.PollRevision Property 784
FormData.SendXmlPacket Property 784
FormData.StoreDataTo Property 785
FormData.Validation Property 785

FormFieldDataItem Class 785
FormFieldDataItem Properties 786
FormFieldDataItem.DataValue Property 786
FormFieldDataItem.FieldName Property 786

FormFieldDatalist Class 786
FormFieldDatalist Properties 787
FormFieldDatalist.CaptionXPath Property 787
FormFieldDatalist.DatalistItems Property 788
FormFieldDatalist.DataNamespaces Property 788

FormFieldDatalist.DataSelect Property 788
FormFieldDatalist.DataSrc Property 788
FormFieldDatalist.Language Property 788
FormFieldDatalist.Name Property 789
FormFieldDatalist.ValueXPath Property 789

FormFieldDatalistItem Class 789
FormFieldDatalistItem Properties 790
FormFieldDatalistItem.DataValue Property 790
FormFieldDatalistItem.DisplayValue Property 790
FormFieldDatalistItem.IsDefault Property 791
FormFieldDatalistItem.IsEnabled Property 791

FormFieldDefinition Class 791
FormFieldDefinition Properties 792
FormFieldDefinition.Basetype Property 792
FormFieldDefinition.ContentType Property 792
FormFieldDefinition.DatalistName Property 793
FormFieldDefinition.Datatype Property 793
FormFieldDefinition.DisplayName Property 793
FormFieldDefinition.FieldName Property 794
FormFieldDefinition.Language Property 794
FormFieldDefinition.XPath Property 794

FormFieldList Class 794
FormFieldList Properties 795
FormFieldList.Datalists Property 795
FormFieldList.Fields Property 795

FormRequestData Class 796
FormRequestData Properties 796
FormRequestData.AutoFill Property 797
FormRequestData.FieldList Property 797
FormRequestData.Id Property 797
FormRequestData.IsPoll Property 797
FormRequestData.Language Property 798
FormRequestData.PostBackData Property 798
FormRequestData.PreFillFielddata Property 798
FormRequestData.TagRequired Property 799

FormSubmittedData Class 799
FormSubmittedData Properties 799

FormSubmittedData.DataItems Property 800
FormSubmittedData.DateSubmitted Property 800
FormSubmittedData.FormDataID Property 800
FormSubmittedData.UserId Property 800
FormSubmittedData.UserInfo Property 801

UserData Class 801
UserData Properties 801
UserData.AcceptedTerms Property 802
UserData.AdUserName Property 802
UserData.Avatar Property 803
UserData.DateCreated Property 803
UserData.DisplayName Property 803
UserData.DisplayUserName Property 803
UserData.Email Property 804
UserData.FirstName Property 804
UserData.GroupId Property 804
UserData.GroupName Property 805
UserData.Id Property 805
UserData.IsMemberShip Property 805
UserData.LanguageId Property 805
UserData.LanguageName Property 806
UserData.LastLoginDate Property 806
UserData.LastName Property 806
UserData.LoginAttempts Property 807
UserData.Password Property 807
UserData.Rank Property 807
UserData.Signature Property 807
UserData.Username Property 808
UserData.UserPreference Property 808

Index a

Welcome

Welcome to Ektron CMS400.NET API (see page 363) Online Help. This documentation helps you understand the Server Control API (see page 7), Plug-in Extension API (see page 325), .NET Assemblies (see page 362) and the Web Services API (see page 575). 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 325) 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 362) 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 575) exposes many of the same methods as the .NET Assemblies for use with SOAP over HTTP.

Copyright (c) Ektron Inc. 2007

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 456)

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 459)

Content

create a Web page that allows users to add content? (see page 470)

create a Web page that allows users to delete content by entering the content's ID? (see page 475)

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 479)

ContentRating

create a Web page that retrieves a content rating element used to rate content? (see page 488)

Folder

create a Web page from which a folder can be added to the CMS400.NET? (see page 385)

create a Web page from which a folder can be deleted from CMS400.NET by entering its ID? (see page 386)

create a Web page from which a folder ID can be entered and list of child folders returned? (see page 387)

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 397)

Metadata

create a Web page from which you can add a new metadata definition? (see page 414)

create a Web page from which all metadata for a specific content ID can be retrieved? (see page 416)

Copyright (c) Ektron Inc. 2007

SiteMap

create a Web page from which the breadcrumb path for a given folder can be retrieved? (see page 431)

create a Web page that displays the site map for a given folder? (see page 432)

Threaded Discussions

create a Web page from which a forum can be added to a discussion board? (see page 505)

create a Web page from which forum data can be retrieved? (see page 511)

Users and Membership Users

create a Web page from which a new membership user can be added? (see page 545)

create a Web page that retrieves user details for a given username? (see page 556)

create a Web page that programmatically logs a user into CMS400.NET? (see page 568)

Copyright (c) Ektron Inc. 2007

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

Copyright (c) Ektron Inc. 2007

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 (see page 363) 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)	

Ektron.Cms.Controls Namespace

Namespaces

Name	Description
Personalization (see page 10)	This is namespace Ektron.Cms.Controls.Personalization.

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 17)	Use the AnalyticsTracker server control to track statistics about visits to your Web site.
	AssetControl (see page 19)	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 22)	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 28)	The BlogArchive server control displays a list of past months or years that have posts.
	BlogCalendar (see page 30)	The BlogCalendar server control allows a user (see page 534) to display a calendar (see page 447) on a Web page and associate it with a blog.
	BlogCategories (see page 33)	The BlogCategories server control displays a blog's categories on a Web form.
	BlogEntries (see page 36)	The BlogEntries server control allows you to display a list of multiple blog posts on a Web form.
	BlogPost (see page 41)	The BlogPost server control allows you to display an individual blog post on a page.
	BlogRecentPosts (see page 43)	The BlogRecentPosts server control displays a list of recent blog posts links on a Web form.
	BlogRoll (see page 46)	The BlogRoll server control displays the blog roll for a blog on a Web form.
	BlogRSS (see page 49)	The BlogRSS server control displays the icon for the blog's RSS feed on the Web form.
	BreadCrumb (see page 51)	The BreadCrumb server control displays history breadcrumbs. Your site visitors can use the breadcrumb trail to jump to previously visited pages.
	BreadCrumbItem (see page 57)	Class to contain compass item data.
	BusinessRules (see page 61)	Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example: • **Condition** - a site visitor visits your site for the first time • **Action** - add a cookie to the site visitor's system and pop up a welcome message With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 61)
	Calendar (see page 64)	Summary description for Calendar.
	Collection (see page 68)	Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	ContentBlock (see page 78)	Summary description for ContentBlock.
	ContentList (see page 85)	The ContentList server control is used to display a list of content (see page 449) blocks on a Web page. In contrast to a List Summary, where content (see page 449) must be in a specified folder, the ContentList server control uses content (see page 449) from any CMS400.NET folder.
	ContentRating (see page 94)	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 (see page 449) item. A visitor can rank the content (see page 449) on a numerical scale of one through nine, as well as submit a text comment.
	ContentReview (see page 102)	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 (see page 449) 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 (see page 449) item or reviews by... more (see page 102)
	DesignTimeDiagnostic (see page 107)	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 229) Control** - displays the Login (see page 229) 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 110)	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 118)	The Directory server control lets you customize the behavior of the Taxonomy (see page 498) feature. You place this control on any Web form to display a taxonomy.
	EktronCatalogPart (see page 129)	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 135)	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 137)	Similar to a Microsoft WebPartZone, the EktronWebPartZone server control provides a location for content (see page 449) 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 138)	The base class from which all Ektron CMS400 Server Controls are derived.
	EkWebPart (see page 144)	This is class Ektron.Cms.Controls.EkWebPart.
	EkXsltBase (see page 145)	Summary description for EkXsltBase.
	FlexMenu (see page 147)	The FlexMenu server control displays a menu on a Web form. Although it looks and acts like a Smart Menu (see page 246), 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 153)	The FolderBreadcrumb server control displays sitemap breadcrumbs. This type of trail is typically made up of the folder path for the content (see page 449). 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 449)" > "Managing Content (see page 449) Folders" > "Breadcrumbs" For additional information on this server control, see the Developer Manual section "Introduction... more (see page 153)
	FormBlock (see page 156)	The FormBlock server control displays a content (see page 449) block associated with a form.
	FormFieldItem (see page 164)	Defines a field within the form.
	Forum (see page 167)	The Forum server control displays a discussion board on a Web page.
	HtmlEditDisplay (see page 172)	This is class Ektron.Cms.Controls.HtmlEditDisplay.
	HtmlEditor (see page 175)	This server control provides the ability to edit specific CMS content (see page 449). The eWebEditPro editor can easily be placed onto a page for content (see page 449) editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor. The control, at run time, produces the fields listed below. They are posted with the content (see page 449) and can be processed in the receiving page. • [ControlName]_Action = the action to take - save, check in, undo checkout • [ControlName]_ContentTitle = the content (see page 449) title • [ControlName]_ContentId = the ID of the content (see page 449) • [ControlName]_Language = the language of the content (see page 449) The content (see page 449) title caption... more (see page 175)
	ImageControl (see page 191)	The ImageControl server control, when viewed on a Web form, displays an image stored within the CMS400.NET document management feature. When a user (see page 534) 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 (see page 534) can use to update the file.
	IndexSearch (see page 194)	Use the IndexSearch server control to display a Smart Form (see page 493)'s search (see page 514) screen on any Web form. This lets a user (see page 534) search (see page 514) your site based on XML parameters.
	LanguageAPI (see page 211)	Control that can be used to override the default system language assignment behavior.
	LanguageSelect (see page 215)	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 142) settings screen.
	ListSummary (see page 217)	Summary description for ListSummary.
	Login (see page 229)	Summary description for Login.
	Map (see page 232)	Map control
	Membership (see page 240)	The Membership server control creates a form on a Web page that allows a user (see page 534) to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user (see page 534) properties and any custom user (see page 534) 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 246)	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 248) property.

	MetaData (see page 249)	The Metadata server control lets you add the metadata from content (see page 449) 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 (see page 449) block, multiple content (see page 449) blocks, or by dynamically passing a content (see page 449) ID from a URL.
	MetaDataList (see page 252)	Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	PersonalizationManager (see page 262)	The PersonalizationManager server control allows site users to personalize, add, edit and hide content (see page 449) on a Web page. The level of privileges assigned to a site user (see page 534) in CMS400.NET determines what personalization functionality is available when the control is active. The Personalization (see page 10) manager only appears when a user (see page 534) is logged in to the site.
	Poll (see page 268)	The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page.
	PostHistory (see page 270)	The PostHistory server control displays a list of posts for a given user (see page 534). 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 449) - the content (see page 449) with in the post
	RssAggregator (see page 273)	Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces.
	RssItemBase (see page 278)	Container for Title (see page 279), Link (see page 279), Description (see page 279) and DatePosted (see page 278) article information.
	Search (see page 279)	Search object returns the control and results.
	Sitemap (see page 301)	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 305)	Class extends the Menu (see page 246)-Sever-Control, adding CSS based menus.
	WebSearch (see page 311)	The Web Search (see page 279) server control lets you customize the behavior of the search (see page 514). You place this control on any Web form from which a site visitor can search (see page 514) your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 424)" > "Web Site (see page 424) Search (see page 279)."

Legend

	Class
	abstract

Ektron.Cms.Controls.Personalization Namespace

This is namespace Ektron.Cms.Controls.Personalization.

Classes

	Name	Description
	EkEditorPart (see page 10)	This is class Ektron.Cms.Controls.Personalization.EkEditorPart.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	EkEditorPart (see page 10)	This is class Ektron.Cms.Controls.Personalization.EkEditorPart.

Legend

	Class

EkEditorPart Class

Class Hierarchy

C#

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

Visual Basic

```
Friend Class EkEditorPart
Inherits EditorPart
```

File

EkEditorPart.cs

Description

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

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	ActiveTopics (see page 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 17)	Use the AnalyticsTracker server control to track statistics about visits to your Web site.
	AssetControl (see page 19)	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 22)	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 28)	The BlogArchive server control displays a list of past months or years that have posts.
	BlogCalendar (see page 30)	The BlogCalendar server control allows a user (see page 534) to display a calendar (see page 447) on a Web page and associate it with a blog.
	BlogCategories (see page 33)	The BlogCategories server control displays a blog's categories on a Web form.
	BlogEntries (see page 36)	The BlogEntries server control allows you to display a list of multiple blog posts on a Web form.
	BlogPost (see page 41)	The BlogPost server control allows you to display an individual blog post on a page.
	BlogRecentPosts (see page 43)	The BlogRecentPosts server control displays a list of recent blog posts links on a Web form.
	BlogRoll (see page 46)	The BlogRoll server control displays the blog roll for a blog on a Web form.
	BlogRSS (see page 49)	The BlogRSS server control displays the icon for the blog's RSS feed on the Web form.
	BreadCrumb (see page 51)	The BreadCrumb server control displays history breadcrumbs. Your site visitors can use the breadcrumb trail to jump to previously visited pages.
	BreadCrumbItem (see page 57)	Class to contain compass item data.
	BusinessRules (see page 61)	Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example: • **Condition** - a site visitor visits your site for the first time • **Action** - add a cookie to the site visitor's system and pop up a welcome message With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 61)
	Calendar (see page 64)	Summary description for Calendar.
	Collection (see page 68)	Ektron Collection class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	ContentBlock (see page 78)	Summary description for ContentBlock.

	ContentList (see page 85)	The ContentList server control is used to display a list of content (see page 449) blocks on a Web page. In contrast to a List Summary, where content (see page 449) must be in a specified folder, the ContentList server control uses content (see page 449) from any CMS400.NET folder.
	ContentRating (see page 94)	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 (see page 449) item. A visitor can rank the content (see page 449) on a numerical scale of one through nine, as well as submit a text comment.
	ContentReview (see page 102)	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 (see page 449) 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 (see page 449) item or reviews by... more (see page 102)
	DesignTimeDiagnostic (see page 107)	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 229) Control** - displays the Login (see page 229) 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 110)	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 118)	The Directory server control lets you customize the behavior of the Taxonomy (see page 498) feature. You place this control on any Web form to display a taxonomy.
	EktronCatalogPart (see page 129)	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 135)	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 137)	Similar to a Microsoft WebPartZone, the EktronWebPartZone server control provides a location for content (see page 449) 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 138)	The base class from which all Ektron CMS400 Server Controls (see page 7) are derived.
	EkWebPart (see page 144)	This is class Ektron.Cms.Controls.EkWebPart.
	EkXsltBase (see page 145)	Summary description for EkXsltBase.
	FlexMenu (see page 147)	The FlexMenu server control displays a menu on a Web form. Although it looks and acts like a Smart Menu (see page 246), 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 153)	The FolderBreadcrumb server control displays sitemap breadcrumbs. This type of trail is typically made up of the folder path for the content (see page 449). 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 449)" > "Managing Content (see page 449) Folders" > "Breadcrumbs" For additional information on this server control, see the Developer Manual section "Introduction... more (see page 153)
	FormBlock (see page 156)	The FormBlock server control displays a content (see page 449) block associated with a form.
	FormFieldItem (see page 164)	Defines a field within the form.
	Forum (see page 167)	The Forum server control displays a discussion board on a Web page.
	HtmlEditDisplay (see page 172)	This is class Ektron.Cms.Controls.HtmlEditDisplay.

	HtmlEditor (see page 175)	This server control provides the ability to edit specific CMS content (see page 449). The eWebEditPro editor can easily be placed onto a page for content (see page 449) editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor. The control, at run time, produces the fields listed below. They are posted with the content (see page 449) and can be processed in the receiving page. • [ControlName]_Action = the action to take - save, check in, undo checkout • [ControlName]_ContentTitle = the content (see page 449) title • [ControlName]_ContentId = the ID of the content (see page 449) • [ControlName]_Language = the language of the content (see page 449) The content (see page 449) title caption... more (see page 175)
	ImageControl (see page 191)	The ImageControl server control, when viewed on a Web form, displays an image stored within the CMS400.NET document management feature. When a user (see page 534) 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 (see page 534) can use to update the file.
	IndexSearch (see page 194)	Use the IndexSearch server control to display a Smart Form (see page 493)'s search (see page 514) screen on any Web form. This lets a user (see page 534) search (see page 514) your site based on XML parameters.
	LanguageAPI (see page 211)	Control that can be used to override the default system language assignment behavior.
	LanguageSelect (see page 215)	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 142) settings screen.
	ListSummary (see page 217)	Summary description for ListSummary.
	Login (see page 229)	Summary description for Login.
	Map (see page 232)	Map control
	Membership (see page 240)	The Membership server control creates a form on a Web page that allows a user (see page 534) to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user (see page 534) properties and any custom user (see page 534) 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 246)	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 248) property.
	MetaData (see page 249)	The Metadata server control lets you add the metadata from content (see page 449) 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 (see page 449) block, multiple content (see page 449) blocks, or by dynamically passing a content (see page 449) ID from a URL.
	MetaDataList (see page 252)	Ektron ContentByMeta class. Allows access to a ContentBlockItem object through IEnumerator, ICollection and IList interfaces.
	PersonalizationManager (see page 262)	The PersonalizationManager server control allows site users to personalize, add, edit and hide content (see page 449) on a Web page. The level of privileges assigned to a site user (see page 534) in CMS400.NET determines what personalization functionality is available when the control is active. The Personalization (see page 10) manager only appears when a user (see page 534) is logged in to the site.
	Poll (see page 268)	The Poll Server Control displays a poll or survey created from an Ektron CMS400.NET form on a Web page.
	PostHistory (see page 270)	The PostHistory server control displays a list of posts for a given user (see page 534). 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 449) - the content (see page 449) with in the post
	RssAggregator (see page 273)	Ektron RssAggregator class. Allows access to a RSS Feed object through IEnumerator, ICollection and IList interfaces.
	RssItemBase (see page 278)	Container for Title (see page 279), Link (see page 279), Description (see page 279) and DatePosted (see page 278) article information.
	Search (see page 279)	Search object returns the control and results.
	Sitemap (see page 301)	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 305)	Class extends the Menu (see page 246)-Sever-Control, adding CSS based menus.
	WebSearch (see page 311)	The Web Search (see page 279) server control lets you customize the behavior of the search (see page 514). You place this control on any Web form from which a site visitor can search (see page 514) your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 424)" > "Web Site (see page 424) Search (see page 279)."

Legend

	Class
	abstract

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

System.Web.UI.WebControls.WebControl → Ektron.Cms.Controls.EkWebControl → Ektron.Cms.Controls.ActiveTopics

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 14)	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.

Legend

	Method

ActiveTopics.Fill Method

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

C#

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

Visual Basic

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

ActiveTopics.ToString Method

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

C#

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

Visual Basic

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

ActiveTopics.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

ActiveTopics Properties

The properties of the ActiveTopics class are listed here.

Public Properties

	Name	Description
	BoardID (see page 15)	The ID of the board from which to get entries.
V	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.
	MaxNumber (see page 16)	The maximum number of topics listed. The default is 10.
	URLPath (see page 16)	The URL path to the Forum (see page 167) server control's Web page. For example:

Legend

	Property
V	virtual

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 int 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 Integer
```

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("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
```

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.MaxNumber Property

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

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(5)]
[Description("The maximum number of topics to return.")]
public int MaxNumber;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(5)>
<Description("The maximum number of topics to return.")>
Public Property MaxNumber() As Integer
```

ActiveTopics.URLPath Property

The URL path to the Forum (see page 167) server control's Web page. For example:

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

or

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

If your Forum (see page 167) 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 18)	

Legend

	Method

AnalyticsTracker.Wrap Method

C#

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

Visual Basic

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

Parameters

Parameters	Description
Html	

AnalyticsTracker Properties

The properties of the AnalyticsTracker class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 18)	The Content (see page 449) ID to flag as "visited" in the event that no other ID is specified by QueryString (e.g. "?id=2"). Set to zero (0) if you do not wish to a Default Content (see page 449) ID.
	DynamicParameter (see page 18)	Gets or Sets the QueryString parameter to read a Content (see page 449) ID dynamically. Set to "" (empty) to always display the default ContentID (static.)
	EnableAnalytics (see page 19)	Determines if the Analytics (see page 442) feature is enabled for this page and (if identified) the content (see page 449) item. Choices are: • **ConfigSpecified** - Take the value from the web.config file's enableAnalytics property. • **True** - Enable Analytic tracking • **False** - Disable Analytic tracking

Legend

	Property

AnalyticsTracker.DefaultContentID Property

The Content (see page 449) ID to flag as "visited" in the event that no other ID is specified by QueryString (e.g. "?id=2"). Set to zero (0) if you do not wish to a Default Content (see page 449) ID.

C#

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

Visual Basic

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

AnalyticsTracker.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 449) 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 442) feature is enabled for this page and (if identified) the content (see page 449) 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 20)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

AssetControl.Fill Method

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

C#

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

Visual Basic

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

AssetControl Properties

The properties of the AssetControl class are listed here.

Public Properties

	Name	Description
	DefaultAssetID (see page 20)	The ID of the asset you want to update. This property is used when the UploadType (see page 21) property is set to Update.
	DefaultFolderID (see page 21)	The ID of the folder where assets are added. This property is used when the UploadType (see page 21) property is set to Add. Root is 0.
	DynamicParameter (see page 21)	Gets or Sets the QueryString parameter to read a Content (see page 449) ID or Folder (see page 383) ID dynamically. Content (see page 449) ID will be read if mode is Update. Folder (see page 383) ID will be read if mode is Add. Set to String.Empty if you wish to always display the default DocumentID or FolderID (static.)
	UploadType (see page 21)	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 21) property. Select Update to update assets.

Legend

	Property

AssetControl.DefaultAssetID Property

The ID of the asset you want to update. This property is used when the UploadType (see page 21) 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.")]
public int 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.")>
Public Property DefaultAssetID() As Integer
```

AssetControl.DefaultFolderID Property

The ID of the folder where assets are added. This property is used when the UploadType (see page 21) 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))]
public int 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))>
Public Property DefaultFolderID() As Integer
```

AssetControl.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 449) ID or Folder (see page 383) ID dynamically. Content (see page 449) ID will be read if mode is Update. Folder (see page 383) ID will be read if mode is Add. Set to String.Empty if you wish to always display the default DocumentID or FolderID (static.)

C#

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

Visual Basic

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

AssetControl.UploadType Property

Selects whether the control will add a document to a folder or overwrite an existing document. Select Add to add assets and use the DefaultFolderID (see page 21) 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.")]
public UploadTypeEnum UploadType;
```

Visual Basic

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

Blog Class

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 22)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 23)	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 23)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Blog.Fill Method

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

C#

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

Visual Basic

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

Blog.ToString Method

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

C#

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

Visual Basic

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

Blog.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

Blog Properties

The properties of the Blog class are listed here.

Public Properties

	Name	Description
	ArchiveMode (see page 24)	Select whether the archive appears in month format or year format. The default is month.
	BlogID (see page 24)	The ID of the blog in CMS400.NET.
	CacheInterval (see page 24)	Sets the cache interval in seconds.
	DateToStart (see page 25)	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.
	DynamicParameter (see page 25)	Gets or Sets the QueryString parameter to read a Blog (see page 22) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 22) (static.)
	EditorHeight (see page 25)	Sets the height of the blog editor in pixels.
	EditorWidth (see page 26)	Sets the width of the blog editor in pixels.
	JavascriptEditorHTMLMode (see page 26)	Set to False to not allow users to edit their posts and replies in HTML. The default is True.
	MaxResults (see page 26)	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, the all post for the month are shown. • If set to -3, the control uses the # of Post Visible setting in the Workarea.
	PostParameter (see page 27)	Checks the QueryString for this value and shows the post when specified. Leave blank to always display the blog.
	RecentPosts (see page 27)	Sets the number of recent posts to display. Integer value, must be greater than 0 (Zero).
	ShowHeader (see page 27)	Controls (see page 7) the display of the title and tagline.
	ShowRSS (see page 27)	Displays the icon for the RSS feed when set to True.

Legend

	Property

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 in CMS400.NET.

C#

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

Visual Basic

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

Blog.CacheInterval Property

Sets the cache interval in seconds.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The interval to cache this control in seconds.")]
[DefaultValue(0)]
public double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The interval to cache this control in seconds.")>
```

```
<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 to get posts from.")]
public DateTime DateToStart;
```

Visual Basic

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

Blog.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 22) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 22) (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.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 False to not allow users to edit their posts and replies in HTML. The default is True.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Determines if HTML mode is allowed for the Javascript Editor.")]
public bool JavascriptEditorHTMLMode;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Determines if HTML mode is allowed for the Javascript Editor.")>
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, the 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 for a given day. -3 will use the workarea value.")]
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 for a given day. -3 will use the workarea value.")>
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

Controls (see page 7) the display of the title and tagline.

C#

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

Visual Basic

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

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 28)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 29)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogArchive.Fill Method

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

C#

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

Visual Basic

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

BlogArchive.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

BlogArchive Properties

The properties of the BlogArchive class are listed here.

Public Properties

	Name	Description
	ArchiveMode (see page 29)	Select whether the archive appears in month format or year format. The default is month.
	BlogID (see page 29)	The ID of the blog in CMS400.NET. Can be selected via the CMS Explorer in design time.
V	CacheInterval (see page 30)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a 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 30)	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 30)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
V	virtual

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 int 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 Integer
```

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

Visual Basic

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

BlogArchive.Text Property

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

C#

```
public override string Text;
```

Visual Basic

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

BlogArchive.WrapTag Property

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

C#

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

Visual Basic

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

BlogCalendar Class

The BlogCalendar server control allows a user (see page 534) to display a calendar (see page 447) 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 31)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 31)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogCalendar.Fill Method

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

C#

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

Visual Basic

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

BlogCalendar.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar.

BlogCalendar Properties

The properties of the BlogCalendar class are listed here.

Public Properties

	Name	Description
	BlogID (see page 32)	The ID of the blog you wish to display a calendar (see page 447) for. Can be selected via the CMS Explorer in design time.
V	CacheInterval (see page 32)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a 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 32)	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 33)	Allows a developer to specify control's tag. Default is assumed " ".

Legend

	Property
V	virtual

BlogCalendar.BlogID Property

The ID of the blog you wish to display a calendar (see page 447) for. Can be selected via the CMS Explorer in design time.

C#

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

Visual Basic

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

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 34)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 34)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogCategories.Fill Method

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

C#

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

Visual Basic

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

BlogCategories.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

BlogCategories Properties

The properties of the BlogCategories class are listed here.

Public Properties

	Name	Description
	BlogID (see page 34)	The ID of the blog in CMS400.NET. Can be selected via the CMS Explorer in design time.
V	CacheInterval (see page 35)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a 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 35)	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 35)	Allows a developer to specify control's tag. Default is assumed " ".

Legend

	Property
V	virtual

BlogCategories.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 you wish to list categories from.")]
[Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int BlogID;
```

Visual Basic

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

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 36)	Method that can be overridden and used to grab data from the back end for each server control.
	GetRssFeed (see page 37)	This method returns a blog post RSS feed.
	GetView (see page 37)	Gets the view data using dataset
	GetViewNames (see page 37)	gets the view names
	ToString (see page 37)	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 37)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogEntries.Fill Method

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

C#

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

Visual Basic

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

BlogEntries.GetRssFeed Method

This method returns a blog post RSS feed.

C#

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

Visual Basic

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

Returns

String

BlogEntries.GetView Method

Gets the view data using dataset

C#

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

Visual Basic

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

Parameters

Parameters	Description
viewName	

Returns

DataSourceView

BlogEntries.GetViewNames Method

gets the view names

C#

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

Visual Basic

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

Returns

View names

BlogEntries.ToString Method

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

C#

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

Visual Basic

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

BlogEntries.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

BlogEntries Properties

The properties of the BlogEntries class are listed here.

Public Properties

	Name	Description
	BlogID (see page 38)	The ID of the blog in CMS400.NET.
	BlogPostParameter (see page 39)	Checks the QueryString for this value and replaces the blog entries list with the blog post when specified. Leave blank to always display the blog entries.
V	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).
	DateToStart (see page 39)	The date of the last blog entries you want to appear.
	DisplayXslt (see page 40)	The Xslt used to render the blog entry list (optional).
	MaxResults (see page 40)	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, the all post for the month are shown. • If set to -3, the control uses the # of Post Visible setting in the Workarea.
	ShowHeader (see page 40)	Controls (see page 7) the display of the title and tagline.
	WrapTag (see page 41)	Allows a developer to specify control's tag. Default is assumed " ".
	XmlDoc (see page 41)	XmlDocument used in this control.

Legend

	Property
V	virtual

BlogEntries.BlogID Property

The ID of the blog in CMS400.NET.

C#

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

Visual Basic

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

```
<DefaultValue(0)>
<Description("The ID of the blog from which to select entries.")>
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BlogID() As Integer
```

BlogEntries.BlogPostParameter Property

Checks the QueryString for this value and replaces the blog entries list with the blog post when specified. Leave blank to always display the blog entries.

C#

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

Visual Basic

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

BlogEntries.CacheInterval Property

Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a 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
```

BlogEntries.DateToStart Property

The date of the last blog entries you want to appear.

C#

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

Visual Basic

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

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.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, the 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 results to fetch. If set to zero, there is no limit. If
set to -1, all posts for the day are shown. If set to -2, the all post for the month are
shown. -3 will use the workarea value.")]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(-3)>
<Description("The maximum number of results to fetch. If set to zero, there is no limit. If
set to -1, all posts for the day are shown. If set to -2, the all post for the month are
shown. -3 will use the workarea value.")>
Public Property MaxResults() As Integer
```

BlogEntries.ShowHeader Property

Controls (see page 7) the display of the title and tagline.

C#

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

```
[Description("Controls the display of the title and tagline.")]
public bool ShowHeader;
```

Visual Basic

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

BlogEntries.WrapTag Property

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

C#

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

Visual Basic

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

BlogEntries.XmlDoc Property

XmlDocument used in this control.

C#

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

Visual Basic

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

BlogPost Class

The BlogPost server control 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 42)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogPost.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

BlogPost Properties

The properties of the BlogPost class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 42)	Gets or Sets the Default Content (see page 449) ID for this Control.
	ShowType (see page 43)	Shows a blogpost's content (see page 449) and its comments or just the blogpost's comments. The default is Content (see page 449). • **Content (see page 449)** - Show a blogpost's content (see page 449) and its comments. • **Description** - Show a blogpost's comments only.

Legend

	Property

BlogPost.DefaultContentID Property

Gets or Sets the Default Content (see page 449) ID for this Control.

C#

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

```
[Category("Ektron")]
[Editor(typeof(BlogPostUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default Content ID for this Control.")]
public new int DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(BlogPostUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default Content ID for this Control.")>
Public new Property DefaultContentID() As Integer
```

BlogPost.ShowType Property

Shows a blogpost's content (see page 449) and its comments or just the blogpost's comments. The default is Content (see page 449).

- **Content (see page 449)** - Show a blogpost's content (see page 449) and its comments.
- **Description** - Show a blogpost's comments only.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("Content")]
[Localizable(true)]
public eShowType ShowType;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("Content")>
<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 44)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 44)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogRecentPosts.Fill Method

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

C#

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

Visual Basic

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

BlogRecentPosts.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

BlogRecentPosts Properties

The properties of the BlogRecentPosts class are listed here.

Public Properties

	Name	Description
	BlogID (see page 45)	The ID of the blog from which to select recent posts. Can be selected via the CMS Explorer in design time.
	CacheInterval (see page 45)	Sets the amount of time the server control's data is cached. The default 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 45)	Sets the number of recent posts to display. Integer value, must be greater than 0 (zero).
	Text (see page 46)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.
	WrapTag (see page 46)	Allows a developer to specify control's tag. Default is assumed " ".

Legend

	Property
	virtual

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 int 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 Integer
```

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 47)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 47)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BlogRoll.Fill Method

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

C#

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

Visual Basic

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

BlogRoll.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

BlogRoll Properties

The properties of the BlogRoll class are listed here.

Public Properties

	Name	Description
	BlogID (see page 47)	The ID of the blog in CMS400.NET. Can be selected via the CMS Explorer in design time.
	CacheInterval (see page 48)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a 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 48)	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 48)	Allows a developer to specify control's tag. Default is assumed " ".

Legend

	Property
	virtual

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 int 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 Integer
```

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

System.Web.UI.WebControls.WebControl → Ektron.Cms.Controls.EkWebControl → Ektron.Cms.Controls.BlogRSS

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

Legend

	Method

BlogRSS.Fill Method

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

C#

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

Visual Basic

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

BlogRSS Properties

The properties of the BlogRSS class are listed here.

Public Properties

	Name	Description
	BlogID (see page 50)	The ID of the blog to be output as an RSS feed.
	CacheInterval (see page 50)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a control's data is cached. For example, if you want to cache the data for five minutes, you set the CacheInterval property to 300 (three hundred).
	Text (see page 50)	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 " ".

Legend

	Property
	virtual

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 int 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 Integer
```

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 52)	Method that can be overridden and used to grab data from the back end for each server control.
	GenerateTrail_HTML (see page 52)	Generates Trail HTML for both vertical and horizontal styles This demonstrates usage of a HtmlTextWriter.
	ResetTrail (see page 52)	This method is overloaded. The ResetTrail method: • Resets and removes all current Trail items • Resets the Trail with an array list of new Trail items • Resets the Trail to the given Trail item

Legend

	Method

BreadCrumb.Fill Method

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

C#

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

Visual Basic

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

BreadCrumb.GenerateTrail_HTML Method

Generates Trail HTML for both vertical and horizontal styles

This demonstrates usage of 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 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 52)	Resets and removes all current Trail items.
	BreadCrumb.ResetTrail (ArrayList) (see page 53)	Resets the trail with an array list of new trail items.
	BreadCrumb.ResetTrail (BreadCrumbItem) (see page 53)	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 54)	If desired, enter one or more symbols or characters that identify the current page in the breadcrumb trail.
	DefaultContentID (see page 54)	Gets or sets the default content (see page 449) ID for this control.
	DisplayStyle (see page 54)	Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal.
	DisplayTitle (see page 55)	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 55)	Gets or Sets the QueryString parameter to read a content (see page 449) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	IconAlt (see page 55)	If you define an image in the IconPath (see page 55) property, enter any “Alt” text that should appear when a site visitor hovers the cursor over that image.
	IconPath (see page 55)	If you want the breadcrumb trail to display an image to identify this Web form, enter the path to the image. **Important:** The image location must be relative to the Web root.
	LinkLastItem (see page 56)	Determine whether the last breadcrumb item appears as a hyperlink on this Web form.
	LinkTarget (see page 56)	Determines the type of window that appears on this Web form when a user (see page 534) clicks an item in the breadcrumb trail. The default is _self. • **_Self** = opens in same window • **_Top** = opens in parent window • **_Blank** = opens in new window • **_Parent** = opens in the parent frame
	MaxItems (see page 56)	The maximum number of items in the breadcrumb trail on this Web form. (0 = Unlimited)

	MetadataName (see page 56)	Gets or sets the metadata of the content (see page 449) block for BreadCrumb (see page 51) display title. If not specified it will use content (see page 449) block title.
	Mode (see page 57)	Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal. • **Normal** = breadcrumb trail is hyperlinked • **DisplayOnly** = breadcrumb trail is plain text
	separator (see page 57)	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 (>).

Legend

	Property

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 (see page 449) 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 int 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 Integer
```

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 string DisplayTitle;
```

Visual Basic

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

BreadCrumb.DynamicParameter Property

Gets or Sets the QueryString parameter to read a content (see page 449) 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 55) 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 (see page 534) 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 (see page 449) block for BreadCrumb (see page 51) display title. If not specified it will use content (see page 449) block title.

C#

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

Visual Basic

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

BreadCrumb.Mode Property

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

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

C#

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

Visual Basic

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

BreadCrumb.separator Property

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

C#

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

Visual Basic

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

BreadCrumbItem Class

Class to contain compass item data.

Class Hierarchy

Ektron.Cms.Controls.BreadCrumbItem

C#

```
public class BreadCrumbItem;
```

Visual Basic

```
Public Class BreadCrumbItem
```

File

BreadCrumbItem.cs

BreadCrumbItem Constructor

Overload List

	Name	Description
	BreadCrumbItem.BreadCrumbItem () (see page 58)	Default no parameter constructor. This is overloaded. See Also: BreadCrumbItem.BreadCrumbItem Constructor (string, string, string, string, string) (see page 58).
	BreadCrumbItem.BreadCrumbItem (string, string, string, string, string) (see page 58)	All parameter constructor. This is overloaded. See Also: BreadCrumbItem.BreadCrumbItem Constructor () (see page 58)

BreadCrumbItem.BreadCrumbItem Constructor ()

Default no parameter constructor.

This is overloaded. See Also: BreadCrumbItem.BreadCrumbItem Constructor (string, string, string, string, string) (see page 58).

C#

```
public BreadCrumbItem();
```

Visual Basic

```
Public Sub BreadCrumbItem()
```

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

All parameter constructor.

This is overloaded. See Also: BreadCrumbItem.BreadCrumbItem Constructor () (see page 58)

C#

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

Visual Basic

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

Parameters

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

BreadCrumbItem Methods

The methods of the BreadCrumbItem class are listed here.

Public Methods

	Name	Description
	Equals (see page 59)	Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 59)() do not match, The ToString (see page 59)() function is also overridden to allow this to work.
	GetHashCode (see page 59)	Overridden otherwise VS.NET gives a warning.
	ToString (see page 59)	Returns a concatenated string of all class variables.ToString().

Legend

	Method

BreadCrumbItem.Equals Method

Overridden equals function returns false on null object comparison. And, when the given object and current object .ToString (see page 59)() do not match, The ToString (see page 59)() function is also overridden to allow this to work.

C#

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

Visual Basic

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

Parameters

Parameters	Description
obj	_nt_

Returns

BreadCrumbItem.GetHashCode Method

Overridden otherwise VS.NET gives a warning.

C#

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

Visual Basic

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

Returns

BreadCrumbItem.ToString Method

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

C#

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

Visual Basic

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

BreadCrumbItem Properties

The properties of the BreadCrumbItem class are listed here.

Public Properties

	Name	Description
S	CaseSensitiveUrlComparison (see page 60)	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 60)	Gets / Sets the title for use with the display in the compass.
	IconAlt (see page 60)	Gets / Sets the text to use as the item alt text.
	IconPath (see page 60)	Gets / Sets the root relative path to the item icon.
	RawUrl (see page 60)	Gets / Sets the URL to use for the item link.
	Target (see page 61)	Gets / Sets the target of the item.

Legend

	Property
S	static

BreadCrumbItem.CaseSensitiveUrlComparison Property

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

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

C#

```
public static bool CaseSensitiveUrlComparison;
```

Visual Basic

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

BreadCrumbItem.DisplayTitle Property

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

C#

```
public string DisplayTitle;
```

Visual Basic

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

BreadCrumbItem.IconAlt Property

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

C#

```
public string IconAlt;
```

Visual Basic

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

BreadCrumbItem.IconPath Property

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

C#

```
public string IconPath;
```

Visual Basic

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

BreadCrumbItem.RawUrl Property

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

C#

```
public string RawUrl;
```

Visual Basic

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

BreadCrumbItem.Target Property

Gets / Sets the target of the item.

C#

```
public string Target;
```

Visual Basic

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

BusinessRules Class

Class Hierarchy

C#

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

Visual Basic

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

File

BusinessRules.cs

Description

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 365) API (see page 363) allows you to work with results of a Business Rule. The BusinessRules server control API (see page 363) 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 365) API (see page 363), click here (see page 365).

BusinessRules Methods

The methods of the BusinessRules class are listed here.

Public Methods

	Name	Description
	Fill (see page 62)	Overrideable method that can be used to grab data from the back end for each server control.
	Wrap (see page 62)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

BusinessRules.Fill Method

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

C#

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

Visual Basic

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

BusinessRules.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

BusinessRules Properties

The properties of the BusinessRules class are listed here.

Public Properties

	Name	Description
	Parameters (see page 62)	User (see page 534) Defined parameters that can be evaluated in rules.
	Result (see page 63)	An array of boolean values representing the results of the evaluations.
	RuleID (see page 63)	The ID number of the Ruleset to evaluate.
	Text (see page 63)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

Legend

	Property

BusinessRules.Parameters Property

User (see page 534) 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 int 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 Integer
```

BusinessRules.Text Property

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

C#

```
public override string Text;
```

Visual Basic

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

Calendar Class

Summary description for Calendar.

Class Hierarchy

C#

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

Visual Basic

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

File

Calendar.cs

Calendar Methods

The methods of the Calendar class are listed here.

Public Methods

	Name	Description
	Fill (see page 64)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 64)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Calendar.Fill Method

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

C#

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

Visual Basic

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

Calendar.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

Calendar Properties

The properties of the Calendar class are listed here.

Public Properties

	Name	Description
	DefaultCalendarID (see page 65)	The default ID of the Calendar (see page 64) to be displayed.
	DisplayType (see page 66)	Specify the the Calendar (see page 64) view. Choices are: • "Month" • "Day" • "Inline" • "MonthUpWithEvent"
	DynamicParameter (see page 66)	The QueryString to check for a dynamic ID.
	EndDate (see page 66)	Specify the end date the Calendar (see page 64) displays.
	HighlightEventID (see page 67)	Specify the event ID which the Calendar (see page 64) will highlight.
	HighlightEventTypeID (see page 67)	Specify the event type and the Calendar (see page 64) highlights that type of event(s).
	ShowXML (see page 67)	Text (see page 68) property will hold raw calendar (see page 447) XML. The Calendar (see page 64) is not visible at runtime.
	StartDate (see page 68)	Specify the start date the calendar (see page 447) displays.
	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.

Legend

	Property

Calendar.DefaultCalendarID Property

The default ID of the Calendar (see page 64) to be displayed.

C#

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

Visual Basic

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

```
<Editor(typeof(CalendarUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("The default ID of the Calendar to be displayed.")>
<Personalizable()>
Public Property DefaultCalendarID() As Integer
```

Calendar.DisplayType Property

Specify the the Calendar (see page 64) 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 64) 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 64) will highlight.

C#

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

Visual Basic

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

Calendar.HighlightEventTypeID Property

Specify the event type and the Calendar (see page 64) highlights that type of event(s).

C#

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

Visual Basic

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

Calendar.ShowXML Property

Text (see page 68) property will hold raw calendar (see page 447) XML. The Calendar (see page 64) is not visible at runtime.

C#

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

```
public bool ShowXML;
```

Visual Basic

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

Calendar.StartDate Property

Specify the start date the calendar (see page 447) displays.

C#

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

Visual Basic

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

Calendar.Text Property

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

C#

```
public override string Text;
```

Visual Basic

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

Collection Class

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

Class Hierarchy

C#

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

Visual Basic

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

File

Collection.cs

Collection Methods

The methods of the Collection class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 69)	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 69)	Calls the database and updates the contents of the collection if the data is out of sync with the specified parameters.
	GetEnumerator (see page 70)	Gets a System.Collections.IEnumerator for the collection.
	GetRssFeed (see page 70)	This method returns a Collection RSS feed.
	GetView (see page 70)	Gets the view data using a dataset.
	GetViewNames (see page 70)	Gets the view names.
	MoveNext (see page 71)	Moves up the index of the Collection. Returns false if the end of the collection has been surpassed.
	Reset (see page 71)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 71)	Index operator into a strongly typed dataset holding some common properties for a Collection item.
	ToString (see page 71)	Wraps the resulting collection with the Ektron toolbars, which will only display when a user (see page 534) is logged in.
	Wrap (see page 71)	Overrideable method that's used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Collection.CopyTo Method

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

C#

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

Visual Basic

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

Collection.Fill Method

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

C#

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

Visual Basic

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

Collection.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

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

Visual Basic

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

Collection.GetRssFeed Method

This method returns a Collection 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. 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 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 (see page 534) 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 72)	Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 449)' could be changed to 'Add Report')
	CacheInterval (see page 73)	Sets the cache interval in second(s).
	CollectionDescription (see page 73)	Returns the description of the given folder ID.
	CollectionTitle (see page 73)	Returns the name of the given folder ID.
	ContentParameter (see page 74)	Checks the QueryString for this value and replaces the collection with that content (see page 449) block when specified. Leave blank to always display the collection.
	Count (see page 74)	Indicates the number of elements in your Collection.
	Current (see page 74)	Gets the current object in the collection.
	DefaultCollectionID (see page 74)	Gets or Sets the Default CollectionID for this Control.
	DynamicParameter (see page 75)	The parameter to query for a Dynamic Collection.
	EkItems (see page 75)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection. Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 75)	Enables paging if the total number of records exceeds the MaxResult value.
	GetHtml (see page 76)	If set to true, retrieves the content (see page 449) (Html body) of every content (see page 449) block in the collection for display.
	IncludeIcons (see page 76)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 76)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 77)	Defines the way a link acts when a link is clicked. Choices are: • **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named. • **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target. • **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - This target makes the link load in... more (see page 77)
	MarkupLanguage (see page 77)	Enter the name of a markup file that manages the display of this server control.
	MaxResults (see page 77)	Maximum number of results to be returned. If set to zero, there is no limit.
	Random (see page 77)	If set to true, grabs a single content (see page 449) block randomly from the collection. Set to false for all data in the collection.
	SuppressWrapperTags (see page 78)	Suppresses the output of span/div tags around the control.
	SyncRoot (see page 78)	Gets an object that's used to synchronize the collection.
	XmlDoc (see page 78)	Returns the Xslt object for the collection.

Legend

	Property
	virtual

Collection.AddText Property

Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 449)' 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()]
public virtual double CacheInterval;
```

Visual Basic

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

Collection.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 (see page 449) block when specified. Leave blank to always display the collection.

C#

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

Visual Basic

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

Collection.Count Property

Indicates the number of elements in your Collection.

C#

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

Visual Basic

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

Collection.Current Property

Gets the current object in the collection.

C#

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

Visual Basic

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

Collection.DefaultCollectionID Property

Gets or Sets the Default CollectionID for this Control.

C#

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

```
public int DefaultCollectionID;
```

Visual Basic

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

Collection.DynamicParameter Property

The parameter to query for a Dynamic Collection.

C#

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

Visual Basic

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

Collection.EkItems Property

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

C#

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

Visual Basic

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

Collection.EnablePaging Property

Enables paging if the total number of records exceeds the MaxResult value.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enables paging if the total number of records exceeds the MaxResult value. Use
this property with non zero value of MaxNumbers.")]
[Personalizable()]
public bool EnablePaging;
```

Visual Basic

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

```
<Description("Enables paging if the total number of records exceeds the MaxResult value. Use
this property with non zero value of MaxNumbers.")>
<Personalizable()>
Public Property EnablePaging() As bool
```

Collection.GetHtml Property

If set to true, retrieves the content (see page 449) (Html body) of every content (see page 449) block in the collection for display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")]
[Personalizable()]
public bool GetHtml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")>
<Personalizable()>
Public Property GetHtml() As bool
```

Collection.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
[Personalizable()]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
<Personalizable()>
Public Property IncludeIcons() As bool
```

Collection.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

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

Visual Basic

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

Collection.LinkTarget Property

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.")]
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
```

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()]
public int MaxResults;
```

Visual Basic

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

Collection.Random Property

If set to true, grabs a single content (see page 449) block randomly from the collection. Set to false for all data in the collection.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, grabs a single content block randomly from the collection. Set
to false for all data in the collection.")]
[Personalizable()]
public bool Random;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, grabs a single content block randomly from the collection. Set
to false for all data in the collection.")>
<Personalizable()>
Public Property Random() As bool
```

Collection.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
```

ContentBlock Class

Summary description for ContentBlock.

Class Hierarchy

C#

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

Visual Basic

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

File

ContentBlock.cs

ContentBlock Methods

The methods of the ContentBlock class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 79)	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 80)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 80)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 80)	Moves up the index of the Collection. Returns false if the end of the collection has been surpassed.
	Reset (see page 80)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 80)	Index operator into a strongly typed dataset holding some common properties for a Collection item.
	ToString (see page 80)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user (see page 534) is logged in.
	Wrap (see page 81)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

ContentBlock.CopyTo Method

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

C#

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

Visual Basic

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

ContentBlock.Fill Method

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

C#

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

Visual Basic

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

ContentBlock.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

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

Visual Basic

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

ContentBlock.MoveNext Method

Moves up the index of the Collection. 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 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 (see page 534) 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 81)	Sets the cache interval in second(s).
	Count (see page 82)	Indicates the number of elements in your Collection.
	Current (see page 82)	Gets the current object in the collection.
	DefaultContentID (see page 82)	Gets or Sets the Default ContentID for this Control.
	DisplayXslt (see page 83)	Gets or Sets the path to an Xslt that can be applied to this content (see page 449) block. Set to String.Empty if you wish to always display the default Content (see page 449) view.
	DynamicParameter (see page 83)	Gets or Sets the QueryString parameter to read a Content (see page 449) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItem (see page 83)	Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 449) Block.
	IsSynchronized (see page 84)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	OverrideXslt (see page 84)	Ektron CMS400.NET allows an Administrator to specify up to three custom Xslt's which can be applied to a content (see page 449) block from the System. This allows the user (see page 534) to override the default ContentBlock (see page 78) Xslt and display one of these custom Xsl transforms, if specified in the system. In addition, you can use the Xslt property to perform a transform on a content (see page 449) block (see Xslt property.)
	StartingFolderID (see page 84)	For use with Web Parts only. Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 85)	Gets an object that can be used to synchronize the collection.

Legend

	Property
	virtual

ContentBlock.CacheInterval Property

Sets the cache interval in second(s).

C#

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

Visual Basic

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

ContentBlock.Count Property

Indicates the number of elements in your Collection.

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()]
public int DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control.")>
<Personalizable()>
Public Property DefaultContentID() As Integer
```

ContentBlock.DisplayXslt Property

Gets or Sets the path to an Xslt that can be applied to this content (see page 449) block. Set to String.Empty if you wish to always display the default Content (see page 449) 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()]
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()>
Public override Property DisplayXslt() As string
```

ContentBlock.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 449) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)

C#

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

Visual Basic

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

ContentBlock.EkItem Property

Gets the Ektron.Cms.Common.ContentBase for the current Content (see page 449) Block.

C#

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

Visual Basic

```
<Browsable(false)>
Public Property EkItem() As Ektron.Cms.Common.ContentBase
```

ContentBlock.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

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

Visual Basic

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

ContentBlock.OverrideXslt Property

Ektron CMS400.NET allows an Administrator to specify up to three custom Xslt's which can be applied to a content (see page 449) block from the System. This allows the user (see page 534) to override the default ContentBlock (see page 78) Xslt and display one of these custom Xsl transforms, if specified in the system. In addition, you can use the Xslt property to perform a transform on a content (see page 449) 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()]
public int StartingFolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron WebParts")>
<Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")>
<DefaultValue(0)>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable()>
Public Property StartingFolderID() As Integer
```

ContentBlock.SyncRoot Property

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

C#

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

Visual Basic

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

ContentList Class

The ContentList server control is used to display a list of content (see page 449) blocks on a Web page. In contrast to a List Summary, where content (see page 449) must be in a specified folder, the ContentList server control uses content (see page 449) 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 86)	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 86)	Populates the data for the ContentList (see page 85).

	GetEnumerator (see page 86)	Gets a System.Collections.IEnumerator for the ContentList (see page 85).
	GetRssFeed (see page 86)	This method returns the ContentList (see page 85) RSS feed.
	MoveNext (see page 87)	Move to next index.
	Reset (see page 87)	Resets the index.
	this (see page 87)	Returns the current ContentItem.ContentRow.
	ToString (see page 87)	Overrides the system ToString function.
	Wrap (see page 88)	Wrap the Html toolbar.

Legend

	Method

ContentList.CopyTo Method

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

C#

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

Visual Basic

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

ContentList.Fill Method

Populates the data for the ContentList (see page 85).

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 85).

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 85) 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 449) 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 89)	Sets the cache interval in second(s).
	ContentIds (see page 89)	The comma separated list of content (see page 449) IDs you wish to display.
	Count (see page 89)	Indicates the number of elements in your ContentList (see page 85).
	Current (see page 89)	Gets the current index.
	DefaultContentID (see page 90)	Gets or Sets the Default ContentID for this Control.
	Direction (see page 90)	Direction for sorting results, "ascending" or "descending".
	DynamicParameter (see page 90)	Gets or Sets the QueryString parameter to read a Content (see page 449) ID dynamically. Set to String.Empty if you wish to always display the default ContentID (static.)
	EkItems (see page 91)	Returns the array of ContentBase.
	GetHtml (see page 91)	If set to true, retrieves the content (see page 449) (Html body) of every content (see page 449) block in the collection for display.
	IncludeIcons (see page 91)	Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.
	IsSynchronized (see page 91)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 92)	Determines the type of window that appears on this Web form when a user (see page 534) clicks an item in the breadcrumb trail. The default is _self. • **_Self** = opens in same window • **_Top** = opens in parent window • **_Blank** = opens in new window • **_Parent** = opens in the parent frame
	MarkupLanguage (see page 92)	Enter name of a markup file that manages the display of this server control.
	MetaTag (see page 92)	Gets or Sets the meta tag value. Once this value is set, associated content (see page 449)(s) will be listed.
	OrderKey (see page 93)	Sort order key.
	Random (see page 93)	If set to true, grabs a single content (see page 449) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.
	SuppressWrapperTags (see page 93)	Suppresses the output of span/div tags around the control.
	SyncRoot (see page 93)	Gets an object that can be used to synchronize the ContentList (see page 85).
	XmlDoc (see page 94)	Returns the Xml object loaded for ContentList (see page 85).

Legend

	Property
	virtual

ContentList.CacheInterval Property

Sets the cache interval in second(s).

C#

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

Visual Basic

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

ContentList.ContentIds Property

The comma separated list of content (see page 449) IDs you wish to display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Description("The comma separated list of content IDs you wish to display.")]
[Personalizable()]
public string ContentIds;
```

Visual Basic

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

ContentList.Count Property

Indicates the number of elements in your ContentList (see page 85).

C#

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

Visual Basic

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

ContentList.Current Property

Gets the current index.

C#

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

Visual Basic

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

ContentList.DefaultContentID Property

Gets or Sets the Default ContentID for this Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control.")]
[Personalizable()]
public int DefaultContentID;
```

Visual Basic

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

ContentList.Direction Property

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

C#

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

Visual Basic

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

ContentList.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 449) 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 (see page 449) (Html body) of every content (see page 449) block in the collection for display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")]
public bool GetHtml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")>
Public Property GetHtml() As bool
```

ContentList.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: Only works with ecmSummary and ecmTeaser.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: Only works with ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: Only works with ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
Public Property IncludeIcons() As bool
```

ContentList.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

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

Visual Basic

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

ContentList.LinkTarget Property

Determines the type of window that appears on this Web form when a user (see page 534) 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.")]
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
```

ContentList.MetaTag Property

Gets or Sets the meta tag value. Once this value is set, associated content (see page 449)(s) will be listed.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Gets or Sets the meta tag value.  Once this value is set, associated content(s)
will be listed.")]
public string MetaTag;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Gets or Sets the meta tag value.  Once this value is set, associated content(s)
will be listed.")>
Public Property MetaTag() As string
```

ContentList.OrderKey Property

Sort order key.

C#

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

Visual Basic

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

ContentList.Random Property

If set to true, grabs a single content (see page 449) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.

C#

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

Visual Basic

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

ContentList.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 85).

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 85).

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 (see page 449) item. A visitor can rank the content (see page 449) 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 94)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 95)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user (see page 534) is logged in.
	Wrap (see page 95)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

ContentRating.Fill Method

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

C#

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

Visual Basic

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

ContentRating.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user (see page 534) 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 96)	The label displayed to a user (see page 534) thanking them for their content (see page 449) rating. Defaults to "Thank you for your feedback".
	BadLabel (see page 96)	The label corresponding to a "Bad" content (see page 449) rating. This defaults to "Poor".
	CharsRemainingLabel (see page 96)	Sets or gets the label for the number of additional characters a user (see page 534) may enter into the comment. The default is "Characters Remaining:"
	CompleteGraphDisplay (see page 97)	Described the layout manner which the graph is displayed to a user (see page 534) after content (see page 449) rating. The graph will be displayed to the right of the dialog in the event of Horizontal, below in the case of Vertical, and not displayed in the case of None.
	CookiesRequiredMessage (see page 97)	The message that appears when a user (see page 534) 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 (see page 449)."
	DefaultContentID (see page 98)	Gets or Sets the Default ContentID for this Control. This is the ContentID that will be displayed if the DynamicParameter (see page 98) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 98) of "id", and will display DefaultContentID otherwise.)
	DynamicParameter (see page 98)	Gets or Sets the QueryString parameter to read a Content (see page 449) ID dynamically. Set to "" (empty) if you wish to always display the default ContentID (static.)
	FeedbackHeader (see page 98)	The label corresponding to how the user (see page 534) would rate the content (see page 449) block. Defaults to "How would you rate the usefulness of this content (see page 449)?".
	GoodLabel (see page 99)	The label corresponding to a "Good" content (see page 449) rating. This defaults to "Outstanding".
	GraphBarColor (see page 99)	The color of the bars in the Content (see page 449) Rating bar graph.
	InitialGraphDisplay (see page 99)	Describes the layout manner which the graph is displayed to a user (see page 534) before content (see page 449) 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 100)	Padding, in pixels, to be applied between the content (see page 449) rating input and display graph.

	RatingLevelLabel (see page 100)	The label corresponding to a "Good" content (see page 449) rating. This defaults to "Outstanding".
	ThankYouMessage (see page 100)	The label displayed to a user (see page 534) thanking them for their content (see page 449) rating. Defaults to "Thank you for your feedback.".
	TotalRatingsLabel (see page 101)	The label that appears below the graph, and lists the number of rating submissions received. The default is: "Total Ratings".
	UserCommentsHeader (see page 101)	The label corresponding to why the user (see page 534) rated content (see page 449) the way they did. Defaults to "Tell us why you rated the content (see page 449) this way".
	ValidationMessage (see page 102)	The label displayed to a user (see page 534) when they try to submit a content (see page 449) rating without selecting a value of 1-9. Defaults to "Please select a rating before submitting your feedback.".

Legend

	Property

ContentRating.AlreadyRatedMessage Property

The label displayed to a user (see page 534) thanking them for their content (see page 449) rating. Defaults to "Thank you for your feedback".

C#

```
[Bindable(true)]
[Category("Ektron Labels")]
[Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_RATED)]
public string AlreadyRatedMessage;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label displayed to a user thanking them for their content rating. Defaults
to \"Thank you for your feedback.\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_RATED)>
Public Property AlreadyRatedMessage() As string
```

ContentRating.BadLabel Property

The label corresponding to a "Bad" content (see page 449) 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 (see page 534) 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 (see page 534) after content (see page 449) 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 (see page 534) 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 (see page 449)."

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 98) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 98) of "id", and will display DefaultContentID otherwise.)

C#

```
[Bindable(true)]
[Category("Ektron")]
[Editor(typeof(ContentRatingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[DefaultValue(0)]
[Description("Gets or Sets the Default ContentID for this Control. This is the ContentID that
will be displayed if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will get
the value 2 for a DynamicParameter of \"id\", and will display DefaultContentID otherwise.)")]
public int DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentRatingUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control. This is the ContentID that
will be displayed if the DynamicParameter does not get a value. (e.g. index.aspx?id=2 will get
the value 2 for a DynamicParameter of \"id\", and will display DefaultContentID otherwise.)")>
Public Property DefaultContentID() As Integer
```

ContentRating.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 449) 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 (see page 534) would rate the content (see page 449) block. Defaults to "How would you rate the usefulness of this content (see page 449)?".

C#

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

```
[Description("The label corresponding to how the user would rate the content block. Defaults
to \"How would you rate the usefullness of this content?\"")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_CNTRATE_FEED)]
public string FeedbackHeader;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron Labels")>
<Description("The label corresponding to how the user would rate the content block. Defaults
to \"How would you rate the usefullness of this content?\"")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_CNTRATE_FEED)>
Public Property FeedbackHeader() As string
```

ContentRating.GoodLabel Property

The label corresponding to a "Good" content (see page 449) 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 449) 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 (see page 534) before content (see page 449) 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 (see page 449) 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 (see page 449) 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 (see page 534) thanking them for their content (see page 449) 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 (see page 534) rated content (see page 449) the way they did. Defaults to "Tell us why you rated the content (see page 449) 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 (see page 534) when they try to submit a content (see page 449) 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 (see page 449) 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 (see page 449) item or reviews by a specific site or membership user (see page 534).

Class Hierarchy

C#

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

Visual Basic

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

File

ContentReview.cs

ContentReview Methods

The methods of the ContentReview class are listed here.

Public Methods

	Name	Description
	Fill (see page 103)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 103)	Wraps the resulting collection with the Ektron Toolbars, which will only display when a user (see page 534) is logged in.
	Wrap (see page 103)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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.ToString Method

Wraps the resulting collection with the Ektron Toolbars, which will only display when a user (see page 534) 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 104)	Gets or Sets the default content (see page 449) ID for this control. This is the content (see page 449) ID that will be rated if the DynamicParameter (see page 105) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 105) of "id", and will display DefaultContentID otherwise.)

	DisplayXslt (see page 105)	Select the type of review to display, or enter the path to a custom XSLT. (optional). Choices are: • **Ajax 5 Stars** - a five star rating system utilizing Ajax for display. • **Ajax 5 Stars Comment** - a five star rating system utilizing Ajax for display. In addition, when you hover over the stars, a review box pops-up below the stars. Comments that are added are submitted via an Ajax call. • **Ajax 5 Stars with Increments** - a five star rating system with half star increments that utilizes Ajax for display. • **5 Stars** - a five star rating system that allows... more (see page 105)
	DynamicParameter (see page 105)	To make this content (see page 449) review control dynamic, select id. When you do, this server control is attached to the content (see page 449) block passed as a URL parameter. When using this control to retrieve a user (see page 534)'s reviews, set this property to UserId to make the User (see page 534)'s ID dynamic.
	GetReviews (see page 106)	This property returns a list of reviews for a content (see page 449) item or a specified user (see page 534). 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 106)	The number of reviews to retrieve if the GetReview property is set to Content (see page 449) or User (see page 534). 0 (zero) = unlimited results.
	Moderate (see page 106)	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 449)" > "Content (see page 449) Rating"
	RatingsMinimum (see page 107)	Sets a minimum number of the reviews and ratings before displaying the average rating.
	UserID (see page 107)	The user (see page 534) ID for which to get reviews. If left blank, reviews from all users are returned. If a user (see page 534) ID is specified, only reviews for that user (see page 534) are returned. Important: The GetReviews (see page 106) property must be set to User (see page 534) for the control to use this property.
	XmlDoc (see page 107)	Returns the Xslt object for the collection.

Legend

	Property

ContentReview.DefaultContentID Property

Gets or Sets the default content (see page 449) ID for this control. This is the content (see page 449) ID that will be rated if the DynamicParameter (see page 105) does not get a value. (e.g. index.aspx?id=2 will get the value 2 for a DynamicParameter (see page 105) 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 int 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 Integer
```

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 (see page 449).
- **5 Stars with Increments** - a five star rating system with half star increments that allows visitors to add text reviews of content (see page 449).
- **Review List** - displays a list of reviews for content (see page 449) or a user (see page 534). 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 (see page 449) review control dynamic, select id. When you do, this server control is attached to the content (see page 449) block passed as a URL parameter. When using this control to retrieve a user (see page 534)'s reviews, set this property to UserId to make the User (see page 534)'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.GetReviews Property

This property returns a list of reviews for a content (see page 449) item or a specified user (see page 534). 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 449) or User (see page 534). 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 449)” > “Content (see page 449) 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 (see page 534) ID for which to get reviews. If left blank, reviews from all users are returned. If a user (see page 534) ID is specified, only reviews for that user (see page 534) are returned.

Important: The GetReviews (see page 106) property must be set to User (see page 534) 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 int 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 Integer
```

ContentReview.XmlDoc Property

Returns the Xslt object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

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

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 229) Control** - displays the Login (see page 229) server control to show that the Web service is connected and working properly. When not connecting properly, an error message is displayed.

Class Hierarchy

System.Web.UI.WebControls.WebControl → Ektron.Cms.Controls.EkWebControl → Ektron.Cms.Controls.DesignTimeDiagnostic

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

Legend

	Method

DesignTimeDiagnostic.Fill Method

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

C#

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

Visual Basic

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

DesignTimeDiagnostic Properties

The properties of the DesignTimeDiagnostic class are listed here.

Public Properties

	Name	Description
	Authenticated (see page 109)	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 109)	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 109)	Property that lets a designer view content blocks, collections etc. in different languages.
	SuppressWrapperTags (see page 109)	Suppresses the output of span/div tags around the control.
	WrapTag (see page 109)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property

DesignTimeDiagnostic.Authenticated Property

DESIGN TIME ONLY: Used to indicate to a user that they are logged in, and permits them to login through a login dialog in Visual Studio.NET.

C#

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

Visual Basic

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

DesignTimeDiagnostic.Hide Property

Alias for the 'Visible' property. Visible is set to mirror this value. This property is used to hide output in design time, whereas Visible does not hide output in design time.

C#

```
[Browsable(false)]
public new bool Hide;
```

Visual Basic

```
<Browsable(false)>
Public new Property Hide() As bool
```

DesignTimeDiagnostic.Language Property

Property that lets a designer view content blocks, collections etc. in different languages.

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.DhtmlMenu Constructor

C#

```
public DhtmlMenu();
```

Visual Basic

```
Public Sub DhtmlMenu()
```

Description

Class constructor calls the base class's constructor

DhtmlMenu Methods

The methods of the DhtmlMenu class are listed here.

Public Methods

	Name	Description
	Fill (see page 110)	Calls up the database and updates the contents of the Menu (see page 246) if the data is out of sync with the specified parameters.
	Wrap (see page 111)	Method that can be overridden and used to wrap an Ektron Toolbar corresponding to a particular Control around some Html text.

Legend

	Method

DhtmlMenu.Fill Method

Calls up the database and updates the contents of the Menu (see page 246) 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 112)	Ignored. The property has no effect.
	Align (see page 112)	Specify the horizontal alignment for the root menu text.
	BackColor (see page 112)	BackColor sets the menu background color.
	BorderColor (see page 112)	Sets the border color around the editor.
	BorderStyle (see page 112)	Sets the border style around the editor.
	BorderWidth (see page 113)	Sets the border width around the editor.
	CssClass (see page 113)	Specify the class name to use instead of internal style.
	DisplayXslt (see page 113)	Ignored. The property has no effect.
	Enabled (see page 113)	Ignored. The property has no effect.
	EnableIE6FormsFix (see page 113)	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 114)	The height (in pixels or percent) of the editor, including toolbars.
	HorizontalDisplay (see page 114)	Popup root menu horizontally.
	MenuHoverOverColor (see page 114)	`HoverOverColor` sets the menu hoverover color.
	MenuStyle (see page 115)	Specify the custom style to use. If specified, all specified appearance properties will be ignored.
	Stylesheet (see page 115)	Specify the style sheet relative to site root. Leave blank for default.
	SubMenuBackColor (see page 115)	`SubMenuBackColor` sets the sub menu background color.
	SubMenuHoverOverColor (see page 116)	`SubMenuHoverOverColor` sets the menu hoverover color.
	SubMenuItemHeight (see page 116)	`SubMenuItemHeight` is the height in pixels of submenu items.
	SubMenuItemWidth (see page 116)	`SubMenuItemWidth` is the width in pixels of submenu items.
	SubMenuTopOffset (see page 117)	`SubMenuTopOffset` sets the horizontal offset of submenu popups relative to the parent submenu item.
	TabIndex (see page 117)	Ignored. The property has no effect.
	ToolTip (see page 117)	Ignored. The property has no effect.
	Width (see page 117)	The width (in pixels or percent) of the editor, including toolbars.

Legend

	Property

DhtmlMenu.AccessKey Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string AccessKey;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property AccessKey() As string
```

DhtmlMenu.Align Property

Specify the horizontal alignment for the root menu text.

C#

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

Visual Basic

```
<Description("Specify the horizontal alignment for the root menu text."), Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(HorizontalAlign))>
Public Property Align() As HorizontalAlign
```

DhtmlMenu.BackColor Property

BackColor sets the menu background color.

C#

```
[NotifyParentProperty(true), Browsable(true), Bindable(true)]
[Description("BackColor sets the menu background color.")]
public override System.Drawing.Color BackColor;
```

Visual Basic

```
<NotifyParentProperty(true), Browsable(true), Bindable(true)>
<Description("BackColor sets the menu background color.")>
Public override Property BackColor() As System.Drawing.Color
```

DhtmlMenu.BorderColor Property

Sets the border color around the editor.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override System.Drawing.Color BorderColor;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property BorderColor() As System.Drawing.Color
```

DhtmlMenu.BorderStyle Property

Sets the border style around the editor.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override BorderStyle BorderStyle;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property BorderStyle() As BorderStyle
```

DhtmlMenu.BorderWidth Property

Sets the border width around the editor.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override Unit BorderWidth;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property BorderWidth() As Unit
```

DhtmlMenu.CssClass Property

Specify the class name to use instead of internal style.

C#

```
[Description("Specify the class name to use instead of internal style."), Category("Ektron")]
public override string CssClass;
```

Visual Basic

```
<Description("Specify the class name to use instead of internal style."), Category("Ektron")>
Public override Property CssClass() As string
```

DhtmlMenu.DisplayXslt Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string DisplayXslt;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property DisplayXslt() As string
```

DhtmlMenu.Enabled Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override bool Enabled;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property Enabled() As bool
```

DhtmlMenu.EnableIE6FormsFix Property

This should only be enabled when you have a DHTML menu rendering over an HTML Select element. In IE6, the DHTML will render below the drop down. This will temporarily disable the element when the DHTML menu is rendered.

C#

```
[Browsable(true)]
[Description("This should only be enabled when you have a DHTML menu rendering over an HTML
Select element. In IE6, the DHTML will render below the drop down. This will temporarily
disable the element when the DHTML menu is rendered.")]
[Bindable(true)]
[Category("Ektron")]
public bool EnableIE6FormsFix;
```

Visual Basic

```
<Browsable(true)>
<Description("This should only be enabled when you have a DHTML menu rendering over an HTML
Select element. In IE6, the DHTML will render below the drop down. This will temporarily
disable the element when the DHTML menu is rendered.")>
<Bindable(true)>
<Category("Ektron")>
Public Property EnableIE6FormsFix() As bool
```

DhtmlMenu.Height Property

The height (in pixels or percent) of the editor, including toolbars.

C#

```
[NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultHeight)]
[Description("The height (in pixels or percent) of the editor, including toolbars.")]
public override Unit Height;
```

Visual Basic

```
<NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultHeight)>
<Description("The height (in pixels or percent) of the editor, including toolbars.")>
Public override Property Height() As Unit
```

DhtmlMenu.HorizontalDisplay Property

Popup root menu horizontally.

C#

```
[Description("Popup root menu horizontally."), Category("Ektron")]
[Browsable(true)]
[Bindable(false)]
[DefaultValue(false)]
public bool HorizontalDisplay;
```

Visual Basic

```
<Description("Popup root menu horizontally."), Category("Ektron")>
<Browsable(true)>
<Bindable(false)>
<DefaultValue(false)>
Public Property HorizontalDisplay() As bool
```

DhtmlMenu.MenuHoverOverColor Property

C#

```
[Description("HoverOverColor sets the menu hoverover color."), Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color MenuHoverOverColor;
```

Visual Basic

```
<Description("HoverOverColor sets the menu hoverover color."), Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property MenuHoverOverColor() As Color
```

Description

`HoverOverColor` sets the menu hoverover color.

DhtmlMenu.MenuStyle Property

Specify the custom style to use. If specified, all specified appearance properties will be ignored.

C#

```
[Description("Specify the custom style to use. If specified, all specified appearance
properties will be ignored."), Category("Ektron")]
public string MenuStyle;
```

Visual Basic

```
<Description("Specify the custom style to use. If specified, all specified appearance
properties will be ignored."), Category("Ektron")>
Public Property MenuStyle() As string
```

DhtmlMenu.Stylesheet Property

Specify the style sheet relative to site root. Leave blank for default.

C#

```
[Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(StringConverter))]
public override string Stylesheet;
```

Visual Basic

```
<Description("Specify the style sheet relative to site root. Leave blank for default."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(StringConverter))>
Public override Property Stylesheet() As string
```

DhtmlMenu.SubMenuBackColor Property

C#

```
[Description("SubMenuBackColor sets the sub menu background color."), Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color SubMenuBackColor;
```

Visual Basic

```
<Description("SubMenuBackColor sets the sub menu background color."), Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property SubMenuBackColor() As Color
```

Description

`SubMenuBackColor` sets the sub menu background color.

DhtmlMenu.SubMenuHoverOverColor Property

C#

```
[Description("SubMenuHoverOverColor sets the sub menu hoverover color."), Category("Ektron")]
[Bindable(true)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color SubMenuHoverOverColor;
```

Visual Basic

```
<Description("SubMenuHoverOverColor sets the sub menu hoverover color."), Category("Ektron")>
<Bindable(true)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property SubMenuHoverOverColor() As Color
```

Description

`SubMenuHoverOverColor` sets the menu hoverover color.

DhtmlMenu.SubMenuItemHeight Property

C#

```
[Description("SubMenuItemHeight is the height in pixels of submenu items."),
Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(Int16Converter))]
public Int16 SubMenuItemHeight;
```

Visual Basic

```
<Description("SubMenuItemHeight is the height in pixels of submenu items."),
Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(Int16Converter))>
Public Property SubMenuItemHeight() As Int16
```

Description

`SubMenuItemHeight` is the height in pixels of submenu items.

DhtmlMenu.SubMenuItemWidth Property

C#

```
[Description("SubMenuItemWidth is the width in pixels of submenu items."), Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(Int16Converter))]
public Int16 SubMenuItemWidth;
```

Visual Basic

```
<Description("SubMenuItemWidth is the width in pixels of submenu items."), Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(Int16Converter))>
Public Property SubMenuItemWidth() As Int16
```

Description

`SubMenuItemWidth` is the width in pixels of submenu items.

DhtmlMenu.SubMenuTopOffset Property

C#

```
[Description("SubMenuTopOffset sets the horizontal offset of submenu popups relative to the
parent submenu item."), Category("Ektron")]
[Bindable(true)]
[TypeConverter(typeof(Int16Converter))]
public Int16 SubMenuTopOffset;
```

Visual Basic

```
<Description("SubMenuTopOffset sets the horizontal offset of submenu popups relative to the
parent submenu item."), Category("Ektron")>
<Bindable(true)>
<TypeConverter(typeof(Int16Converter))>
Public Property SubMenuTopOffset() As Int16
```

Description

`SubMenuTopOffset` sets the horizontal offset of submenu popups relative to the parent submenu item.

DhtmlMenu.TabIndex Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override short TabIndex;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property TabIndex() As short
```

DhtmlMenu.ToolTip Property

Ignored. The property has no effect.

C#

```
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public override string ToolTip;
```

Visual Basic

```
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public override Property ToolTip() As string
```

DhtmlMenu.Width Property

The width (in pixels or percent) of the editor, including toolbars.

C#

```
[NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultWidth)]
[Description("The width (in pixels or percent) of the editor, including toolbars.")]
public override Unit Width;
```

Visual Basic

```
<NotifyParentProperty(true), Browsable(true), Bindable(true), Category("Layout"),
DefaultValue(m_DefaultWidth)>
<Description("The width (in pixels or percent) of the editor, including toolbars.")>
Public override Property Width() As Unit
```

Directory Class

The Directory server control lets you customize the behavior of the Taxonomy (see page 498) 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 119)	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 119)	Populates the data for the Taxonomy (see page 498).
	GetEnumerator (see page 119)	Gets a System.Collections.IEnumerator for the Taxonomy (see page 498).
	GetView (see page 119)	Gets the view data using dataset
	GetViewNames (see page 119)	gets the view names
	MoveNext (see page 120)	Move to next index.
	Reset (see page 120)	Resets the index.
	this (see page 120)	Returns the current ContentItem.ContentRow.
	ToString (see page 120)	Overrides the system ToString function.
	Wrap (see page 121)	Wrap the Html toolbar.

Legend

	Method

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 498).

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 498).

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 449) 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
	CacheInterval (see page 122)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a 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 122)	Indicates the number of elements in your Taxonomy (see page 498).
	Current (see page 123)	Gets the current index.
	DisplayXslt (see page 123)	The Xslt to use to render the control (optional).
	DynamicParameter (see page 123)	Gets or Sets the QueryString parameter to read a Taxonomy (see page 498) ID dynamically. Set to String.Empty if you wish to always display the default Taxonomy (see page 498) ID (static.)
	EkItems (see page 123)	Returns the array of ContentBase.
	EnableAjax (see page 124)	Set to true to enable Ajax searches. When enabled, the MaxResults (see page 126) property determines the maximum number of results per page. • **True** = Enable Ajax Search (see page 279) (default value) • **False** = Original HTML Search (see page 279)
	EnablePaging (see page 124)	This property, in conjunction with the MaxResults (see page 126) 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 126) property.
	EnableSearch (see page 124)	Set to true if you want a search (see page 514) box (shown below) to appear above this taxonomy display.
	FolderID (see page 124)	The ID of the folder you wish to select data from. Root is 0.
	IsSynchronized (see page 125)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 125)	Defines the way a link acts when it is clicked. The choices are: • **_blank** = loads the link in a new blank window. This window is not named. • **_self** = loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. • **_parent** = loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** = loads the link in the full body of the window. This defaults to acting like "_self" if the document is already... more (see page 125)
	MarkupLanguage (see page 125)	Enter name of a markup file that manages the display of this server control.
	MaxResults (see page 126)	Maximum number of results to be returned. If set to zero, there is no limit.
	Recursive (see page 126)	Determines if subfolders of the selected folders should be brought into the list.
	ShowAllChecked (see page 126)	Check show all category checkbox.

	ShowCount (see page 126)	Indicates if the number of taxonomy items appears next to each category when displayed in the Web site. The default is: False
	SyncRoot (see page 127)	Gets an object that can be used to synchronize the Taxonomy (see page 498).
	TaxonomyCols (see page 127)	Enter the number of columns in which this taxonomy/category will appear on the page.
	TaxonomyDepth (see page 127)	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 appears. **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 498)'s XML.
	TaxonomyDescription (see page 127)	Returns taxonomy's description.
	TaxonomyId (see page 128)	Enter the ID number of the taxonomy or category to appear in this server control. Root is 0.
	TaxonomyItemCols (see page 128)	Enter the number of columns in which this taxonomy/category items(articles) will appear on the page.
	TaxonomyPath (see page 128)	Returns taxonomy's path.
	TaxonomyTitle (see page 128)	Returns the Taxonomy (see page 498)'s title.
	TaxonomyTreeData (see page 128)	Returns taxonomy's data.
	TaxonomyXml (see page 129)	Returns taxonomy's XML. (includes subcategory and items)
	XmlDoc (see page 129)	Returns the XML object loaded for Taxonomy (see page 498).

Legend

	Property
	virtual

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).")]
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).")>
Public virtual Property CacheInterval() As double
```

Directory.Count Property

Indicates the number of elements in your Taxonomy (see page 498).

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 498) ID dynamically. Set to String.Empty if you wish to always display the default Taxonomy (see page 498) 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 126) property determines the maximum number of results per page.

- **True** = Enable Ajax Search (see page 279) (default value)
- **False** = Original HTML Search (see page 279)

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true), Description("Enable/Disable ajax
based retrival.")]
public bool EnableAjax;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true), Description("Enable/Disable ajax
based retrival.")>
Public Property EnableAjax() As bool
```

Directory.EnablePaging Property

This property, in conjunction with the MaxResults (see page 126) 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 126) 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.")]
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.")>
Public Property EnablePaging() As bool
```

Directory.EnableSearch Property

Set to true if you want a search (see page 514) 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.")]
public 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.")>
Public Property EnableSearch() As Boolean
```

Directory.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))]
public int FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderID() As Integer
```

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.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.")]
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
```

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 fetch. If set to zero, there is no limit.")]
public int MaxResults;
```

Visual Basic

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

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 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 Property Recursive() As bool
```

Directory.ShowAllChecked Property

Check show all category checkbox.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("false"), Browsable(true)]
[Description("Check show all category checkbox.")]
public bool ShowAllChecked;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("false"), Browsable(true)>
<Description("Check show all category checkbox.")>
Public 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")]
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")>
Public Property ShowCount() As bool
```

Directory.SyncRoot Property

Gets an object that can be used to synchronize the Taxonomy (see page 498).

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.")]
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
```

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 appears.

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 498)'s XML.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)]
[Description("The depth of the taxonomy category.")]
public int TaxonomyDepth;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("The depth of the taxonomy category.")>
Public 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))]
public int 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 Integer
```

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.")]
public int TaxonomyItemCols;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue("1"), Browsable(true)>
<Description("This aligns the taxonomy categories in number of columns.")>
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 498)'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 498).

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 130)	The ID of the blog in CMS400.NET.
	BoardID (see page 130)	The ID of the discussion board.
	CalendarDynamicParameter (see page 131)	To make this calendar (see page 447) dynamic, select calendar_id. When you do, this server control uses the calendar (see page 447) passed as a URL parameter.
	CollectionDynamicParameter (see page 131)	To make this collection dynamic, select id. When you do, this server control uses the collection passed as a URL parameter.
	ContentBlockDynamicParameter (see page 131)	Select id. When you do, this server control uses the content (see page 449) block passed as a URL parameter.
	DefaultCalendarID (see page 131)	The ID number of the calendar (see page 447) that appears where you inserted this server control.

	DefaultCollectionID (see page 132)	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 132)	The ID of a content (see page 449) block that appears where you insert this server control.
	DefaultFormID (see page 132)	The ID of a FormBlock (see page 156) that appears where you inserted this server control if no other form block is identified.
	EnableBlogControl (see page 132)	Set to True to allow a Blog (see page 22) Control to appear in the catalog.
	EnableCalendarControl (see page 133)	Set to true to allow a Calendar (see page 64) control to appear in the Catalog.
	EnableCollectionControl (see page 133)	Set to True to allow a Collection control to appear in the Catalog.
	EnableContentBlockControl (see page 133)	Set to True to allow a Content (see page 449) Block Control to appear in the catalog.
	EnableDiscussionBoardsControl (see page 133)	Set to True to allow a Forum (see page 167) (Discussion Board) control to appear in the catalog.
	EnableFormBlockControl (see page 133)	Set to True to allow a FormBlock (see page 156) Control to appear in the catalog.
	EnableListSummaryControl (see page 134)	Set to True to allow a ListSummary (see page 217) Control to appear in the catalog.
	EnableThreadedDiscussionsControl (see page 134)	Set to True to allow a Forum (see page 167) (Discussion Board) control to appear in the catalog.
	FolderID (see page 134)	The folder that contains the content (see page 449) blocks which appear in the list summary.
	FormBlockDynamicParameter (see page 134)	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 135)	The descriptive name of the Ektron Catalog. The title appears as clickable link in the catalog when the Ektron Catalog is not shown.

Legend

	Property

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")]
public int BlogID;
```

Visual Basic

```
<Editor(typeof(DefaultBlogIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron Blog Control")>
Public Property BlogID() As Integer
```

EktronCatalogPart.BoardID Property

The ID of the discussion board.

C#

```
[Browsable(true), DefaultValue(0)]
[Category("Ektron Discussion Boards Control")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int BoardID;
```

Visual Basic

```
<Browsable(true), DefaultValue(0)>
<Category("Ektron Discussion Boards Control")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As Integer
```

EktronCatalogPart.CalendarDynamicParameter Property

To make this calendar (see page 447) dynamic, select calendar_id. When you do, this server control uses the calendar (see page 447) passed as a URL parameter.

C#

```
[TypeConverter(typeof(DynamicCalendarTypeConverter))]
[Browsable(true), DefaultValue("")]
[Category("Ektron Calendar Control")]
public string CalendarDynamicParameter;
```

Visual Basic

```
<TypeConverter(typeof(DynamicCalendarTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron Calendar Control")>
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")]
public string CollectionDynamicParameter;
```

Visual Basic

```
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron Collection Control")>
Public Property CollectionDynamicParameter() As string
```

EktronCatalogPart.ContentBlockDynamicParameter Property

Select id. When you do, this server control uses the content (see page 449) block passed as a URL parameter.

C#

```
[TypeConverter(typeof(DynamicContentIDTypeConverter))]
[Browsable(true), DefaultValue("")]
[Category("Ektron Content Block")]
public string ContentBlockDynamicParameter;
```

Visual Basic

```
<TypeConverter(typeof(DynamicContentIDTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron Content Block")>
Public Property ContentBlockDynamicParameter() As string
```

EktronCatalogPart.DefaultCalendarID Property

The ID number of the calendar (see page 447) 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")]
public int DefaultCalendarID;
```

Visual Basic

```
<Editor(typeof(CalendarUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
```

```
<Category("Ektron Calendar Control")>
Public Property DefaultCalendarID() As Integer
```

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")]
public int DefaultCollectionID;
```

Visual Basic

```
<Browsable(true), DefaultValue(0)>
<Editor(typeof(CollectionUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Category("Ektron Collection Control")>
Public Property DefaultCollectionID() As Integer
```

EktronCatalogPart.DefaultContentID Property

The ID of a content (see page 449) 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")]
public int DefaultContentID;
```

Visual Basic

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

EktronCatalogPart.DefaultFormID Property

The ID of a FormBlock (see page 156) 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")]
public int DefaultFormID;
```

Visual Basic

```
<Editor(typeof(FormUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron FormBlock Control")>
Public Property DefaultFormID() As Integer
```

EktronCatalogPart.EnableBlogControl Property

Set to True to allow a Blog (see page 22) Control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Blog Control")]
public bool EnableBlogControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Blog Control")>
```

```
Public Property EnableBlogControl() As bool
```

EktronCatalogPart.EnableCalendarControl Property

Set to true to allow a Calendar (see page 64) control to appear in the Catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Calendar Control")]
public bool EnableCalendarControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Calendar Control")>
Public Property EnableCalendarControl() As bool
```

EktronCatalogPart.EnableCollectionControl Property

Set to True to allow a Collection control to appear in the Catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Collection Control")]
public bool EnableCollectionControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Collection Control")>
Public Property EnableCollectionControl() As bool
```

EktronCatalogPart.EnableContentBlockControl Property

Set to True to allow a Content (see page 449) Block Control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Content Block")]
public bool EnableContentBlockControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Content Block")>
Public Property EnableContentBlockControl() As bool
```

EktronCatalogPart.EnableDiscussionBoardsControl Property

Set to True to allow a Forum (see page 167) (Discussion Board) control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Discussion Boards Control")]
public bool EnableDiscussionBoardsControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Discussion Boards Control")>
Public Property EnableDiscussionBoardsControl() As bool
```

EktronCatalogPart.EnableFormBlockControl Property

Set to True to allow a FormBlock (see page 156) Control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron FormBlock Control")]
public bool EnableFormBlockControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron FormBlock Control")>
Public Property EnableFormBlockControl() As bool
```

EktronCatalogPart.EnableListSummaryControl Property

Set to True to allow a ListSummary (see page 217) Control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron ListSummary")]
public bool EnableListSummaryControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron ListSummary")>
Public Property EnableListSummaryControl() As bool
```

EktronCatalogPart.EnableThreadedDiscussionsControl Property

Set to True to allow a Forum (see page 167) (Discussion Board) control to appear in the catalog.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron Discussion Boards Control")]
public bool EnableThreadedDiscussionsControl;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron Discussion Boards Control")>
Public Property EnableThreadedDiscussionsControl() As bool
```

EktronCatalogPart.FolderID Property

The folder that contains the content (see page 449) blocks which appear in the list summary.

C#

```
[Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Browsable(true), DefaultValue(0)]
[Category("Ektron ListSummary")]
public int FolderID;
```

Visual Basic

```
<Editor(typeof(FolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Browsable(true), DefaultValue(0)>
<Category("Ektron ListSummary")>
Public Property FolderID() As Integer
```

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")]
public string FormBlockDynamicParameter;
```

Visual Basic

```
<TypeConverter(typeof(DynamicFormIDTypeConverter))>
<Browsable(true), DefaultValue("")>
<Category("Ektron FormBlock Control")>
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")]
public override string Title;
```

Visual Basic

```
<Browsable(true), DefaultValue("Ektron Catalog Part")>
<Category("Ektron")>
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 136)	When set to True, the option to add a blog control to a Web part is available.
	EnableDiscussionBoards (see page 136)	When set to True, the option to add a Forum (see page 167) (Discussion Board) to a Web part is available.
	EnableForums (see page 136)	When set to True, the option to add a Forum (see page 167) (Discussion Board) to a Web part is available.
	Title (see page 136)	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.

Legend

	Property

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")]
public bool EnableBlogs;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron")>
Public Property EnableBlogs() As bool
```

EktronCommunityCatalogPart.EnableDiscussionBoards Property

When set to True, the option to add a Forum (see page 167) (Discussion Board) to a Web part is available.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron")]
public bool EnableDiscussionBoards;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron")>
Public Property EnableDiscussionBoards() As bool
```

EktronCommunityCatalogPart.EnableForums Property

When set to True, the option to add a Forum (see page 167) (Discussion Board) to a Web part is available.

C#

```
[Browsable(true), DefaultValue(true)]
[Category("Ektron")]
public bool EnableForums;
```

Visual Basic

```
<Browsable(true), DefaultValue(true)>
<Category("Ektron")>
Public Property EnableForums() As bool
```

EktronCommunityCatalogPart.Title Property

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.

C#

```
[Browsable(true), DefaultValue("Ektron Catalog Part")]
[Category("Ektron")]
public override string Title;
```

Visual Basic

```
<Browsable(true), DefaultValue("Ektron Catalog Part")>
<Category("Ektron")>
Public override Property Title() As string
```

EktronWebPartZone Class

Similar to a Microsoft WebPartZone, the EktronWebPartZone server control provides a location for content (see page 449) 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 137)	DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.

Legend

	Property

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

System.Web.UI.WebControls.WebControl → Ektron.Cms.Controls.EkWebControl

C#

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

Visual Basic

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

File

EkWebControl.cs

EkWebControl.EkWebControl Constructor

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

C#

```
public EkWebControl();
```

Visual Basic

```
Public Sub EkWebControl()
```

EkWebControl Methods

The methods of the EkWebControl class are listed here.

Public Methods

	Name	Description
V	Fill (see page 138)	Method that can be overridden and used to grab data from the back end for each server control.
	LocalizeString (see page 139)	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 (see page 534) would like to be localized to the sites default language should begin with the constant STR_LOCALIZE defined in the EkWebControl (see page 138) class.
	RegisterClientScriptInclude (see page 139)	This method adds JavaScript include in the page header. You can call this method multiple times with same script path because it will only register once.
	RegisterStylesheet (see page 139)	This methods adds a stylesheet link tag in the page header. You can call this method multiple times with same stylesheet path because it will only register once.
	SearchHeader (see page 140)	This methods searches for the literal-control containing the head-definition
	ToString (see page 140)	Renders the control for output as a string. Use this method to display your controls if you instantiate your control as a component in code behind.
V	Wrap (see page 140)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method
V	virtual

EkWebControl.Fill Method

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

C#

```
public virtual void Fill();
```

Visual Basic

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

EkWebControl.LocalizeString Method

Used to localize strings in the control properties window and at render time. Strings should be defined in resource files. Any property that the user (see page 534) would like to be localized to the sites default language should begin with the constant STR_LOCALIZE defined in the EkWebControl (see page 138) 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 363).

EkWebControl.RegisterClientScriptInclude Method

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

C#

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

Visual Basic

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

Parameters

Parameters	Description
path	JavaScript include path

EkWebControl.RegisterStylesheet Method

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

C#

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

Visual Basic

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

Parameters

Parameters	Description
path	Stylesheet path

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(index As ref int) As LiteralControl
```

Parameters

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

Returns

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

EkWebControl.ToString Method

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

C#

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

Visual Basic

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

EkWebControl.Wrap Method

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

C#

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

Visual Basic

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

Parameters

Parameters	Description
Html	Html string to be wrapped with a toolbar

Returns

Html string wrapped with a toolbar

EkWebControl Properties

The properties of the EkWebControl class are listed here.

Public Properties

	Name	Description
	Authenticated (see page 141)	DESIGN TIME ONLY: Used to indicate to a user (see page 534) that they are logged in, and permits them to login through a login dialog in Visual Studio.NET.
	CmsSitePath (see page 141)	DESIGN TIME ONLY: Used to find the web service required to access the CMS in Design Time.
	Error (see page 142)	An exception caught in the rendering of this Server Control.
	Hide (see page 142)	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 142)	DESIGN TIME ONLY: Read only property that lets a developer know if the user (see page 534) is currently log into the site as a user (see page 534).
	Language (see page 142)	Property that lets a designer view content (see page 449) blocks, collections etc. in different languages.
	LanguageID (see page 143)	This property lets a designer view content (see page 449) blocks, collections etc in different languages.
	loggedInUserID (see page 143)	DESIGN TIME ONLY: Read only property that lets a developer know the user (see page 534) ID of the user (see page 534) who is currently logged in.
	loggedInUserName (see page 143)	DESIGN TIME ONLY: Read only property that lets a developer know the user (see page 534) name of the user (see page 534) who is currently logged in.
V	SuppressWrapperTags (see page 143)	Suppresses the output of span/div tags around the control.
V	Text (see page 144)	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 144)	Allows a developer to specify control's tag. Default is assumed "".

Legend

	Property
V	virtual

EkWebControl.Authenticated Property

DESIGN TIME ONLY: Used to indicate to a user (see page 534) 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 (see page 534) is currently log into the site as a user (see page 534).

C#

```
public bool IsLoggedIn;
```

Visual Basic

```
Public Property IsLoggedIn() As bool
```

EkWebControl.Language Property

Property that lets a designer view content (see page 449) blocks, collections etc. in different languages.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(1033)]
```

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

Visual Basic

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

EkWebControl.LanguageID Property

This property lets a designer view content (see page 449) blocks, collections etc in different languages.

C#

```
public int LanguageID;
```

Visual Basic

```
Public Property LanguageID() As Integer
```

EkWebControl.loggedInUserID Property

DESIGN TIME ONLY: Read only property that lets a developer know the user (see page 534) ID of the user (see page 534) who is currently logged in.

C#

```
public int loggedInUserID;
```

Visual Basic

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

EkWebControl.loggedInUserName Property

DESIGN TIME ONLY: Read only property that lets a developer know the user (see page 534) name of the user (see page 534) who is currently logged in.

C#

```
public string loggedInUserName;
```

Visual Basic

```
Public Property loggedInUserName() As string
```

EkWebControl.SuppressWrapperTags Property

Suppresses the output of span/div tags around the control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Suppresses the output of span/div tags around the control.")]
public virtual bool SuppressWrapperTags;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Suppresses the output of span/div tags around the control.")>
```

```
Public virtual Property SuppressWrapperTags() As bool
```

EkWebControl.Text Property

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

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Allows a developer to treat the server control as a Label, taking information
from the Control and painting it on top of the Control for display. Leave blank for default
display.")]
[DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)]
[Browsable(false)]
public virtual string Text;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Allows a developer to treat the server control as a Label, taking information
from the Control and painting it on top of the Control for display. Leave blank for default
display.")>
<DesignerSerializationVisibility(System.ComponentModel.DesignerSerializationVisibility.Hidden)>
<Browsable(false)>
Public virtual Property Text() As string
```

EkWebControl.WrapTag Property

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

C#

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

Visual Basic

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

EkWebPart Class

Class Hierarchy

C#

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

Visual Basic

```
Public abstract Class EkWebPart
```

```
Inherits EkWebControl
Inherits IWebPart
Inherits IWebEditable
```

File

EkWebPart.cs

Description

This is class Ektron.Cms.Controls.EkWebPart.

EkXsltBase Class

Summary description for EkXsltBase.

Class Hierarchy

C#

```
public abstract class EkXsltBase : EkWebControl;
```

Visual Basic

```
Public abstract Class EkXsltBase
Inherits EkWebControl
```

File

EkXsltBase.cs

EkXsltBase.EkXsltBase Constructor

Constructor for EkXlstBase

C#

```
public EkXsltBase();
```

Visual Basic

```
Public Sub EkXsltBase()
```

EkXsltBase Methods

The methods of the EkXsltBase class are listed here.

Public Methods

	Name	Description
	SetDirty (see page 145)	Marks the Control to call the database since output-dependent parameters have been changed.

Legend

	Method

EkXsltBase.SetDirty Method

Marks the Control to call the database since output-dependent parameters have been changed.

C#

```
public new void SetDirty();
```

Visual Basic

```
Public new Function SetDirty() As void
```

EkXsltBase Properties

The properties of the EkXsltBase class are listed here.

Public Properties

	Name	Description
	DisplayXslt (see page 146)	The Xslt to use to render the control (optional).
	XmlDoc (see page 146)	XmlDocument used in this control.
	XsltDoc (see page 146)	Xslt used in this control.

Legend

	Property
	virtual

EkXsltBase.DisplayXslt Property

The Xslt to use to render the control (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltTypeConverter))]
[DefaultValue("ecmNavigation")]
[Personalizable()]
public virtual string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltTypeConverter))>
<DefaultValue("ecmNavigation")>
<Personalizable()>
Public virtual Property DisplayXslt() As string
```

EkXsltBase.XmlDoc Property

XmlDocument used in this control.

C#

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

Visual Basic

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

EkXsltBase.XsltDoc Property

Xslt used in this control.

C#

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

Visual Basic

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

FlexMenu Class

The FlexMenu server control displays a menu on a Web form. Although it looks and acts like a Smart Menu (see page 246), 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 147)	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 147)	Marks the Control to call the database since output-dependent parameters have been changed.

Legend

	Method

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 148)	Set to True to automatically close other menu branches when a new one is opened.
	CacheInterval (see page 148)	Sets the cache interval in second(s).
	DisplayXslt (see page 149)	The Xslt used to render the Menu (see page 246).
	EnableAjax (see page 149)	Set to true to enable Ajax, which only downloads sub-menus as needed.
	EnableMouseOverPopUp (see page 149)	Set to True and the submenus appear as soon as the cursor moves over them.
	EnableSmartOpen (see page 150)	When true, menu will auto-open to show matching IDs or folder associations.
	EnableSubmenuLinkPathMatching (see page 150)	Set to enable auto-submenu opening based on link match (if all else fails).
	EnableSubmenuTemplateMatching (see page 150)	Set to enable auto-submenu opening based on a template filename match (if all else fails).
	MasterControlId (see page 151)	If this control is a slave Flex menu, Enter the ID of the master Flex menu control. Otherwise, leave blank.
	MenuDepth (see page 151)	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 151)	Set to True to initially render menu with submenus closed
	StartLevel (see page 152)	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 152)	The Style-Sheet to use to render the Menu (see page 246) (optional).
	XmlDoc (see page 152)	XmlDocument used in this control.
	XsltDoc (see page 153)	Xslt used in this control.

Legend

	Property

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 cache interval in second(s).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("interval to cache this control in second(s)")]
[DefaultValue(600.0)]
public override double CacheInterval;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("interval to cache this control in second(s)")>
<DefaultValue(600.0)>
Public override Property CacheInterval() As double
```

FlexMenu.DisplayXslt Property

The Xslt used to render the Menu (see page 246).

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 to render the Menu (see page 246) (optional).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Style-Sheet to use to render the Menu.")]
[TypeConverter(typeof(StringConverter))]
[DefaultValue("")]
public override string Stylesheet;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Style-Sheet to use to render the Menu.")>
<TypeConverter(typeof(StringConverter))>
<DefaultValue("")>
Public override Property Stylesheet() As string
```

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 (see page 449). 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 449)" > "Managing Content (see page 449) 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 : EkWebControl;
```

Visual Basic

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

File

FolderBreadcrumb.cs

FolderBreadcrumb Methods

The methods of the FolderBreadcrumb class are listed here.

Public Methods

	Name	Description
	Fill (see page 153)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

FolderBreadcrumb.Fill Method

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

C#

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

Visual Basic

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

FolderBreadcrumb Properties

The properties of the FolderBreadcrumb class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 154)	Gets or sets a Content (see page 449) ID and displays the breadcrumb trail for the folder in which the content (see page 449) resides. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (see page 383)] > Folder (see page 383) properties > Breadcrumb tab.
	DefaultFolderID (see page 155)	The folder ID for which you want the breadcrumb trail to display. If a DefaultContentID (see page 154) 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 383)] > Folder (see page 383) properties > Breadcrumb tab.
	DisplayStyle (see page 155)	Indicate how to display the breadcrumb trail: horizontally or vertically. The default is Horizontal.
	DynamicParameter (see page 155)	Gets or Sets the QueryString parameter to read a Content (see page 449) ID dynamically - Set to String.Empty if you wish to always display the default ContentID (static.)
	LinkTarget (see page 156)	Determines the type of window that appears on this Web form when a user (see page 534) clicks an item in the breadcrumb trail. The default is _self. • **_Self** - opens link in the same window. • **_Top** - opens link in a parent window. • **_Blank** - opens link in a new window. • **_Parent** - opens link in the parent frame.
	Mode (see page 156)	Allows you to make the breadcrumb trail appear as non-hyperlinked plain text. The default is Normal. • **Normal** = breadcrumb trail is hyperlinked • **DisplayOnly** = breadcrumb trail is plain text
	separator (see page 156)	Enter one or more characters to separate the items in a breadcrumb trail.

Legend

	Property

FolderBreadcrumb.DefaultContentID Property

Gets or sets a Content (see page 449) ID and displays the breadcrumb trail for the folder in which the content (see page 449) resides. To use this property, breadcrumb information for a folder must be defined in the Workarea > [Specified Folder (see page 383)] > Folder (see page 383) 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 int 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 Integer
```

FolderBreadcrumb.DefaultFolderID Property

The folder ID for which you want the breadcrumb trail to display. If a DefaultContentID (see page 154) 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 383)] > Folder (see page 383) 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 int 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 Integer
```

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.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Content (see page 449) 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 (see page 534) 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
```

FormBlock Class

The FormBlock server control displays a content (see page 449) 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 157)	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 158)	Overrideable method that can be used to grab data from the back end for each server control.
	GetEnumerator (see page 158)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 158)	Gets the view data using a dataset.
	GetViewNames (see page 158)	Gets the view names.
	MoveNext (see page 158)	Moves up the index of the Collection. Returns false if the end of the collection has been surpassed.
	Reset (see page 159)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 159)	Index operator into a strongly typed dataset holding some common properties for a Collection item.
	ToString (see page 159)	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 159)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

FormBlock.CopyTo Method

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

C#

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

Visual Basic

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

FormBlock.Fill Method

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

C#

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

Visual Basic

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

FormBlock.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

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

Visual Basic

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

FormBlock.GetView Method

Gets the view data using a dataset.

C#

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

Visual Basic

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

Parameters

Parameters	Description
viewName	

Returns

DataSourceView

FormBlock.GetViewNames Method

Gets the view names.

C#

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

Visual Basic

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

Returns

View names

FormBlock.MoveNext Method

Moves up the index of the Collection. 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 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 160)	Set this value to true if you wish to add the validation to your main runat=server form.
V	CacheInterval (see page 160)	Sets the cache interval in second(s).
	Count (see page 161)	Indicates the number of elements in your Collection.
	Current (see page 161)	Gets the current object in the collection.
	DefaultFormID (see page 161)	Gets or Sets the Default FormID for this Control.
	DynamicParameter (see page 162)	Gets or sets the QueryString parameter to read a Form (see page 493) ID dynamically - Set to String.Empty if you wish to always display the default FormID (static.)
	EkItem (see page 162)	Gets the Ektron.Cms.Common.ContentBase for the current Form (see page 493) Block.
	Fields (see page 162)	Returns a collection of fields defined in the selected form.
	FormName (see page 162)	Set this value to the name of your main runat=server form.
	IncludeTags (see page 163)	Set this to true if you wish to have this Ektron Form (see page 493) element to wrap its own Form (see page 493) tags around itself.
	IsSynchronized (see page 163)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	StartingFolderID (see page 163)	WEB PARTS ONLY: Allows a developer to set the starting folder for the personalization/collaboration selection dialog.
	SyncRoot (see page 164)	Gets an object that can be used to synchronize the collection.

Legend

	Property
V	virtual

FormBlock.AddValidation Property

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

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Set this value to true if you wish to add the validation to your main
runat=server form.")]
[Personalizable()]
[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 if you wish 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 cache interval in second(s).

C#

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

Visual Basic

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

FormBlock.Count Property

Indicates the number of elements in your Collection.

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 int 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 Integer
```

FormBlock.DynamicParameter Property

Gets or sets the QueryString parameter to read a Form (see page 493) ID dynamically - Set to String.Empty if you wish to always display the default FormID (static.)

C#

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

Visual Basic

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

FormBlock.EkItem Property

Gets the Ektron.Cms.Common.ContentBase for the current Form (see page 493) Block.

C#

```
[Browsable(false)]
public ContentBase EkItem;
```

Visual Basic

```
<Browsable(false)>
Public Property EkItem() As ContentBase
```

FormBlock.Fields Property

Returns a collection of fields defined in the selected form.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("List of fields defined for this form.")]
public FormFieldCollection Fields;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("List of fields defined for this form.")>
Public Property Fields() As FormFieldCollection
```

FormBlock.FormName Property

Set this value to the name of your main runat=server form.

C#

```
[Bindable(true)]
[Browsable(false)]
[Category("Ektron")]
[Description("Set this value to the name of your main runat=server form.")]
[Personalizable()]
```

```
public string FormName;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<Category("Ektron")>
<Description("Set this value to the name of your main runat=server form.")>
<Personalizable()>
Public Property FormName() As string
```

FormBlock.IncludeTags Property

Set this to true if you wish to have this Ektron Form (see page 493) element to wrap its own Form (see page 493) tags around itself.

C#

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

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Whether or not to wrap form tags around this Ektron Form element.")>
<Personalizable()>
Public Property IncludeTags() As bool
```

FormBlock.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

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

Visual Basic

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

FormBlock.StartingFolderID Property

WEB PARTS ONLY: Allows a developer to set the starting folder for the personalization/collaboration selection dialog.

C#

```
[Bindable(true)]
[Category("Ektron WebParts")]
[Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")]
[DefaultValue(0)]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Personalizable()]
public int StartingFolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron WebParts")>
<Description("WEB PARTS ONLY: Allows a developer to set the starting folder for the
personalization/collaboration selection dialog.")>
<DefaultValue(0)>
```

```
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable()>
Public Property StartingFolderID() As Integer
```

FormBlock.SyncRoot Property

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

C#

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

Visual Basic

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

FormFieldItem Class

Defines a field within the form.

Class Hierarchy

Ektron.Cms.Controls.FormFieldItem

C#

```
[DefaultProperty("FieldName")]
[ValidationProperty("Value")]
[TypeConverter(typeof(FormFieldItemConverter))]
public class FormFieldItem;
```

Visual Basic

```
<DefaultProperty("FieldName")>
<ValidationProperty("Value")>
<TypeConverter(typeof(FormFieldItemConverter))>
Public Class FormFieldItem
```

File

FormField.cs

FormFieldItem Properties

The properties of the FormFieldItem class are listed here.

Public Properties

	Name	Description
	Basetype (see page 164)	Gets the basic datatype of the field.
	ContentType (see page 165)	Gets the type of content (see page 449) allowed within the field. This is loosely associated with the datatype.
	DatalistItems (see page 165)	Gets the list of options for this field's value, if applicable, otherwise null/Nothing.
	Datatype (see page 165)	Gets the datatype of the field.
	DisplayName (see page 166)	Gets the name of the field to be displayed.
	FieldName (see page 166)	Gets the name of the field.
	Language (see page 166)	Gets the language code of the datalist. For example, en or en-US.
	Value (see page 167)	Gets or set the value of this field.

Legend

	Property

FormFieldItem.Basetype Property

Gets the basic datatype of the field.

C#

```
[Category("Data")]
[Description("Basic type of the field's data value, for example, 'text'.")]
public string Basetype;
```

Visual Basic

```
<Category("Data")>
<Description("Basic type of the field's data value, for example, 'text'.")>
Public Property Basetype() As string
```

Remarks

The possible values of the basic datatype vary, including, text, textbox, calendar (see page 447), and checkbox.

FormFieldItem.ContentType Property

Gets the type of content (see page 449) allowed within the field. This is loosely associated with the datatype.

C#

```
[Category("Data")]
[Description("Type of content allowed within the field")]
public string ContentType;
```

Visual Basic

```
<Category("Data")>
<Description("Type of content allowed within the field")>
Public Property ContentType() As string
```

Remarks

The possible values for content (see page 449) 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;
```

Visual Basic

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

File

Forum.cs

Forum Methods

The methods of the Forum class are listed here.

Public Methods

	Name	Description
	Fill (see page 167)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 168)	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 168)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Forum.Fill Method

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

C#

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

Visual Basic

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

Forum.ToString Method

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

C#

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

Visual Basic

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

Forum.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

Forum Properties

The properties of the Forum class are listed here.

Public Properties

	Name	Description
	BoardID (see page 169)	The ID of the discussion board.
	CacheInterval (see page 169)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a 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 169)	Gets or Sets the QueryString parameter to read a Blog (see page 22) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 22) (static.)
	ExcludeList (see page 170)	Gets or sets the comma separated list of custom properties to exclude on the forum profile page.
	HideUserProfile (see page 170)	The control will be forced to hide users' profile.
	JavascriptEditorHeight (see page 170)	Set the height in pixels for the editor. The default is 400.
	JavascriptEditorHTMLMode (see page 171)	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 171)	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 172)	Set the width in pixels for the editor. The default is 625.

	Theme (see page 172)	The image files for a theme must be located in the same location as the stylesheet used to display that forum. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Using a Custom Theme"

Legend

	Property
	virtual

Forum.BoardID Property

The ID of the discussion board.

C#

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

Visual Basic

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

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()]
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
```

Forum.DynamicParameter Property

Gets or Sets the QueryString parameter to read a Blog (see page 22) ID dynamically. Set to String.Empty if you wish to always display the default Blog (see page 22) (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(DynamicContentIDTypeConverter))]
[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(DynamicContentIDTypeConverter))>
<DefaultValue("boardid")>
Public Property DynamicParameter() As string
```

Forum.ExcludeList Property

Gets or sets the comma separated list of custom properties to exclude on the forum profile page.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Gets or Sets the comma separated list of custom properties to exclude on the
forum profile page.")]
[DefaultValue("")]
public string ExcludeList;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Gets or Sets the comma separated list of custom properties to exclude on the
forum profile page.")>
<DefaultValue("")>
Public Property ExcludeList() As string
```

Forum.HideUserProfile Property

The control will be forced to hide users' profile.

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.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.")]
[Personalizable()]
public int JavascriptEditorHeight;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(400)>
<Description("The pixel height for the Javascript Editor.")>
<Personalizable()>
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("Determines if HTML mode is allowed for the Javascript Editor.")]
[Personalizable()]
public bool JavascriptEditorHTMLMode;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Determines if HTML mode is allowed for the Javascript Editor.")>
<Personalizable()>
Public Property JavascriptEditorHTMLMode() As bool
```

Forum.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.")]
[Personalizable()]
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.")>
<Personalizable()>
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.")]
[Personalizable()]
public int JavascriptEditorWidth;
```

Visual Basic

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

Forum.Theme Property

The image files for a theme must be located in the same location as the stylesheet used to display that forum. See the CMS400.NET Developer Manual section "Discussion Board Server Controls (see page 7)" > "Using a Custom Theme"

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("The path to custom theme files.")]
[Personalizable()]
public string Theme;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("The path to custom theme files.")>
<Personalizable()>
Public Property Theme() As string
```

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 173)	Create the javascript that handles the onready event and other CMS specific setup functionality.
	BeginJavaScript (see page 174)	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 174)	This will end the surrounding of the JavaScript created for the CMS.
S	MakeFieldNameFromId (see page 174)	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 174)	Creates the fields used when the content (see page 449) is posted. This creates fields which are posted to server with these IDs: • [ControlName]_Action = the action to take - save, check in, undo checkout • [ControlName]_ContentTitle = the content (see page 449) title • [ControlName]_ContentId = The ID of the content (see page 449) • [ControlName]_Language = The language of the content (see page 449) The content (see page 449) title caption is surrounded with a span that has this ID: • [ControlName]_TitleLabel All of the information for all the content (see page 449) information fields (except the content (see page 449) itself) is surrounded with a div tag with this ID: • [ControlName]_ContentInfo The valid values for the [ControlName]_Action value are: • Publish - Submit... more (see page 174)
	SelectedJavaScript (see page 175)	Renders any JavaScript based on options set by the developer through the properties.

Legend

	Method
S	static

HtmlEditDisplay.BaseJavaScript Method

Create the javascript that handles the onready event and other CMS specific setup functionality.

C#

```
public string BaseJavaScript(string strTranslationFile, bool bShowCMSToolbar);
```

Visual Basic

```
Public Function BaseJavaScript(strTranslationFile As string, bShowCMSToolbar As bool) As string
```

Parameters

Parameters	Description
`strTranslationFile`	Use m_strTranslationFile for this.
`bShowCMSToolbar`	Use m_bShowCMSToolbar for this.
`iContentID`	Use m_iContentId for this.
`bIsLoggedIn`	Use IsLoggedIn for this.

HtmlEditDisplay.BeginJavaScript Method

This returns the string which will surround all the JavaScript which is given in the CMSRender methods. This may contain any syntax, setup, includes, copyright message, or other options. This does not include the core JS files or the hidden field. Those are placed in by the eWebEditPro control.

C#

```
public string BeginJavaScript(int iContentId, int iLanguageID);
```

Visual Basic

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

Parameters

Parameters	Description
iContentId	Use m_iContentId for this.
iLanguageID	Use m_iLanguageID for this.
strWorkareaPath	Get using FindWorkareaLoc().

HtmlEditDisplay.EndJavaScript Method

This will end the surrounding of the JavaScript created for the CMS.

C#

```
public string EndJavaScript();
```

Visual Basic

```
Public Function EndJavaScript() As string
```

HtmlEditDisplay.MakeFieldNameFromId Method

This makes the names used by variable fields. Caution must be taken if this method is modified. The resulting names must match what the JavaScript is expecting for names. The JS is also assembling field names to set and retrieve data.

C#

```
public static string MakeFieldNameFromId(string strBaseId, string strSuffix);
```

Visual Basic

```
Public static Function MakeFieldNameFromId(strBaseId As string, strSuffix As string) As string
```

Parameters

Parameters	Description
strBaseId	The name, or ID of the editor.
strSuffix	The descriptive piece of the field name.

HtmlEditDisplay.RenderCMSFields Method

Creates the fields used when the content (see page 449) is posted. This creates fields which are posted to server with these IDs:

- [ControlName]_Action = the action to take - save, check in, undo checkout
- [ControlName]_ContentTitle = the content (see page 449) title
- [ControlName]_ContentId = The ID of the content (see page 449)
- [ControlName]_Language = The language of the content (see page 449)

The content (see page 449) title caption is surrounded with a span that has this ID:

- [ControlName]_TitleLabel

All of the information for all the content (see page 449) information fields (except the content (see page 449) itself) is surrounded with a div tag with this ID:

- [ControlName]_ContentInfo

The valid values for the [ControlName]_Action value are:

- Publish - Submit the posted content (see page 449) for publication
- CheckIn - Check in the content (see page 449)
- Save - Save the content (see page 449) and redisplay in the editor
- UndoCheckout - Undo the checkout of the content (see page 449)

C#

```
public string RenderCMSFields(string strContentTitle, string strTitleLabel, bool bShowTitle);
```

Visual Basic

```
Public Function RenderCMSFields(strContentTitle As string, strTitleLabel As string, bShowTitle
As bool) As string
```

Parameters

Parameters	Description
strContentTitle	The title value text.
strTitleLabel	Use RetrieveMessage(m_enumMessages.emsgContentTitle) for this.
bShowTitle	Use m_bShowTitle for this.

HtmlEditDisplay.SelectedJavaScript Method

Renders any JavaScript based on options set by the developer through the properties.

C#

```
public string SelectedJavaScript(int iSysDefaultLangID);
```

Visual Basic

```
Public Function SelectedJavaScript(iSysDefaultLangID As Integer) As string
```

HtmlEditor Class

This server control provides the ability to edit specific CMS content (see page 449). The eWebEditPro editor can easily be placed onto a page for content (see page 449) editing. There is flexibility to control the toolbars, the available functionality, and the operations of the editor.

The control, at run time, produces the fields listed below. They are posted with the content (see page 449) and can be processed in the receiving page.

- [ControlName]_Action = the action to take - save, check in, undo checkout
- [ControlName]_ContentTitle = the content (see page 449) title
- [ControlName]_ContentId = the ID of the content (see page 449)
- [ControlName]_Language = the language of the content (see page 449)

The content (see page 449) title caption is surrounded with a span that has this ID: [ControlName]_TitleLabel.

All of the information for all the content (see page 449) information fields (except the content (see page 449) 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 181)

- [ControlName]_ContentTitle read with .ContentTitle (see page 182)
- [ControlName]_ContentId read with .ContentId
- [ControlName]_Language read with .ContentLanguage
- [ControlName]_SafeEncode read with .SafeEncode (see page 187)

The content (see page 449) sent or posted is retrieved with the simple: .Text (see page 189)

The valid values for the [ControlName]_Action value are:

- Publish - Submit the posted content (see page 449) for publication
- CheckIn - Check in the content (see page 449)
- Save - Save the content (see page 449) and redisplay in the editor
- UndoCheckout - Undo the checkout of the content (see page 449)

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 (see page 449) editor. There are no parameters. The control depends on the settings given at design time or at run time.

C#

```
public HtmlEditor();
```

Visual Basic

```
Public Sub HtmlEditor()
```

HtmlEditor Methods

The methods of the HtmlEditor class are listed here.

Public Methods

	Name	Description
	ExecuteContentAction (see page 177)	This uses the ContentAction (see page 181) value and performs the specified action on the content (see page 449) contained within the server control's posted values. The content (see page 449) and the other property values are placed into the CMS400 according to the Action value. The possible action values are: • "Submit" • "CheckIn" • "Save" • "UndoCheckout" (cancel or undo)
	ToString (see page 177)	Returns the string which would be produced when the server control is rendered in the browser.
	WorkareaMode (see page 178)	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 (see page 449).

Legend

	Method

HtmlEditor.ExecuteContentAction Method

This uses the ContentAction (see page 181) value and performs the specified action on the content (see page 449) contained within the server control's posted values. The content (see page 449) 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 (see page 449).

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 179)	This signifies that the selected content (see page 449) is archived.
	Authenticated (see page 180)	DESIGN TIME ONLY: Used to indicate to a user (see page 534) that they are logged in, and permits them to login through a login dialog in VisualStudio.NET.
	CmsSitePath (see page 180)	DESIGN TIME ONLY: Used to find the Web service required to access the CMS in Design Time.
	ContentAction (see page 181)	The action to perform on the posted content (see page 449). 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 181)	Specifies the content (see page 449) ID to load. This value must match a value that exists in the CMS database. The selection of content (see page 449) is similar to that in the ContentBlock (see page 78) control. The content (see page 449) can only be accessed if the user (see page 534) 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 (see page 449) and setting the content (see page 449) at edit time by placing it into the .Text (see page 189) property for the... more (see page 181)
	ContentTitle (see page 182)	The title for the selected Content (see page 449). This is set when new content (see page 449) 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 (see page 534) can change it there.
	ContentType (see page 182)	This value describes the type of content (see page 449) to which the selected ID refers. If this value is blank, then the content (see page 449) is a custom type, to be defined type, or unsupported type. These are the possible types: • HTML • Form (see page 493)
	EditorReadyCall (see page 183)	This specifies which JavaScript function to run when the HTML editor is ready to start processing commands. This happens after the initialization of the editor and the loading of the initial content (see page 449). 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 183)	If 'true', the user (see page 534) can upload locally selected images and linked files that they have put into their content (see page 449). If enabled, the automatic upload mechanism will allow uploading of the local files automatically with minimal user (see page 534) intervention. If 'false', the site developer must provide an upload page, such as the CMS library, where the upload can occur.
	IncludeContentInfo (see page 184)	If true, includes the information about the content (see page 449) 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 (see page 534), then define the class [ControlName]_TitleLabel as having no display, or set the ShowTitle (see page 188) property to false.
	IsLoggedIn (see page 184)	DESIGN TIME ONLY: Read only property that lets a developer know if the user (see page 534) is currently log into the site as a user (see page 534).
	Language (see page 185)	Specifies the ID of the language to use for the content (see page 449) to load. This value must match a value that exists in the CMS database.

	loggedInUserID (see page 185)	DESIGN TIME ONLY: Read only property that lets a developer know the user (see page 534) ID of the user (see page 534) who is currently logged in.
	loggedInUserName (see page 186)	DESIGN TIME ONLY: Read only property that lets a developer know the user (see page 534) name of the user (see page 534) who is currently logged in.
	Path (see page 186)	The path to the site that contains this control. This is not available in the properties. It is meant for code behind which wants to override the path used by the CMS. The path is automatically determined by the control at run time, if it is not set by the code behind. This is only available from Code Behind.
	PreferredType (see page 186)	The display to use for viewing the content (see page 449). This can be used to simplify the display or to debug the operation of editing content (see page 449) on the site. The options selectable are: • "activex" = Use the ActiveX control • "textarea" = Show HTML in text area The default is "activex". This is only available from Code Behind.
	SafeEncode (see page 187)	If true then the posted HTML is encoded using URL encoding. This makes the transfer secure. If the content (see page 449) is encoded then it must be decoded on the receiving side. If the decode process is not wanted, then set this value to false. This will require the validateRequest attribute in the page definition be set to false.
	ShowCMSToolbar (see page 188)	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 188)	If 'true', the title of the selected content (see page 449) is displayed. The user (see page 534) can then edit the title. The submission of content (see page 449) posts up the title, even if it is not displayed to the user (see page 534), unless the IncludeContentInfo (see page 184) property is set to a false value.
	StyleSheet (see page 188)	Assigns a single stylesheet.
	Text (see page 189)	The HTML text being edited.
	ToolbarLevel (see page 189)	This specifies the level of the toolbar feature set offered to the user (see page 534). 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 (see page 534). These are the value values: • "None" • "Reduced" • "Default" • "DataDesigner" • "DataEntry" • "Calendar (see page 64)" • "FormDesign" • "XSLTDesign" • "Full" • "Custom"
	ToolbarResetCall (see page 190)	This specifies which JavaScript function to run when the toolbar resets. The toolbar reset offers the opportunity for the script to affect the operation of the toolbar with actions such as adding or removing commands, disabling commands, or creating entirely new toolbars. This is the name, not the full function definition, of a JavaScript function. The function must follow the format defined in the eWebEditPro documentation for this type of method. Below is an example:
	TranslationFile (see page 191)	The translation file (or stream) that contains the XML elements for the translation of strings used in the editor. If this is blank, the client system's windows setting is used to select form the series of default translation files.

Legend

	Property

HtmlEditor.Archived Property

This signifies that the selected content (see page 449) 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 (see page 534) 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 (see page 449). 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 (see page 449) ID to load. This value must match a value that exists in the CMS database. The selection of content (see page 449) is similar to that in the ContentBlock (see page 78) control. The content (see page 449) can only be accessed if the user (see page 534) 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 (see page 449) and setting the content (see page 449) at edit time by placing it into the .Text (see page 189) 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 (see page 449).

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 int 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 Integer
```

HtmlEditor.ContentTitle Property

The title for the selected Content (see page 449). This is set when new content (see page 449) 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 (see page 534) can change it there.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The title of the content.")]
public string ContentTitle;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The title of the content.")>
Public Property ContentTitle() As string
```

HtmlEditor.ContentType Property

This value describes the type of content (see page 449) to which the selected ID refers. If this value is blank, then the content (see page 449) is a custom type, to be defined type, or unsupported type. These are the possible types:

- HTML
- Form (see page 493)

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The type of content selected.")]
public string ContentType;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The type of content selected.")>
Public Property ContentType() As string
```

HtmlEditor.EditorReadyCall Property

This specifies which JavaScript function to run when the HTML editor is ready to start processing commands. This happens after the initialization of the editor and the loading of the initial content (see page 449).

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 (see page 534) can upload locally selected images and linked files that they have put into their content (see page 449). If enabled, the automatic upload mechanism will allow uploading of the local files automatically with minimal user (see page 534) 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 (see page 449) 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 (see page 534), then define the class [ControlName]_TitleLabel as having no display, or set the ShowTitle (see page 188) 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 (see page 534) is currently log into the site as a user (see page 534).

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 (see page 449) 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 (see page 534) ID of the user (see page 534) 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 int loggedInUserID;
```

Visual Basic

```
<Browsable(true)>
<DesignOnly(true)>
<Bindable(true)>
<Category("Ektron Authentication")>
<DefaultValue(0)>
<Description("DESIGN TIME ONLY: Read only property which displays the validated user ID for
the CMS site.")>
Public Property loggedInUserID() As Integer
```

HtmlEditor.loggedInUserName Property

DESIGN TIME ONLY: Read only property that lets a developer know the user (see page 534) name of the user (see page 534) 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. It is meant for code behind which wants to override the path used by the CMS.

The path is automatically determined by the control at run time, if it is not set by the code behind.

This is only available from Code Behind.

C#

```
[Bindable(true)]
[Browsable(false)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[DefaultValue("")]
[Description("The path to the site that contains this control.")]
public new string Path;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<DefaultValue("")>
<Description("The path to the site that contains this control.")>
Public new Property Path() As string
```

HtmlEditor.PreferredType Property

The display to use for viewing the content (see page 449). This can be used to simplify the display or to debug the operation of editing content (see page 449) on the site.

The options selectable are:

- "activex" = Use the ActiveX control
- "textarea" = Show HTML in text area

The default is "activex". This is only available from Code Behind.

C#

```
[Bindable(true)]
[Browsable(false)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(true)]
[DefaultValue("")]
[Description("The path to the site that contains this control.")]
public new string PreferredType;
```

Visual Basic

```
<Bindable(true)>
<Browsable(false)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(true)>
<DefaultValue("")>
<Description("The path to the site that contains this control.")>
Public new Property PreferredType() As string
```

HtmlEditor.SafeEncode Property

If true then the posted HTML is encoded using URL encoding. This makes the transfer secure. If the content (see page 449) 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("Whether to encode the HTML content.")]
public bool SafeEncode;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue(true)>
<Description("Whether to encode the HTML content.")>
Public Property SafeEncode() As bool
```

Example 1

validateRequest="false"

Example 2

@Page Language (see page 185)="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 (see page 449) is displayed. The user (see page 534) can then edit the title. The submission of content (see page 449) posts up the title, even if it is not displayed to the user (see page 534), unless the IncludeContentInfo (see page 184) property is set to a false value.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue(true)]
[Description("Show the content title for edit.")]
public bool ShowTitle;
```

Visual Basic

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

HtmlEditor.StyleSheet Property

Assigns a single stylesheet.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The stylesheet to use for the edited content.")]
public new string StyleSheet;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The stylesheet to use for the edited content.")>
Public new Property StyleSheet() As string
```

HtmlEditor.Text Property

The HTML text being edited.

C#

```
[Bindable(true)]
[Browsable(true)]
[DesignOnly(false)]
[Category("Ektron")]
[NotifyParentProperty(true)]
[ParenthesizePropertyName(false)]
[DefaultValue("")]
[Description("The HTML text.")]
public new string Text;
```

Visual Basic

```
<Bindable(true)>
<Browsable(true)>
<DesignOnly(false)>
<Category("Ektron")>
<NotifyParentProperty(true)>
<ParenthesizePropertyName(false)>
<DefaultValue("")>
<Description("The HTML text.")>
Public new Property Text() As string
```

HtmlEditor.ToolbarLevel Property

This specifies the level of the toolbar feature set offered to the user (see page 534). 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 (see page 534).

These are the value values:

- "None"
- "Reduced"
- "Default"
- "DataDesigner"
- "DataEntry"

- "Calendar (see page 64)"
- "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 (see page 534) 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 (see page 534) 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 (see page 534) 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 (see page 534) 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 192)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 192)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

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 192)	Gets or Sets the Default ImageID for this Control.
	DynamicParameter (see page 193)	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 193)	The ID of the folder you wish to select data from. Root is 0.

Legend

	Property

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

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()]
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()>
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))]
public int FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
```

```
Public Property FolderID() As Integer
```

IndexSearch Class

Use the IndexSearch server control to display a Smart Form (see page 493)'s search (see page 514) screen on any Web form. This lets a user (see page 534) search (see page 514) 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 IndexSearch()
```

IndexSearch Structures

The structures of the IndexSearch class are listed here.

Structures

	Name	Description
	SearchPram (see page 194)	parameter structure used for search (see page 514).

Legend

	Structure

Ektron.Cms.Controls.IndexSearch.SearchPram Structure

parameter structure used for search (see page 514).

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 195)	
	CopyTo (see page 196)	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 196)	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 196)	Gets a System.Collections.IEnumerator for the collection.
	GetSearchParmDataType (see page 196)	Returns the search (see page 514) parameter data type.
	GetSearchParmSearchType (see page 197)	Returns the search (see page 514) parameter search (see page 514) type.
	GetXPathValues (see page 197)	Returns array of strings using Xpaths.
	MoveNext (see page 197)	Moves up the index of the Collection. Returns false if the end of the collection has been surpassed.
	Reset (see page 198)	Brings the index of the collection back to zero. Does not clear the collection.
	Search (see page 198)	
	this (see page 198)	Index operator into a strongly typed dataset holding some common properties for a Collection item.
	Wrap (see page 198)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some HTML text.

Legend

	Method

AddParm Method

Overload List

	Name	Description
	IndexSearch.AddParm () (see page 195)	Adds a parameter to the search (see page 514).
	IndexSearch.AddParm (SearchPram) (see page 196)	Adds parameter.

IndexSearch.AddParm Method ()

Adds a parameter to the search (see page 514).

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 194)

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 (see page 514) 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 (see page 514) parameter search (see page 514) type.

C#

```
public EkEnumeration.XMLSearchRangeType GetSearchParmSearchType(string val);
```

Visual Basic

```
Public Function GetSearchParmSearchType(val As string) As EkEnumeration.XMLSearchRangeType
```

Parameters

Parameters	Description
val	search (see page 514) 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. Returns false if the end of the collection has been surpassed.

C#

```
public bool MoveNext();
```

Visual Basic

```
Public Function MoveNext() As bool
```

Returns

Boolean indicating if the move was within the range of the array.

IndexSearch.Reset Method

Brings the index of the collection back to zero. Does not clear the collection.

C#

```
public void Reset();
```

Visual Basic

```
Public Function Reset() As void
```

Search Method

Overload List

	Name	Description
	IndexSearch.Search () (see page 198)	Performs the search action.
	IndexSearch.Search (System.Web.HttpRequest) (see page 198)	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 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 200)	If you want to display an image on the submit button, enter the server path to it. (Optional)
	ButtonText (see page 200)	The text that appears on the button.
	ContentParameter (see page 201)	Checks the QueryString for this value and replaces the collection with that content (see page 449) block when specified. Leave blank to always display the collection.
	Count (see page 201)	Indicates the number of elements in your Collection.
	Current (see page 201)	Gets the current object in the collection.
	EkItems (see page 201)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection. Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 202)	The message that displays if the search (see page 514) returns no hits. (Optional)
	EncodeContentHtml (see page 202)	Set the property to true if you wish to html encode the Content (see page 449) Html in returned result."
	FolderId (see page 202)	The ID of the folder where to start the search (see page 514). Root starts at "0".
	FormName (see page 203)	The form name.
	HitsMarker (see page 203)	This symbol shows how many search (see page 514) criteria each hit matched
	IsSynchronized (see page 203)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	Labelafter (see page 203)	Specify a label for the date field where search (see page 514) criteria is "After".
	Labelbefore (see page 204)	Specify a label for the date field where search (see page 514) criteria is "Before".
	LabelBetween (see page 204)	Specify label for numeric field where search (see page 514) criteria is based on range of values.
	LabelContains (see page 204)	Specify the label for a string field where search (see page 514) criteria "Contains".
	LabeldateBetween (see page 205)	Specify a label for the date field where search (see page 514) criteria based on a range of dates.
	LabelEqual (see page 205)	Specify label for numeric field where search (see page 514) criteria is "Equal".
	LabelExactPhrase (see page 205)	Specify a label for the string field where search (see page 514) criteria "Exact Phrase".
	LabelGreaterThan (see page 205)	Specify label for numeric field where search (see page 514) criteria is "Greater Than".
	Labelinputdate (see page 206)	Specify a label for the message that appears when date entered are blank/incorrect.
	Labelinputnumber (see page 206)	Specify a label for the message that appears when values entered are blank/incorrect.
	LabelLessThan (see page 206)	Specify label for numeric field where search (see page 514) criteria "Less Than".
	LabelOn (see page 207)	Specify a label for the Date field where search (see page 514) criteria is "On".
	LabelSelect (see page 207)	Specify labels for first item on selection dropdown.

	LinkTarget (see page 207)	Gets or sets the target window for the result to be displayed.
	MaxResults (see page 208)	The total number of results to display. If set to zero, unlimited results will be set.
	OrderBy (see page 208)	Specify the order of the results. Choices are: • **Title** - the title of the content (see page 449) block • **ID** - the XML content (see page 449) block ID number • **Date Created** - the date the content (see page 449) block was created • **Date Modified** - the date the content (see page 449) block was last modified • **AuthorLName** - the last author's last name • **AuthorFname** - the last author's first name • **StartDate** - the GoLive date
	OrderByDirection (see page 208)	The direction the results will be sorted in. This property works with the OrderBy (see page 208) property only. Choices are: • Ascending • Descending
	Recursive (see page 209)	Select true if you wish to search (see page 514) the subfolders of the selected FolderPath.
	SearchParmXML (see page 209)	Predefine a set of search (see page 514) criteria.
	ShowSearchBoxAlways (see page 209)	If this property is set to false, the search (see page 514) box will not be visible on PostBack.
	SyncRoot (see page 210)	Gets an object that can be used to synchronize the collection.
	Weighted (see page 210)	Select true if you wish to search (see page 514) the subfolders of the selected FolderPath.
	XmlConfigId (see page 210)	ID of the Smart Form (see page 493) whose indexed fields will appear on the search (see page 514) screen.
	XmlDoc (see page 210)	Returns the XSLT object for the collection.

Legend

	Property

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 button.")]
public string ButtonImgSrc;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: The path to the image you wish to use for the button.")>
Public Property ButtonImgSrc() As string
```

IndexSearch.ButtonText Property

The text that appears on the button.

C#

```
[Category("Ektron")]
[Description("The text of the search button.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_BTN)]
public string ButtonText;
```

Visual Basic

```
<Category("Ektron")>
<Description("The text of the search button.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_BTN)>
Public Property ButtonText() As string
```

IndexSearch.ContentParameter Property

Checks the QueryString for this value and replaces the collection with that content (see page 449) block when specified. Leave blank to always display the collection.

C#

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

Visual Basic

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

IndexSearch.Count Property

Indicates the number of elements in your Collection.

C#

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

Visual Basic

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

IndexSearch.Current Property

Gets the current object in the collection.

C#

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

Visual Basic

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

IndexSearch.EkItems Property

Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection. 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 displays if the search (see page 514) 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 449) 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 (see page 514). 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 int FolderId;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Description("The id of the folder you wish to start the search. Root starts at \"0\".")>
<Editor(typeof(GetFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
```

```
Public Property FolderId() As Integer
```

IndexSearch.FormName Property

The form name.

C#

```
[Bindable(false), Browsable(false)]
[Description("")]
public string FormName;
```

Visual Basic

```
<Bindable(false), Browsable(false)>
<Description("")>
Public Property FormName() As string
```

IndexSearch.HitsMarker Property

This symbol shows how many search (see page 514) criteria each hit matched

C#

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

Visual Basic

```
<Category("Ektron"), TypeConverter(typeof(HitsMarkerTypeConverter)), DefaultValue("*")>
<Description("The symbol shows how many search criteria each hit matched")>
Public Property HitsMarker() As string
```

IndexSearch.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

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

Visual Basic

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

IndexSearch.Labelafter Property

Specify a label for the date field where search (see page 514) criteria is "After".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for date field where search criteria is \"After\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_AFTER)]
public string Labelafter;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for date field where search criteria is \"After\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_AFTER)>
Public Property Labelafter() As string
```

IndexSearch.Labelbefore Property

Specify a label for the date field where search (see page 514) criteria is "Before".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for date field where search criteria is \"Before\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_BFORE)]
public string Labelbefore;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for date field where search criteria is \"Before\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_BFORE)>
Public Property Labelbefore() As string
```

IndexSearch.LabelBetween Property

Specify label for numeric field where search (see page 514) 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 (see page 514) criteria "Contains".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for string field where search criteria \"Contains\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_CONTS)]
public string LabelContains;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for string field where search criteria \"Contains\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_CONTS)>
Public Property LabelContains() As string
```

IndexSearch.LabeldateBetween Property

Specify a label for the date field where search (see page 514) criteria based on a range of dates.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for date field where search criteria based on range of dates.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_DBTWEEN)]
public string LabeldateBetween;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for date field where search criteria based on range of dates.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_DBTWEEN)>
Public Property LabeldateBetween() As string
```

IndexSearch.LabelEqual Property

Specify label for numeric field where search (see page 514) 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 (see page 514) criteria "Exact Phrase".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for string field where search criteria \"Exact Phrase\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_XACT)]
public string LabelExactPhrase;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for string field where search criteria \"Exact Phrase\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_XACT)>
Public Property LabelExactPhrase() As string
```

IndexSearch.LabelGreaterThan Property

Specify label for numeric field where search (see page 514) criteria is "Greater Than".

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for numeric field where search criteria is \"Greater Than\".")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_GRTER)]
public string LabelGreaterThan;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for numeric field where search criteria is \"Greater Than\".")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_GRTER)>
Public Property LabelGreaterThan() As string
```

IndexSearch.Labelinputdate Property

Specify a label for the message that appears when date entered are blank/incorrect.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for message that appears when date entered are blank/incorrect.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_IPUTDT)]
public string Labelinputdate;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for message that appears when date entered are blank/incorrect.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_IPUTDT)>
Public Property Labelinputdate() As string
```

IndexSearch.Labelinputnumber Property

Specify a label for the message that appears when values entered are blank/incorrect.

C#

```
[Bindable(false)]
[Category("Ektron")]
[Description("Specify label for message that appears when values entered are
blank/incorrect.")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_ISRCH_IPUTNUM)]
public string Labelinputnumber;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<Description("Specify label for message that appears when values entered are
blank/incorrect.")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_ISRCH_IPUTNUM)>
Public Property Labelinputnumber() As string
```

IndexSearch.LabelLessThan Property

Specify label for numeric field where search (see page 514) 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 (see page 514) 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 (see page 449) block
- **ID** - the XML content (see page 449) block ID number
- **Date Created** - the date the content (see page 449) block was created
- **Date Modified** - the date the content (see page 449) 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.")]
public string OrderBy;
```

Visual Basic

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

IndexSearch.OrderByDirection Property

The direction the results will be sorted in. This property works with the OrderBy (see page 208) 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 (see page 514) 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 (see page 514) 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 (see page 514) box will not be visible on PostBack.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("If this property is set to false, the search box will not be visible on
PostBack.")]
public bool ShowSearchBoxAlways;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("If this property is set to false, the search box will not be visible on
PostBack.")>
Public Property ShowSearchBoxAlways() As bool
```

IndexSearch.SyncRoot Property

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

C#

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

Visual Basic

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

IndexSearch.Weighted Property

Select true if you wish to search (see page 514) the subfolders of the selected FolderPath.

C#

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

Visual Basic

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

IndexSearch.XmlConfigId Property

ID of the Smart Form (see page 493) whose indexed fields will appear on the search (see page 514) screen.

C#

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

Visual Basic

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

IndexSearch.XmlDoc Property

Returns the XSLT object for the collection.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

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

LanguageAPI Class

Control that can be used to override the default system language assignment behavior.

Class Hierarchy

C#

```
public class LanguageAPI : WebControl;
```

Visual Basic

```
Public Class LanguageAPI
Inherits WebControl
```

File

LanguageAPI.cs

LanguageAPI.LanguageAPI Constructor

Upon initialization, SiteLanguage (see page 215) is set to DefaultLanguage (see page 213).

C#

```
public LanguageAPI();
```

Visual Basic

```
Public Sub LanguageAPI()
```

LanguageAPI Methods

The methods of the LanguageAPI class are listed here.

Public Methods

	Name	Description
	Fill (see page 211)	Use this function along with LanguageLogic to implement your custom language control logic.
	GetLanguage (see page 211)	Returns a Language string corresponding to a given LanguageID.
	GetLanguageID (see page 212)	Returns the LanguageID corresponding to a given Language string.
	IsValid (see page 212)	

Legend

	Method
	virtual

LanguageAPI.Fill Method

Use this function along with LanguageLogic to implement your custom language control logic.

C#

```
public virtual void Fill();
```

Visual Basic

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

LanguageAPI.GetLanguage Method

Returns a Language string corresponding to a given LanguageID.

C#

```
public string GetLanguage(int LanguageID);
```

Visual Basic

```
Public Function GetLanguage(LanguageID As Integer) As string
```

LanguageAPI.GetLanguageID Method

Returns the LanguageID corresponding to a given Language string.

C#

```
public int GetLanguageID(string Language);
```

Visual Basic

```
Public Function GetLanguageID(Language As string) As Integer
```

IsValid Method

Overload List

	Name	Description
	LanguageAPI.IsValid (int) (see page 212)	Returns true if the passed LanguageID is an active system language. This function is overloaded. See Also: LanguageAPI.IsValid Method (string) (see page 212)
	LanguageAPI.IsValid (string) (see page 212)	Returns true if the passed Language is an active system language. This function is overloaded. See Also: LanguageAPI.IsValid Method (int) (see page 212)

LanguageAPI.IsValid Method (int)

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

This function is overloaded. See Also: LanguageAPI.IsValid Method (string) (see page 212)

C#

```
public bool IsValid(int LanguageID);
```

Visual Basic

```
Public Function IsValid(LanguageID As Integer) As bool
```

LanguageAPI.IsValid Method (string)

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

This function is overloaded. See Also: LanguageAPI.IsValid Method (int) (see page 212)

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 213)	Returns the current language (from a cookie).
	CurrentLanguageID (see page 213)	Returns the current language ID (from a cookie).
	DefaultLanguage (see page 213)	If not overridden, the system string "Default" is used.
	DefaultLanguageID (see page 214)	If not overridden, the ID '0' is used.
	LangaugeIdList (see page 214)	Returns a list of all the language IDs that are activated in Ektron CMS400.NET
	LangaugeTitleList (see page 214)	Returns a list of all the languages that are activated in Ektron CMS400.NET.
	MultiLanguageEnabled (see page 214)	Tells if the site supports multi-language mode. • **True** = Multi-language enabled • **False** = Multi-language not enabled
	OverrideDefault (see page 214)	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 215) is not equal to DefaultLanguage (see page 213). If they are equal, false is returned.
	SiteLanguage (see page 215)	Sets the site language of the site. Runs through the IsValid (see page 212) function to verify that the language is an active language in the system.
	SiteLanguageID (see page 215)	If not overridden, the ID '0' is used.

Legend

	Property

LanguageAPI.CurrentLanguage Property

Returns the current language (from a cookie).

C#

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

Visual Basic

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

LanguageAPI.CurrentLanguageID Property

Returns the current language ID (from a cookie).

C#

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

Visual Basic

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

LanguageAPI.DefaultLanguage Property

If not overridden, the system string "Default" is used.

C#

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

Visual Basic

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

LanguageAPI.DefaultLanguageID Property

If not overridden, the ID '0' is used.

C#

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

Visual Basic

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

LanguageAPI.LangaugeIdList Property

Returns 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 215) is not equal to DefaultLanguage (see page 213). 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 212) 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 142) 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 216)	Method that can be overridden and used to grab data from the back end for each server control.
	Wrap (see page 216)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

LanguageSelect.Fill Method

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

C#

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

Visual Basic

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

LanguageSelect.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar.

LanguageSelect Properties

The properties of the LanguageSelect class are listed here.

Public Properties

	Name	Description
	LabelName (see page 217)	Gets or sets the label next to the language select dropdown box.
	Text (see page 217)	Allows a developer to treat the server control as a Label, taking information from the Control and painting it on top of the Control for display. Leave blank for default display.

Legend

	Property

LanguageSelect.LabelName Property

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 218)	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 218)	Populates the data for the ListSummary (see page 217).
	GetEnumerator (see page 218)	Gets a System.Collections.IEnumerator for the ListSummary (see page 217).
	GetRssFeed (see page 219)	This method returns the ListSummary (see page 217) RSS feed.
	GetView (see page 219)	Gets the view data using dataset.
	GetViewNames (see page 219)	Gets the view names.
	MoveNext (see page 219)	Move to next index.
	Reset (see page 220)	Resets the index.
	this (see page 220)	Returns the current ContentItem.ContentRow.
	ToString (see page 220)	Overrides the system ToString function.
	Wrap (see page 220)	Wrap the Html toolbar.

Legend

	Method

ListSummary.CopyTo Method

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

C#

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

Visual Basic

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

ListSummary.Fill Method

Populates the data for the ListSummary (see page 217).

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 217).

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 217) 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 449) 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 222)	Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 449)' could be changed to 'Add Report')
	CacheInterval (see page 222)	Sets the amount of time the server control's data is cached. The default is 0 (zero). This is the amount of time, in seconds, a 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 223)	Checks the QueryString for this value and replaces the list summary with that content (see page 449) block when specified. Leave blank to always display the list summary.
	ContentType (see page 223)	Determines the type for content (see page 449) to return. For example, Forms. Choices are: • **AllTypes** - displays all of the content (see page 449) types for the given folder. • **Content (see page 449)** - displays a list of content (see page 449) blocks. • **Forms** - forms appear in the list summary. • **Archive_Content** - archived content (see page 449) blocks appear in the list summary. • **Archive_Forms** - archived forms appear in the list summary. • **Assets** - assets, such as offices documents, appear in the list summary. • **Archive_Assets** - archived assets appear in the list summary. • **LibraryItem** - library items appear in the list summary. • **Multimedia** - multimedia items appear in the list summary. • **Archive_Media** -... more (see page 223)
	Count (see page 224)	Indicates the number of elements in your ListSummary (see page 217).
	Current (see page 224)	Gets the current index.
	Direction (see page 224)	Direction for sorting results, "ascending" or "descending".
	EkItems (see page 224)	Returns the array of ContentBase.
	EnablePaging (see page 225)	Enables paging if the total records exceeds MaxResult value.
	FolderDescription (see page 225)	Returns a description of the given folder ID.
	FolderID (see page 225)	The ID of the folder you wish to select data from. Root is 0.
	FolderName (see page 225)	Returns name of the given folder ID.
	FolderPath (see page 226)	The Path of the folder from which to select data.
	GetHtml (see page 226)	If set to true, retrieves the content (see page 449) (Html body) of every content (see page 449) block in the collection for display.
	IncludeIcons (see page 226)	Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.
	IsSynchronized (see page 227)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LinkTarget (see page 227)	Gets or Sets the target window for the result.
	MarkupLanguage (see page 227)	Enter name of a markup file that manages the display of this server control.
	MaxResults (see page 227)	Maximum number of results to be returned.
	OrderKey (see page 228)	Sort order key.
	Random (see page 228)	If set to true, grabs a single content (see page 449) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.

	Recursive (see page 228)	Determines if subfolders of the selected folders should be brought into the list.
	SuppressWrapperTags (see page 229)	Suppresses the output of span/div tags around the control.
	SyncRoot (see page 229)	Gets an object that can be used to synchronize the ListSummary (see page 217).
	XmlDoc (see page 229)	Returns the Xml object loaded for ListSummary (see page 217).

Legend

	Property
	virtual

ListSummary.AddText Property

Allows overriding the derived controls default Add-menu button text (i.e. 'Add HTML Content (see page 449)' 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()]
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
```

ListSummary.ContentParameter Property

Checks the QueryString for this value and replaces the list summary with that content (see page 449) 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()]
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()>
Public Property ContentParameter() As string
```

ListSummary.ContentType Property

Determines the type for content (see page 449) to return. For example, Forms. Choices are:

- **AllTypes** - displays all of the content (see page 449) types for the given folder.
- **Content (see page 449)** - displays a list of content (see page 449) blocks.
- **Forms** - forms appear in the list summary.
- **Archive_Content** - archived content (see page 449) 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 (see page 449) 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()]
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()>
Public Property ContentType() As Ektron.Cms.Controls.CmsWebService.CMSContentType
```

ListSummary.Count Property

Indicates the number of elements in your ListSummary (see page 217).

C#

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

Visual Basic

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

ListSummary.Current Property

Gets the current index.

C#

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

Visual Basic

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

ListSummary.Direction Property

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

C#

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

Visual Basic

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

ListSummary.EkItems Property

Returns the array of ContentBase.

C#

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

Visual Basic

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

ListSummary.EnablePaging Property

Enables paging if the total records exceeds MaxResult value.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enables paging if total records exceeds MaxResult value. Use this property with
non zero value of Maxresults.")]
[Personalizable()]
public bool EnablePaging;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enables paging if total records exceeds MaxResult value. Use this property with
non zero value of Maxresults.")>
<Personalizable()>
Public Property EnablePaging() As bool
```

ListSummary.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 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()]
public int FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Personalizable()>
Public Property FolderID() As Integer
```

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.

C#

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

Visual Basic

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

ListSummary.GetHtml Property

If set to true, retrieves the content (see page 449) (Html body) of every content (see page 449) block in the collection for display.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")]
[Personalizable()]
public bool GetHtml;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("If set to true, retrieves the content (Html body) of every content block in the
collection for display.")>
<Personalizable()>
Public Property GetHtml() As bool
```

ListSummary.IncludeIcons Property

Determines whether or not to display an icon next to your navigation lists. NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.

C#

```
[Browsable(true)]
[Bindable(true)]
[Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")]
[DefaultValue(false)]
[Category("Ektron")]
```

```
[Personalizable()]
public bool IncludeIcons;
```

Visual Basic

```
<Browsable(true)>
<Bindable(true)>
<Description("Determines whether or not to display an icon next to your navigation lists.
NOTE: ONLY WORKS WITH ecmSummary AND ecmTeaser.")>
<DefaultValue(false)>
<Category("Ektron")>
<Personalizable()>
Public Property IncludeIcons() As bool
```

ListSummary.IsSynchronized Property

Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

C#

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

Visual Basic

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

ListSummary.LinkTarget Property

Gets or Sets the target window for the result.

C#

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

Visual Basic

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

ListSummary.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
```

ListSummary.MaxResults Property

Maximum number of results to be returned.

C#

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

```
public int MaxResults;
```

Visual Basic

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

ListSummary.OrderKey Property

Sort order key.

C#

```
[Bindable(true)]
[DefaultValue(Ektron.Cms.Controls.CmsWebService.TeasersOrderBy.Title)]
[Description("The key you wish to sort your results by.")]
[Category("Ektron")]
[Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")]
public Ektron.Cms.Controls.CmsWebService.TeasersOrderBy OrderKey;
```

Visual Basic

```
<Bindable(true)>
<DefaultValue(Ektron.Cms.Controls.CmsWebService.TeasersOrderBy.Title)>
<Description("The key you wish to sort your results by.")>
<Category("Ektron")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Order"), WebDescription("Order")>
Public Property OrderKey() As Ektron.Cms.Controls.CmsWebService.TeasersOrderBy
```

ListSummary.Random Property

If set to true, grabs a single content (see page 449) block randomly from the folder (and subfolders, if recursive.) Set to false for all data in the collection.

C#

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

Visual Basic

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

ListSummary.Recursive Property

Determines if subfolders of the selected folders should be brought into the list.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(false)]
[Description("Set Recursive to true if you wish to grab results in subfolders of your
FolderPath.")]
```

```
[Personalizable(), WebBrowsable(true), WebDisplayName("Recursive"), WebDescription("Recursive:
")]
public bool Recursive;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(false)>
<Description("Set Recursive to true if you wish to grab results in subfolders of your
FolderPath.")>
<Personalizable(), WebBrowsable(true), WebDisplayName("Recursive"), WebDescription("Recursive:
")>
Public Property Recursive() As bool
```

ListSummary.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 217).

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 217).

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

Legend

	Method

Login.Fill Method

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

C#

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

Visual Basic

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

Login Properties

The properties of the Login class are listed here.

Public Properties

	Name	Description
	AutoAddType (see page 230)	Set this property to define auto-added Active Directory (see page 118) users as members or authors.
	AutoLogin (see page 231)	Set this property to true if you want to use Auto Login (see page 229) with Active Directory (see page 118).
	OnlyAllowMemberLogin (see page 231)	Set this property to true if you only allow membership users.
	Password (see page 231)	Write Only value for passing in the password.
	PromptLogout (see page 231)	Set this property to true if you only allow membership users.
	SuppressHelpButton (see page 231)	Set this property to true if you don't want to show the Help button.
	Text (see page 232)	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 232)	Write Only value for passing in the UserName.

Legend

	Property

Login.AutoAddType Property

Set this property to define auto-added Active Directory (see page 118) users as members or authors.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to define auto-added
users as members or authors."), DefaultValue(Common.EkEnumeration.AutoAddUserTypes.Author)]
public Common.EkEnumeration.AutoAddUserTypes AutoAddType;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to define auto-added
```

```
users as members or authors."), DefaultValue(Common.EkEnumeration.AutoAddUserTypes.Author)>
Public Property AutoAddType() As Common.EkEnumeration.AutoAddUserTypes
```

Login.AutoLogin Property

Set this property to true if you want to use Auto Login (see page 229) with Active Directory (see page 118).

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to true if you want to
use Auto Login with Active Directory."), DefaultValue(false)]
public bool AutoLogin;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to true if you want to
use Auto Login with Active Directory."), DefaultValue(false)>
Public Property AutoLogin() As bool
```

Login.OnlyAllowMemberLogin Property

Set this property to true if you only allow membership users.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to true if you only allow
membership users."), DefaultValue(false)]
public bool OnlyAllowMemberLogin;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to true if you only allow
membership users."), DefaultValue(false)>
Public Property OnlyAllowMemberLogin() As bool
```

Login.Password Property

Write Only value for passing in the password.

C#

```
public string Password;
```

Visual Basic

```
Public Property Password() As string
```

Login.PromptLogout Property

Set this property to true if you only allow membership users.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to false if you do not
want to prompt users to verify logout."), DefaultValue(true)]
public bool PromptLogout;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to false if you do not
want to prompt users to verify logout."), DefaultValue(true)>
Public Property PromptLogout() As bool
```

Login.SuppressHelpButton Property

Set this property to true if you don't want to show the Help button.

C#

```
[Bindable(false), Category("Ektron"), Description("Set this property to true if you don't want
to show the Help button."), DefaultValue(false)]
```

```
public bool SuppressHelpButton;
```

Visual Basic

```
<Bindable(false), Category("Ektron"), Description("Set this property to true if you don't want
to show the Help button."), DefaultValue(false)>
Public Property SuppressHelpButton() As bool
```

Login.Text Property

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

C#

```
public override string Text;
```

Visual Basic

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

Login.UserName Property

Write Only value for passing in the UserName.

C#

```
public string UserName;
```

Visual Basic

```
Public Property UserName() As string
```

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 233)	

	GetCallbackResult (see page 233)	
	RaiseCallbackEvent (see page 233)	

Legend

	Method

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 234)	A default address while loading control, If this value is set, the Latitude (see page 237) and Longitude (see page 237) properties are ignored.
	ContentId (see page 234)	Use this property in conjunction with the EnableSearchResult (see page 236) property to limit the map to a single content (see page 449) item. Here, you identify the content (see page 449) item to be mapped.
	CustomIcon (see page 235)	Custom icon for the category.
	DateSearch (see page 235)	Enable or disable DateTime selector for map data.
	DisplayTextResult (see page 235)	Enable or disable text output for the search (see page 514) results.
	DistanceUnit (see page 236)	Distance measurement units in Miles or Kilometers..

	EnableSearchResult (see page 236)	Determine if the Map (see page 232) server control accesses the CMS search (see page 514) to return results. • **True** = Map (see page 232) server control accesses the CMS search (see page 514) to return results. • **False** = Display a map with only one content (see page 449) item, that is, a single location. Specify this item at the ContentID property. In this case, the map does not find other CMS content (see page 449) whose longitude and latitude are within the map's boundaries. True is the default. Site (see page 424) visitors viewing the map can still use the Search (see page 279) and Directions tabs to get directions to the location. For example, your site features homes for sale, and you... more (see page 236)
	FolderId (see page 236)	The ID of the folder from which to search (see page 514) map data.
	GeoControl (see page 237)	Enable or disable GeoControl which allows a user (see page 534) to perform a search (see page 514) and get directions.
	Latitude (see page 237)	Startup latitude value while loading control.
	Longitude (see page 237)	Startup longitude value while loading control.
	MapCategory (see page 237)	Category to perform search (see page 514). This category must be created through the taxonomy wizard.
	MapProvider (see page 238)	Provider to connect map server. The supported providers are VirtualEarth and Google.
	MapStyle (see page 238)	Map (see page 232) Style accepted values are "Hybrid", "Road" and "Satellite"
	MarkupLanguage (see page 238)	Name of a markup file that manages the display of this server control.
	MinZoomLevel (see page 238)	Search (see page 279) will not be performed beyond this zoom level.
	PageSize (see page 238)	Page Size to populate search (see page 514) result.
	Recursive (see page 239)	Enable or disable recursive search (see page 514) from the specified FolderId (see page 236).
	SelectedCategoryLogicalType (see page 239)	Logical type operators to search (see page 514) thru the selected map categories.
	StartZoomLevel (see page 239)	Startup zoom level. Accepted values between 1 and 19.
	TypeControl (see page 239)	Enables or disables the Google type control and the zoom, direction and type control for Virtual Earth.
	ZoomControl (see page 240)	Enable or Disable zoom control for Google map only.

Legend

	Property

Map.Address Property

A default address while loading control, If this value is set, the Latitude (see page 237) and Longitude (see page 237) 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.")]
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.")>
Public Property Address() As string
```

Map.ContentId Property

Use this property in conjunction with the EnableSearchResult (see page 236) property to limit the map to a single content (

see page 449) item. Here, you identify the content (see page 449) 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 int 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 Integer
```

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 (see page 514) results.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(true)]
[Description("Enable or disable text output for the search results.")]
public bool DisplayTextResult;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(true)>
<Description("Enable or disable text output for the search results.")>
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 232) server control accesses the CMS search (see page 514) to return results.

- **True** = Map (see page 232) server control accesses the CMS search (see page 514) to return results.
- **False** = Display a map with only one content (see page 449) item, that is, a single location. Specify this item at the ContentID property. In this case, the map does not find other CMS content (see page 449) whose longitude and latitude are within the map's boundaries.

True is the default.

Site (see page 424) visitors viewing the map can still use the Search (see page 279) 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 (see page 449) 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 (see page 514) 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 int 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 Integer
```

Map.GeoControl Property

Enable or disable GeoControl which allows a user (see page 534) to perform a search (see page 514) 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.")]
public string Latitude;
```

Visual Basic

```
<Bindable(true), Category("Ektron")>
<Description("Startup latitude value while loading control.")>
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.")]
public string Longitude;
```

Visual Basic

```
<Bindable(true), Category("Ektron")>
<Description("Startup longitude value while loading control.")>
Public Property Longitude() As string
```

Map.MapCategory Property

Category to perform search (see page 514). 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))]
public int 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))>
Public Property MapCategory() As Integer
```

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.")]
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.")>
Public Property MapProvider() As Provider
```

Map.MapStyle Property

Map (see page 232) 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 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
```

Map.MinZoomLevel Property

Search (see page 279) 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.")]
public int MinZoomLevel;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(1)>
<Description("Search will not be performed beyond this zoom level.")>
Public Property MinZoomLevel() As Integer
```

Map.PageSize Property

Page Size to populate search (see page 514) result.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(10)]
[Description("Page Size to populate search result.")]
public int PageSize;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(10)>
<Description("Page Size to populate search result.")>
Public Property PageSize() As Integer
```

Map.Recursive Property

Enable or disable recursive search (see page 514) from the specified FolderId (see page 236).

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.SelectedCategoryLogicalType Property

Logical type operators to search (see page 514) thru the selected map categories.

C#

```
[Bindable(true), Category("Ektron"), DefaultValue(LogicalType.Or)]
[Description("Logical type operators to search thru the selected map categories.")]
public LogicalType SelectedCategoryLogicalType;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(LogicalType.Or)>
<Description("Logical type operators to search thru 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.")]
public int StartZoomLevel;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), DefaultValue(13)>
<Description("Startup zoom level.  Accepted values between 1 and 19.")>
Public Property StartZoomLevel() As Integer
```

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 (see page 534) to create or update a profile of themselves in your membership list. The control creates the form based on the standard Ektron user (see page 534) properties and any custom user (see page 534) 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 241)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

Membership.Fill Method

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

C#

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

Visual Basic

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

Membership Properties

The properties of the Membership class are listed here.

Public Properties

	Name	Description
	BoardID (see page 242)	Specify the board ID to use for the terms and conditions text.
	DisplayMode (see page 242)	Type of membership form displayed. Choose from the following: • **UserRegistration** - lets a site visitor register as a membership user (see page 534). • **ResetPassword** - lets a membership user (see page 534) reset his password. • **UnsubscribeSecured** - lets a membership user (see page 534) unsubscribe. He needs to enter his username and password. • **UnsubscribeUnsecured** - lets a membership user (see page 534) 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.
	RedirectFailedURL (see page 242)	Redirect the user (see page 534) if registering to the site fails.
	RedirectSuccessURL (see page 243)	Redirect the user (see page 534) after successfully registering to the site.
	RegisterButtonImg (see page 243)	Specify the image to use instead of the register button. The image path should be relative to the root.
	RegisterButtonText (see page 243)	Specify the form submit button text. The default value is "Register".
	ResetButtonImg (see page 243)	Specify the image to use instead of the reset button. The image path should be relative to the root.
	ResetButtonText (see page 243)	Specify the form reset button text. The default is "Reset".
	SaveButtonImg (see page 244)	Specify the image to use instead of the save button. The image path should be relative to the root.
	SaveButtonText (see page 244)	Specify the form submit button text when logged in. The default value is "Save".
	ShowExtended (see page 244)	Include user (see page 534)'s custom properties defined in the CMS. The default value is "True".
	ShowTerms (see page 244)	Display the Terms and Conditions text. The default value is "False".
	Text (see page 245)	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 245)	User (see page 534) already exists error message. The default value is "Username(email) already exists!".
	UserID (see page 245)	Specify the user (see page 534) ID to retrieve information.
	UserSuccessMessage (see page 245)	User (see page 534) successfully registered message. The default value is "You have registered successfully".
	UserUpdateSuccessMessage (see page 246)	Message explaining the user (see page 534) successfully update his/her information. The default value is "You have successfully updated your information".

Legend

	Property

Membership.BoardID Property

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

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The ID of the forum you wish show terms and conditions for.")]
[Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int BoardID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The ID of the forum you wish show terms and conditions for.")>
<Editor(typeof(DefaultForumUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property BoardID() As Integer
```

Membership.DisplayMode Property

Type of membership form displayed. Choose from the following:

- **UserRegistration** - lets a site visitor register as a membership user (see page 534).
- **ResetPassword** - lets a membership user (see page 534) reset his password.
- **UnsubscribeSecured** - lets a membership user (see page 534) unsubscribe. He needs to enter his username and password.
- **UnsubscribeUnsecured** - lets a membership user (see page 534) 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 mode. Each mode has it
own UI and behaves differently."), DefaultValue(0)]
public Mode DisplayMode;
```

Visual Basic

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

Membership.RedirectFailedURL Property

Redirect the user (see page 534) 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 (see page 534) after successfully registering to the site.

C#

```
[Bindable(true), Category("Ektron"), Description("Redirect the user after successfully
registering to the site."), DefaultValue("")]
public string RedirectSuccessURL;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Redirect the user after successfully
registering to the site."), DefaultValue("")>
Public Property RedirectSuccessURL() As string
```

Membership.RegisterButtonImg Property

Specify the image to use instead of the register button. The image path should be relative to the root.

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
register button. The image path should be relative to the root."), DefaultValue("")]
public string RegisterButtonImg;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
register button. The image path should be relative to the root."), DefaultValue("")>
Public Property RegisterButtonImg() As string
```

Membership.RegisterButtonText Property

Specify the form submit button text. The default value is "Register".

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the form submit button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_REGBTN)]
public string RegisterButtonText;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the form submit button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_REGBTN)>
Public Property RegisterButtonText() As string
```

Membership.ResetButtonImg Property

Specify the image to use instead of the reset button. The image path should be relative to the root.

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
reset button. The image path should be relative to the root."), DefaultValue("")]
public string ResetButtonImg;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the image to use instead of the
reset button. The image path should be relative to the root."), DefaultValue("")>
Public Property ResetButtonImg() As string
```

Membership.ResetButtonText Property

Specify the form reset button text. The default is "Reset".

C#

```
[Bindable(true), Category("Ektron"), Description("Specify the form reset button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_RESTBTN)]
public string ResetButtonText;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Specify the form reset button text."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_RESTBTN)>
Public Property ResetButtonText() As string
```

Membership.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. The default value is "Save".

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 (see page 534)'s custom properties defined in the CMS. The default value is "True".

C#

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

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("Include user's custom properties defined in
the CMS."), DefaultValue(true)>
Public Property ShowExtended() As bool
```

Membership.ShowTerms Property

Display the Terms and Conditions text. The default value is "False".

C#

```
[Bindable(true), Category("Ektron"), Description(""), DefaultValue(false)]
```

```
public bool ShowTerms;
```

Visual Basic

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

Membership.Text Property

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

C#

```
public override string Text;
```

Visual Basic

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

Membership.UserExistsMessage Property

User (see page 534) already exists error message. The default value is "Username(email) already exists!".

C#

```
[Bindable(true), Category("Ektron"), Description("User already exists error message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_EXISTS)]
public string UserExistsMessage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("User already exists error message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_EXISTS)>
Public Property UserExistsMessage() As string
```

Membership.UserID Property

Specify the user (see page 534) ID to retrieve information.

C#

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

Visual Basic

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

Membership.UserSuccessMessage Property

User (see page 534) successfully registered message. The default value is "You have registered successfully".

C#

```
[Bindable(true), Category("Ektron"), Description("User successfully registered message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_SUCCEED)]
public string UserSuccessMessage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("User successfully registered message."),
TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_SUCCEED)>
Public Property UserSuccessMessage() As string
```

Membership.UserUpdateSuccessMessage Property

Message explaining the user (see page 534) successfully update his/her information. The default value is "You have successfully updated your information".

C#

```
[Bindable(true), Category("Ektron"), Description("User successfully update his/her
information."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_UPDATE)]
public string UserUpdateSuccessMessage;
```

Visual Basic

```
<Bindable(true), Category("Ektron"), Description("User successfully update his/her
information."), TypeConverter(typeof(LocalizablePropTypeConverter)),
DefaultValue(EkWebControl.RES_MEMBR_UPDATE)>
Public Property UserUpdateSuccessMessage() As string
```

Menu Class

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 248) 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 246)	Calls up the database and updates the contents of the Menu (see page 246) if the data is out of sync with the specified parameters.
	Wrap (see page 247)	Method that can overridden and be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

Menu.Fill Method

Calls up the database and updates the contents of the Menu (see page 246) 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 247)	Sets the cache interval in second(s).
	DefaultMenuID (see page 248)	Gets or Sets the Default MenuID for this Control.
	DisplayXslt (see page 248)	The Xslt to use to render the Menu (see page 246) (optional)
	DynamicParameter (see page 248)	The parameter to query for a Dynamic Menu (see page 246).
	Stylesheet (see page 249)	Specify the style sheet relative to site root. Leave blank for default.
	SuppressAddEdit (see page 249)	etting this property to true will cause the menu to not render the add and edit buttons for the menu. It is defaulted to false.

Legend

	Property
	virtual

Menu.CacheInterval Property

Sets the cache interval in second(s).

C#

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

Visual Basic

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

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

Menu.DefaultMenuID Property

Gets or Sets the Default MenuID for this Control.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Editor(typeof(MenuUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
[Description("Gets or Sets the Default MenuID for this Control.")]
public int DefaultMenuID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Editor(typeof(MenuUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<Description("Gets or Sets the Default MenuID for this Control.")>
Public Property DefaultMenuID() As Integer
```

Menu.DisplayXslt Property

The Xslt to use to render the Menu (see page 246) (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 246).

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

etting this property to true will cause the menu to not render the add and edit buttons for the menu. It is defaulted to false.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Setting this property to true will cause the menu to not render 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 will cause the menu to not render the add and edit
buttons for the menu. It is defaulted to false.")>
<DefaultValue(false)>
Public virtual Property SuppressAddEdit() As bool
```

MetaData Class

The Metadata server control lets you add the metadata from content (see page 449) 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 (see page 449) block, multiple content (see page 449) blocks, or by dynamically passing a content (see page 449) 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 250)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

MetaData.Fill Method

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

C#

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

Visual Basic

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

MetaData Properties

The properties of the MetaData class are listed here.

Public Properties

	Name	Description
	DefaultContentID (see page 250)	The content (see page 449) block ID from which the server control gets the metadata. If you want to add metadata from several content (see page 449) blocks, set this property to 0 (zero) and use the DefaultItemList (see page 251) property to identify the content (see page 449) blocks.
	DefaultItemList (see page 251)	A comma delimited list of content (see page 449) block IDs from which to get metadata. DefaultContentID (see page 250) must be set to 0 (zero) so you can add IDs to the list.
	DynamicParameter (see page 251)	Gets or sets the QueryString parameter to read a content (see page 449) ID dynamically. Set to "Use Default" if you wish to always display the default ContentID (static.)
	GenerateDublinCore (see page 251)	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 249) Server Control" > "Using the Simple Dublin Core Metadata (see page 412) Standard"

Legend

	Property

MetaData.DefaultContentID Property

The content (see page 449) block ID from which the server control gets the metadata. If you want to add metadata from several content (see page 449) blocks, set this property to 0 (zero) and use the DefaultItemList (see page 251) property to identify the content (see page 449) 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 int DefaultContentID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Editor(typeof(ContentBlockUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
<DefaultValue(0)>
<Description("Gets or Sets the Default ContentID for this Control.")>
Public Property DefaultContentID() As Integer
```

MetaData.DefaultItemList Property

A comma delimited list of content (see page 449) block IDs from which to get metadata. DefaultContentID (see page 250) must be set to 0 (zero) so you can add IDs to the list.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The list of content ID's to build the meta data from.")]
public string DefaultItemList;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The list of content ID's to build the meta data from.")>
Public Property DefaultItemList() As string
```

MetaData.DynamicParameter Property

Gets or sets the QueryString parameter to read a content (see page 449) 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 249) Server Control" > "Using the Simple Dublin Core Metadata (see page 412) 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 253)	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 253)	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 253)	Gets a System.Collections.IEnumerator for the collection.
	MoveNext (see page 253)	Moves up the index of the Collection. Returns false if the end of the collection has been surpassed.
	Reset (see page 253)	Brings the index of the collection back to zero. Does not clear the collection.

	this (see page 254)	Index operator into a strongly typed dataset holding some common properties for a Collection item.
	Wrap (see page 254)	Overrideable method that can be used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

MetaDataList.CopyTo Method

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

C#

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

Visual Basic

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

MetaDataList.Fill Method

Calls up the database and updates the items in the control if the data is out of sync with the specified parameters.

C#

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

Visual Basic

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

MetaDataList.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

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

Visual Basic

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

MetaDataList.MoveNext Method

Moves up the index of the Collection. 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 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 255)	Sets the cache interval in second(s).
	ContentType (see page 256)	Gets or sets the ContentType for this control.
	Count (see page 256)	Indicates the number of elements in your Collection.
	Current (see page 256)	Gets the current object in the collection.
	EkItems (see page 256)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection. Holds more information than the exposed strongly typed this[i] index operator.
	EnablePaging (see page 256)	Enables paging if the total records exceeds the MaxResult value.
	ExactPhrase (see page 257)	Gets or Sets the boolean value of ExactPhrase. Determines whether the KeyWordValue (see page 259) needs to match the phrase exactly. For example, if “site” is the KeyWordValue (see page 259) and the title of a content (see page 449) block is “Welcome (see page 1) to the site” and ExactPhrase is set to true, you would not see the content (see page 449) block in the metadata list. This is because “site” does not equal “Welcome (see page 1) to the site”. • **True** = match the exact phrase • **False** = does not need to match exact phrase
	FolderID (see page 257)	The folder ID from which content (see page 449) is pulled from.
	GetHtml (see page 258)	If set to true, retrieves the content (see page 449) (Html body) of every content (see page 449) block in the MetadataList for display.
	IncludeIcons (see page 258)	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 258)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	KeyWordName (see page 258)	Gets or Sets KeyWordName for this control.
	KeyWordValue (see page 259)	Gets or sets the KeyWordValue. The value that is associated with the KeyWord
	KeyWordValueMatchAll (see page 259)	Match all values specified in KeyWordValue (see page 259) separator by KeyWordValueSeparator (see page 259).
	KeyWordValueSeparator (see page 259)	Gets or Sets KeyWordValueSeparator for this control. For example: **;**
	LinkTarget (see page 260)	Gets or Sets the target window for the result. Choices are: • **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named. • **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned BASE target. • **_parent** - this target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - this target makes the link load in the full... more (see page 260)
	MarkupLanguage (see page 260)	Enter name of a markup file that manages the display of this server control.
	MaxNumber (see page 260)	Determine the maximum number of content (see page 449) items to return. If set to zero then all items are returned.
	OrderBy (see page 261)	The order of the list to be returned. Order by: • **Title** - The title of the content (see page 449) block • **ID** - The content (see page 449) block ID number • **Date Created** - The date the content (see page 449) block was created • **Date Modified** - The date the content (see page 449) block was last modified • **LastEditorLname** - The last editor's last name • **LastEditorFname** - The last editor's first name
	Recursive (see page 261)	If set to true, it will carry out a recursive find from the folderID and all it's children folders. Set to false for only content (see page 449) in that folder.
	SortOrder (see page 261)	Gets or sets the sort order. Choose whether to sort by Ascending or Descending.
	SuppressWrapperTags (see page 262)	Suppresses the output of span/div tags around the control.
	SyncRoot (see page 262)	Gets an object that can be used to synchronize the collection.
	XmlDoc (see page 262)	Returns the XSLT object for the MetaDataList (see page 252) control.

Legend

	Property
	virtual

MetaDataList.CacheInterval Property

Sets the cache interval in second(s).

C#

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

Visual Basic

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

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

MetaDataList.ContentType Property

Gets or sets the ContentType for this control.

C#

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

Visual Basic

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

MetaDataList.Count Property

Indicates the number of elements in your Collection.

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. Holds more information than the exposed strongly typed this[i] index operator.

C#

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

Visual Basic

```
<Browsable(false)>
Public Property EkItems() As Ektron.Cms.Common.ContentBase
```

MetaDataList.EnablePaging Property

Enables paging if the total records exceeds the MaxResult value.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Enables paging if total records exceeds MaxResult value. Use this property with
non zero value of MaxNumbers.")]
public bool EnablePaging;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Enables paging if total records exceeds MaxResult value. Use this property with
non zero value of MaxNumbers.")>
Public Property EnablePaging() As bool
```

MetaDataList.ExactPhrase Property

Gets or Sets the boolean value of ExactPhrase. Determines whether the KeyWordValue (see page 259) needs to match the phrase exactly.

For example, if “site” is the KeyWordValue (see page 259) and the title of a content (see page 449) block is “Welcome (see page 1) to the site” and ExactPhrase is set to true, you would not see the content (see page 449) 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("The value that is determines if the KeyWordValue is in the Metadata list(false)
or if it is the only value in the list(true)")]
public bool ExactPhrase;
```

Visual Basic

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

MetaDataList.FolderID Property

The folder ID from which content (see page 449) is pulled from.

C#

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

Visual Basic

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

```
Public Property FolderID() As Integer
```

MetaDataList.GetHtml Property

If set to true, retrieves the content (see page 449) (Html body) of every content (see page 449) 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

Gets or Sets KeyWordName for this control.

C#

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

Visual Basic

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

MetaDataList.KeyWordValue Property

Gets or sets the KeyWordValue. The value that is associated with the KeyWord

C#

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

Visual Basic

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

MetaDataList.KeyWordValueMatchAll Property

Match all values specified in KeyWordValue (see page 259) separator by KeyWordValueSeparator (see page 259).

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

Gets or Sets KeyWordValueSeparator for this control. For example: **;**

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(";")]
[Description("Specify KeyWordValue separated by KeyWordValueSeparator")]
public string KeyWordValueSeparator;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(";")>
<Description("Specify KeyWordValue separated by KeyWordValueSeparator")>
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.")]
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
```

MetaDataList.MaxNumber Property

Determine the maximum number of content (see page 449) 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 (see page 449) block
- **ID** - The content (see page 449) block ID number
- **Date Created** - The date the content (see page 449) block was created
- **Date Modified** - The date the content (see page 449) 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")]
public Ektron.Cms.Controls.CmsWebService.ContentOrderBy OrderBy;
```

Visual Basic

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

MetaDataList.Recursive Property

If set to true, it will carry out a recursive find from the folderID and all it's children folders. Set to false for only content (see page 449) in that folder.

C#

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

Visual Basic

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

MetaDataList.SortOrder Property

Gets or sets the sort order. Choose whether to sort by Ascending or Descending.

C#

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

Visual Basic

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

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

MetaDataList.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 252) control.

C#

```
public override XmlDocument XmlDoc;
```

Visual Basic

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

PersonalizationManager Class

The PersonalizationManager server control allows site users to personalize, add, edit and hide content (see page 449) on a Web page. The level of privileges assigned to a site user (see page 534) in CMS400.NET determines what personalization functionality is available when the control is active. The Personalization (see page 10) manager only appears when a user (see page 534) is logged in to the site.

Class Hierarchy

C#

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

Visual Basic

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

File

PersonalizationManager.cs

See Also

The Administrator Manual section "Personalization (see page 10) Feature" and the Developer Manual section "Personalization (see page 10) 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 263)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

PersonalizationManager.Fill Method

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

C#

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

Visual Basic

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

PersonalizationManager Properties

The properties of the PersonalizationManager class are listed here.

Public Properties

	Name	Description
	AddContentExitText (see page 264)	Gets or sets the text for the Add Content (see page 449) Exit link. The default text is **Done Add Content (see page 449)**.
	AddContentText (see page 264)	Gets or sets the text for the Add Content (see page 449) link. The default is **Add Content (see page 449).**
	CurrentUserButtonText (see page 264)	Gets or sets the text for the Current User (see page 534) radio button. The default is **Current User (see page 534).**
	Display (see page 265)	Gets or sets the display format to vertical or horizontal for the Personalization (see page 10) Manager. The default is Vertical.
	HeaderBackColor (see page 265)	`HeaderBackColor` sets the background color of the personalization manager header.
	HeaderText (see page 265)	The text that appears in the header of the Personalization (see page 10) Manager.
	PersonalizeExitText (see page 266)	Gets or sets the text for the Exit link. The default is **Done**.
	PersonalizeScopeText (see page 266)	Gets or sets the text for the Scope. The default is **Personalization (see page 10) For.**
	PersonalizeStartText (see page 266)	Gets or sets the text for the Start link. The default is **Personalize.**
	PublicUserButtonText (see page 267)	Gets or sets the text for the Public User (see page 534) radio button. The default is **Public User (see page 534).**
	ResetLinkText (see page 267)	Gets or sets the text for the Reset link text. The default is **Reset To Default.**
	ResetLinkToolTip (see page 267)	Gets or sets the tool tip text for the Reset link. The default is **Reset the current user (see page 534)'s personalization data for the page.**

Legend

	Property

PersonalizationManager.AddContentExitText Property

Gets or sets the text for the Add Content (see page 449) Exit link. The default text is **Done Add Content (see page 449)**.

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 449) link. The default is **Add Content (see page 449).**

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 534) radio button. The default is **Current User (see page 534).**

C#

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

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_USRSCOPE)>
<Description("The display text for the curront user radio button. Default value is set as \""
```

```
+ EkWebControl.RES_PMC_USRSCOPE + "\".")>
Public Property CurrentUserButtonText() As string
```

PersonalizationManager.Display Property

Gets or sets the display format to vertical or horizontal for the Personalization (see page 10) 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

C#

```
[Description("HeaderBackColor sets the header background color."), Category("Ektron")]
[Bindable(false)]
[Browsable(true)]
[TypeConverter(typeof(WebColorConverter))]
public Color HeaderBackColor;
```

Visual Basic

```
<Description("HeaderBackColor sets the header background color."), Category("Ektron")>
<Bindable(false)>
<Browsable(true)>
<TypeConverter(typeof(WebColorConverter))>
Public Property HeaderBackColor() As Color
```

Description

`HeaderBackColor` sets the background color of the personalization manager header.

PersonalizationManager.HeaderText Property

The text that appears in the header of the Personalization (see page 10) Manager.

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_HEADER)]
[Description("The HeaderText of the control. Default value is set as \"" +
EkWebControl.RES_PMC_HEADER + "\".")]
public string HeaderText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_HEADER)>
<Description("The HeaderText of the control. Default value is set as \"" +
EkWebControl.RES_PMC_HEADER + "\".")>
Public Property HeaderText() As string
```

PersonalizationManager.PersonalizeExitText Property

Gets or sets the text for the Exit link. The default is **Done**.

C#

```
[Bindable(false)]
[Category("Ektron")]
[TypeConverter(typeof(LocalizablePropTypeConverter))]
[DefaultValue(EkWebControl.RES_PMC_EXIT)]
[Description("Set text for exit personalization mode link.  Default value is set as \"" +
EkWebControl.RES_PMC_EXIT + "\".")]
public string PersonalizeExitText;
```

Visual Basic

```
<Bindable(false)>
<Category("Ektron")>
<TypeConverter(typeof(LocalizablePropTypeConverter))>
<DefaultValue(EkWebControl.RES_PMC_EXIT)>
<Description("Set text for exit personalization mode link.  Default value is set as \"" +
EkWebControl.RES_PMC_EXIT + "\".")>
Public Property PersonalizeExitText() As string
```

PersonalizationManager.PersonalizeScopeText Property

Gets or sets the text for the Scope. The default is **Personalization (see page 10) 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 534) radio button. The default is **Public User (see page 534).**

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 (see page 534)'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
```

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 268)	Fills in the poll area with the content (see page 449) information from the content (see page 449) database. The form data is retrieved and displayed. Only use if the content (see page 449) must be refreshed after an form ID change or other operation that will require regenerating the display.

Legend

	Method

Poll.Fill Method

Fills in the poll area with the content (see page 449) information from the content (see page 449) database. The form data is retrieved and displayed. Only use if the content (see page 449) must be refreshed after an form ID change or other operation that will require regenerating the display.

C#

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

Visual Basic

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

Poll Properties

The properties of the Poll class are listed here.

Public Properties

	Name	Description
	EnableAjax (see page 269)	Determines if the results will show in the area of the page that contains the poll. If the value is 'true' then the results will be shown in the area containing the poll without any modification or refresh of the surrounding contents. Otherwise, the poll results are displayed in place of the page content (see page 449).
	PollID (see page 269)	to be changed

Legend

	Property

Poll.EnableAjax Property

Determines if the results will show in the area of the page that contains the poll. If the value is 'true' then the results will be shown in the area containing the poll without any modification or refresh of the surrounding contents. Otherwise, the poll results are displayed in place of the page content (see page 449).

C#

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

Visual Basic

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

Poll.PollID Property

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.")]
public int 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.")>
Public Property PollID() As Integer
```

Description

to be changed

PostHistory Class

The PostHistory server control displays a list of posts for a given user (see page 534). 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 449) - the content (see page 449) 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 270)	Method that can be overridden and used to grab data from the back end for each server control.
	ToString (see page 270)	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 271)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method

PostHistory.Fill Method

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

C#

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

Visual Basic

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

PostHistory.ToString Method

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

C#

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

Visual Basic

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

PostHistory.Wrap Method

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

C#

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

Visual Basic

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

Parameters

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

Returns

Html string wrapped with a toolbar

PostHistory Properties

The properties of the PostHistory class are listed here.

Public Properties

	Name	Description
	BoardID (see page 271)	The ID of the discussion board from which to get a user (see page 534)'s posts.
V	CacheInterval (see page 272)	Gets or sets the cache interval in second(s).
	MaxNumber (see page 272)	The maximum number of posts listed. The default is **10**.
	URLPath (see page 272)	The URL path to the page the where the Forum (see page 167) server control is located.
	UserID (see page 272)	The ID of the user (see page 534) for whom to get the post history.

Legend

	Property
V	virtual

PostHistory.BoardID Property

The ID of the discussion board from which to get a user (see page 534)'s posts.

C#

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

Visual Basic

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

PostHistory.CacheInterval Property

Gets or sets the cache interval in second(s).

C#

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

Visual Basic

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

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 167) 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 (see page 534) 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 int 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 Integer
```

RssAggregator Class

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

Class Hierarchy

C#

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

Visual Basic

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

File

RssAggregator.cs

RssAggregator Methods

The methods of the RssAggregator class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 274)	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 274)	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 274)	Gets a System.Collections.IEnumerator for the collection.
	GetView (see page 274)	Gets the view data using dataset
	GetViewNames (see page 274)	gets the view names
	MoveNext (see page 275)	Moves up the index of the Collection. Returns false if the end of the collection has been surpassed.
	Reset (see page 275)	Brings the index of the collection back to zero. Does not clear the collection.
	this (see page 275)	Index operator into a strongly typed dataset holding some common properties for a Collection item.

Legend

	Method

RssAggregator.CopyTo Method

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

C#

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

Visual Basic

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

RssAggregator.Fill Method

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

C#

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

Visual Basic

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

RssAggregator.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

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

Visual Basic

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

RssAggregator.GetView Method

Gets the view data using dataset

C#

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

Visual Basic

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

Parameters

Parameters	Description
viewName	

Returns

DataSourceView

RssAggregator.GetViewNames Method

gets the view names

C#

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

Visual Basic

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

Returns

View names

RssAggregator.MoveNext Method

Moves up the index of the Collection. 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 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 276)	Indicates the number of elements in your Collection.
	Current (see page 276)	Gets the current object in the collection.
	EkItems (see page 276)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection. Holds more information than the exposed strongly typed this[i] index operator.
	IsSynchronized (see page 276)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.

	LinkTarget (see page 277)	Gets or Sets the target window for the result. Choices are: • **_blank** - This target causes the link to always be loaded in a new blank window. This window is not named. • **_self** - This target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. • **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - This target makes the link load in the full... more (see page 277)
	MaxResults (see page 277)	The maximum number of results to fetch. If set to zero, there is no limit.
	SyncRoot (see page 277)	Gets an object that can be used to synchronize the collection.
	URL (see page 278)	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

Legend

	Property

RssAggregator.Count Property

Indicates the number of elements in your Collection.

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. 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.") ]
public ItemLinkTargets LinkTarget;
```

Visual Basic

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

RssAggregator.MaxResults Property

The maximum number of results to fetch. If set to zero, there is no limit.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The maximum number of results to fetch. If set to zero, there is no limit.")]
public int MaxResults;
```

Visual Basic

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

RssAggregator.SyncRoot Property

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

C#

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

Visual Basic

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

RssAggregator.URL Property

Gets or Sets the RSS Feed URL for this Control. 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.")]
public string URL;
```

Visual Basic

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

RssItemBase Class

Container for Title (see page 279), Link (see page 279), Description (see page 279) and DatePosted (see page 278) article information.

Class Hierarchy

Ektron.Cms.Controls.RssItemBase

C#

```
public class RssItemBase;
```

Visual Basic

```
Public Class RssItemBase
```

File

RssAggregator.cs

RssItemBase Fields

The fields of the RssItemBase class are listed here.

Public Fields

	Name	Description
	DatePosted (see page 278)	The date the article was posted.
	Description (see page 279)	The description of the article.
	Link (see page 279)	The link for the article.
	Title (see page 279)	The title of the article.

Legend

	Data Member

RssItemBase.DatePosted Field

The date the article was posted.

C#

```
public DateTime DatePosted = new DateTime(0);
```

Visual Basic

```
Public DatePosted As DateTime = new DateTime(0)
```

RssItemBase.Description Field

The description of the article.

C#

```
public string Description = "";
```

Visual Basic

```
Public Description As string = ""
```

RssItemBase.Link Field

The link for the article.

C#

```
public string Link = "";
```

Visual Basic

```
Public Link As string = ""
```

RssItemBase.Title Field

The title of the article.

C#

```
public string Title = "";
```

Visual Basic

```
Public Title As string = ""
```

Search Class

Search object returns the control and results.

Class Hierarchy

C#

```
[DefaultProperty("FolderID"), ToolboxData("<{0}:Search runat=server></{0}:Search>")]
[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 Search()
```

Search Methods

The methods of the Search class are listed here.

Public Methods

	Name	Description
	CopyTo (see page 280)	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 280)	Method that can be overridden and used to grab data from the back end for each server control.
	GetEnumerator (see page 281)	Gets a System.Collections.IEnumerator for the collection.
	LoadSearch (see page 281)	
	MoveNext (see page 281)	Moves up the index of the Collection. Returns false if the end of the collection has been surpassed.
	Reset (see page 281)	Brings the index of the collection back to zero. Does not clear the collection.
V	SearchDisplayUI (see page 281)	
	this (see page 282)	Index operator into a strongly typed dataset holding some common properties for a Collection item.
	Wrap (see page 282)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method
V	virtual

Search.CopyTo Method

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

C#

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

Visual Basic

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

Search.Fill Method

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

C#

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

Visual Basic

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

Search.GetEnumerator Method

Gets a System.Collections.IEnumerator for the collection.

C#

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

Visual Basic

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

Search.LoadSearch Method

C#

```
public void LoadSearch(ekSearchConfig SearchReq);
```

Visual Basic

```
Public Function LoadSearch(SearchReq As ekSearchConfig) As void
```

Parameters

Parameters	Description
SearchReq	

Search.MoveNext Method

Moves up the index of the Collection. 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 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 284)	Optional: The path to the image you wish to use for the button.
	ButtonText (see page 285)	The text of the search (see page 514) button if no image is specified.
	ContentFieldsetLegend (see page 285)	Text (see page 144) that appears in the fieldset border around the content (see page 449) check boxes. This property is used only when the EnableContentFieldset (see page 288) property is set to True. The default is: **Content (see page 449) Items:**
	Count (see page 285)	Indicates the number of elements in your Collection.
	Current (see page 286)	Gets the current object in the collection.
	Display (see page 286)	Determines if the search (see page 514) options are in a row (**Horizontal**) or a column (**Vertical**).
	EkItems (see page 286)	Array of Ektron.Cms.Common.ContentBase items that correspond to the specified Collection. Holds more information than the exposed strongly typed this[i] index operator.
	EmptyResultMsg (see page 286)	Optional: The return massage if the search (see page 514) returns no hits.
	EnableAdvancedLink (see page 287)	Set to true to display the link that makes the advanced properties visible.
	EnableAjax (see page 287)	By default, the search (see page 514) is set to PostBack. To enable an Ajax search (see page 514), 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 287)	Set to true to include Archived items in the search (see page 514) results.
	EnableArchivedBtn (see page 288)	Set to true to provide a check box so users can decide if they want to perform a Archived search (see page 514). When set to False the check box is hidden but still used.

	EnableContent (see page 288)	Set to true to include Content (see page 449) items in the search (see page 514) results.
	EnableContentBtn (see page 288)	Set to true to provide a check box so users can decide if they want to perform a content (see page 449) search (see page 514). When set to False the check box is hidden but still used.
	EnableContentFieldset (see page 288)	Creates a border around the group of content (see page 449) items in an advanced search (see page 514). These content (see page 449) items include Content (see page 449), Forms and DMS Assets. The default is False.
	EnableDMSAssets (see page 289)	Set to true to include DMS-Asset (see page 450) items in the search (see page 514) results.
	EnableDMSAssetsBtn (see page 289)	Set to true to provide a check box so users can decide if they want to perform a DMS-Asset (see page 450) search (see page 514). When set to False the check box is hidden but still used.
	EnableFolderFieldset (see page 289)	Set to true to wrap the Folder (see page 383)-Descriptions in a fieldset border.
	EnableForms (see page 290)	Set to true to include Form (see page 493) items in the search (see page 514) results.
	EnableFormsBtn (see page 290)	Set to true to provide a check box so users can decide if they want to perform a Forms search (see page 514). When set to False the check box is hidden but still used.
	EnableLibFiles (see page 290)	Set to true to include Library (see page 395)-File items in the search (see page 514) results.
	EnableLibFilesBtn (see page 291)	Set to true to provide a check box so users can decide if they want to perform a Library (see page 395)-Files search (see page 514). When set to False the check box is hidden but still used.
	EnableLibHyperlinks (see page 291)	Set to true to include Library (see page 395)-Hyperlink items in the search (see page 514) results.
	EnableLibHyperlinksBtn (see page 291)	Set to true to provide a check box so users can decide if they want to perform a Library (see page 395)-Hyperlinks search (see page 514). When set to False the check box is hidden but still used.
	EnableLibImages (see page 292)	Set to true to include Library (see page 395)-Image items in the search (see page 514) results.
	EnableLibImagesBtn (see page 292)	Set to true to display the Library (see page 395)-Images check box (false to hide, but still use).
	EnableLibraryFieldset (see page 292)	Set to true to wrap the Library (see page 395)-Group in a fieldset border.
	FolderFieldsetLegend (see page 292)	The Legend-text to display, for the Folder (see page 383)-Descriptions fieldset.
	FolderID (see page 293)	The folder at which the search (see page 514) begins. The starting folder need not be the root folder. The Recursive (see page 297) property determines if the search (see page 514) examines this folder's subfolders.
	FolderPath (see page 293)	The Path of the folder you wish to start the search (see page 514). Root starts at "\".
	IncludeIcons (see page 293)	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 294)	Gets a value indicating whether the System.Data.InternalDataCollectionBase is synchronized.
	LibraryFieldsetLegend (see page 294)	The Legend-text to display, for the Library (see page 395)-Group in a fieldset.
	LinkTarget (see page 294)	Gets or Sets the target window for the result. The choices are: • **_blank** - this target causes the link to always be loaded in a new blank window. This window is not named. • **_self** - this target causes the link to always load in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. • **_parent** - This target makes the link load in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - This target makes the link load in the... more (see page 294)
	MaxCharacters (see page 295)	The maximum number of characters the user (see page 534) can enter in the search (see page 514) box.
	MaxResults (see page 295)	The total number of results to display. If set to zero, unlimited results will be set.
	MaxTeaserLength (see page 295)	An optional result formatting parameter. If non-zero, then the matching content (see page 449) 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 296)	Specify the order of the results. Choices are: • **Title** - the title of the content (see page 449) • **Id** - the ID number content (see page 449) • **DateCreated** - the date the creation date of the content (see page 449) • **DateModified** - the date the content (see page 449) was last modified • **UserId** - ID of a user (see page 534) • **LastEditorLname** - the last name of the last content (see page 449) editor • **LastEditorFname** - the first name of the last content (see page 449) editor
	OrderDirection (see page 296)	Specify the order direction of the results. Choices are Ascending or Descending.
	PostResultsAt (see page 296)	The Path of the page you wish to post search (see page 514) results. (NOTE: The search (see page 514) control MUST exists on the posting page.)
	Recursive (see page 297)	Get or set whether to search (see page 514) subfolders of the identified root folder. The root folder is identified in the FolderID (see page 293) property.
	RemoveTeaserHtml (see page 297)	An optional result formatting parameter. If true, any HTML in the content (see page 449) 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 297)	Specify the div tag control ID to display the search (see page 514) results. This property is only active when the EnableAjax (see page 287) property is set to true.
	SearchText (see page 298)	Get or set the default search (see page 514) text.
	SearchType (see page 298)	Get or set the Search (see page 279) Type
	ShowDate (see page 298)	Select true if you wish to display the publication date on each search (see page 514) result.
	ShowExtendedSearch (see page 298)	Allows a user (see page 534) to search (see page 514) via Searchable Property metadata. • **True** = Show extended search (see page 514) • **False** = Hide (see page 142) extended search (see page 514) Search (see page 279) type metadata only appears on the search (see page 514) screen if the following are true: • this property is set to True • the ShowSearchOptions (see page 299) property is set to True • at least one content (see page 449) item in the starting folder (and its subfolders if recursive is set to true) has searchable metadata applied to it
	ShowSearchBoxAlways (see page 299)	If set to false, the search (see page 514) box does not appear on PostBack. The default is True. • **True** = display search (see page 514) box on PostBack • **False** = do not display search (see page 514) box on PostBack
	ShowSearchOptions (see page 299)	Specify "True" to show search (see page 514) options. Set to false to see a very basic UI.
	Stylesheet (see page 299)	Specify the style sheet relative to site root. Leave blank for default.
	SyncRoot (see page 300)	Gets an object that can be used to synchronize the collection.
	TargetFolderDescriptionList (see page 300)	An optional comma-delimited string, which allows the user (see page 534) to override the folders to search (see page 514). For example, set to "Photos=1,NewsHeadlines=0" to search (see page 514) only the Photos folder by default, and allow the user (see page 534) to also select the "NewsHeadlines" folder. Also searches these folders' subfolders, returning any matches found there.
	TextBoxSize (see page 300)	The width of the search (see page 514) box, in characters.
	XmlDoc (see page 301)	Returns the Xslt object for the collection.

Legend

	Property

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 (see page 514) 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 144) that appears in the fieldset border around the content (see page 449) check boxes. This property is used only when the EnableContentFieldset (see page 288) property is set to True. The default is: **Content (see page 449) 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 your Collection.

C#

```
[Category("Data")]
```

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

Visual Basic

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

Search.Current Property

Gets the current object in the collection.

C#

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

Visual Basic

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

Search.Display Property

Determines if the search (see page 514) 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. 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 (see page 514) 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 (see page 514) is set to PostBack. To enable an Ajax search (see page 514), 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 (see page 514) 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 (see page 514). 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 449) items in the search (see page 514) 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 (see page 449) search (see page 514). 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 (see page 449) items in an advanced search (see page 514). These content

(see page 449) items include Content (see page 449), 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 450) items in the search (see page 514) 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 450) search (see page 514). 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 383)-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 493) items in the search (see page 514) 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 (see page 514). 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 395)-File items in the search (see page 514) 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 395)-Files search (see page 514). 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 395)-Hyperlink items in the search (see page 514) 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 395)-Hyperlinks search (see page 514). 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 395)-Image items in the search (see page 514) 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 395)-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 395)-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 383)-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 (see page 514) begins. The starting folder need not be the root folder. The Recursive (see page 297) property determines if the search (see page 514) examines this folder's subfolders.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder you wish to select data from. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderID() As Integer
```

Search.FolderPath Property

The Path of the folder you wish to start the search (see page 514). 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 395)-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 (see page 534) can enter in the search (see page 514) box.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(50)]
[Description("The maximum number of characters the user can enter in the search box.")]
public int MaxCharacters;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(50)>
<Description("The maximum number of characters the user can enter in the search box.")>
Public Property MaxCharacters() As Integer
```

Search.MaxResults Property

The total number of results to display. If set to zero, unlimited results will be set.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("The total number of results to display. If set to zero, unlimited results will
be set.")]
public int MaxResults;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("The total number of results to display. If set to zero, unlimited results will
be set.")>
Public Property MaxResults() As Integer
```

Search.MaxTeaserLength Property

An optional result formatting parameter. If non-zero, then the matching content (see page 449) items' teaser/summary will be truncated to the given maximum number of characters - followed by an ellipsis (...). If zero, then the entire teaser/summary will be returned (which might be further filtered by the XSLT).

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(100)]
[Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")]
public int MaxTeaserLength;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(100)>
<Description("An optional search-result formatting parameter that limits the length of any
returned content items' teaser/summary (or not truncate if set to zero).")>
Public Property MaxTeaserLength() As Integer
```

Search.OrderBy Property

Specify the order of the results. Choices are:

- **Title** - the title of the content (see page 449)
- **Id** - the ID number content (see page 449)
- **DateCreated** - the date the creation date of the content (see page 449)
- **DateModified** - the date the content (see page 449) was last modified
- **UserId** - ID of a user (see page 534)
- **LastEditorLname** - the last name of the last content (see page 449) editor
- **LastEditorFname** - the first name of the last content (see page 449) 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 (see page 514) results. (NOTE: The search (see page 514) 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 (see page 514) subfolders of the identified root folder. The root folder is identified in the FolderID (see page 293) 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 (see page 449) 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 (see page 514) results. This property is only active when the EnableAjax (see page 287) 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 (see page 514) 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 279) 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 (see page 514) 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 (see page 534) to search (see page 514) via Searchable Property metadata.

- **True** = Show extended search (see page 514)
- **False** = Hide (see page 142) extended search (see page 514)

Search (see page 279) type metadata only appears on the search (see page 514) screen if the following are true:

- this property is set to True
- the ShowSearchOptions (see page 299) property is set to True

- at least one content (see page 449) 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 (see page 514)'s starting folder.

Search.ShowSearchBoxAlways Property

If set to false, the search (see page 514) box does not appear on PostBack. The default is True.

- **True** = display search (see page 514) box on PostBack
- **False** = do not display search (see page 514) 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 (see page 514) 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 (see page 534) to override the folders to search (see page 514). For example, set to "Photos=1,NewsHeadlines=0" to search (see page 514) only the Photos folder by default, and allow the user (see page 534) 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 (see page 514) 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 301)	Method that can be overridden and used to grab data from the back end for each server control.

Legend

	Method

Sitemap.Fill Method

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

C#

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

Visual Basic

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

Sitemap Properties

The properties of the Sitemap class are listed here.

Public Properties

	Name	Description
	CacheInterval (see page 302)	Sets the cache interval in second(s).
	ClassName (see page 302)	Stylesheet (see page 305) class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
	DisplayType (see page 303)	Determines how the information on the page is displayed. • **None** - databind only • **List** - lists the title of every content (see page 449) block in the folder • **ListWithDescriptions** - lists the title of every content (see page 449) block in the folder plus the content (see page 449) summary • **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page
	DisplayXslt (see page 303)	The Xslt to use to render the control (optional).
	FlatTopLevel (see page 304)	Includes site nodes from the parent folder but not their items.
	FolderID (see page 304)	The ID of the folder you wish to select data from. Root is 0.
	MaxLevel (see page 304)	Get or set the maximum amount of levels to show. Set it to 0 (zero) for unlimited
	StartingLevel (see page 305)	Get or set to the starting level of the site map. If set to 0 (zero), starts from the root.
	Stylesheet (see page 305)	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 305)	Returns the XML object for the sitemap.

Legend

	Property
	virtual

Sitemap.CacheInterval Property

Sets the cache interval in second(s).

C#

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

Visual Basic

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

Sitemap.ClassName Property

Stylesheet (see page 305) class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.

C#

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

```
[Browsable(true)]
[Description("Stylesheet class to format the HTML. It must exists in
[site]/workarea/csslib/sitemap.css.\nLeave blank to accept default.")]
public string ClassName;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Browsable(true)>
<Description("Stylesheet class to format the HTML. It must exists in
[site]/workarea/csslib/sitemap.css.\nLeave blank to accept default.")>
Public Property ClassName() As string
```

Sitemap.DisplayType Property

Determines how the information on the page is displayed.

- **None** - databind only
- **List** - lists the title of every content (see page 449) block in the folder
- **ListWithDescriptions** - lists the title of every content (see page 449) block in the folder plus the content (see page 449) 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 get render on the page.")]
[Editor(typeof(SitemapDisplayTypeConverter), typeof(System.Drawing.Design.UITypeEditor))]
public string DisplayType;
```

Visual Basic

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

Sitemap.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 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))]
public int FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderID() As Integer
```

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 max level of the sitemap. Set it to 0 for unlimited.")]
public int MaxLevel;
```

Visual Basic

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

Sitemap.StartingLevel Property

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 246)-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 306)	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 307)	The Xslt to use to render the Menu (optional)
	Enable508Compliance (see page 307)	If you set to **True**, the menu will comply with Section 508. This makes the menus accessible to disabled users.
	Enable508NavigationHeadings (see page 307)	Set to **True** to enable accessibility (508-Compliant) navigation headings. This is only used if the Enable508Compliance (see page 307) property is set to True.
	EnableMouseOverPopUp (see page 308)	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 308)	If you set to **True**, any submenu on the Smart Menu (see page 246) can be set to open automatically. To learn about how to do this, see the Ektron CMS400.NET Administrator Manual section “Menu (see page 246) Feature” > “Assigning Folders to a Menu (see page 246).” If you set to **False**, even if all other conditions are present, submenus on a Smart Menu (see page 246) 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 308)	Set to **True** to enable auto-submenu opening based on link match (if all else fails).
	EnableSubmenuTemplateMatching (see page 309)	Set to **True** to enable auto-submenu opening based on a template filename match (if all else fails).
	LaunchLinksInNewWindow (see page 309)	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 310)	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 310)	Set to **True** if you want to a single edit button (and no add button) to appear on the menu when you are logged in. If you set to **False**, Edit Menu (see page 246) and Add Menu (see page 246) options appear at the bottom of every menu and submenu.
	StartCollapsed (see page 310)	If you set to **True**, all submenus are closed when the menu initially displays.
	StartWithRootCollapsed (see page 311)	If you set to **True**, all submenus are closed when the menu initially displays.
	Stylesheet (see page 311)	Enter the style sheet that will determine the appearance of the menus. Menu (see page 246) 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)

Legend

	Property

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 307) property is set to True.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to enable accessibility (508-Compliant) navigation headings (only used if
Enable508Compliance is true)")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool Enable508NavigationHeadings;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to enable accessibility (508-Compliant) navigation headings (only used if
Enable508Compliance is true)")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property Enable508NavigationHeadings() As bool
```

SmartMenu.EnableMouseOverPopUp Property

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 enable submenu activation (pop-ups) on mouse-over.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool EnableMouseOverPopUp;
```

Visual Basic

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

SmartMenu.EnableSmartOpen Property

If you set to **True**, any submenu on the Smart Menu (see page 246) can be set to open automatically. To learn about how to do this, see the Ektron CMS400.NET Administrator Manual section “Menu (see page 246) Feature” > “Assigning Folders to a Menu (see page 246).”

If you set to **False**, even if all other conditions are present, submenus on a Smart Menu (see page 246) 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 enable auto-submenu opening based on link match (if all else fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuLinkPathMatching;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to enable auto-submenu opening based on link match (if all else fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property EnableSubmenuLinkPathMatching() As bool
```

SmartMenu.EnableSubmenuTemplateMatching Property

Set to **True** to enable auto-submenu opening based on a template filename match (if all else fails).

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to enable auto-submenu opening based on a template filename match (if all
else fails).")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(true)]
[Browsable(false), EditorBrowsable(EditorBrowsableState.Never)]
public bool EnableSubmenuTemplateMatching;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to enable auto-submenu opening based on a template filename match (if all
else fails).")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(true)>
<Browsable(false), EditorBrowsable(EditorBrowsableState.Never)>
Public Property EnableSubmenuTemplateMatching() As bool
```

SmartMenu.LaunchLinksInNewWindow Property

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 to a single edit button (and no add button) to appear on the menu when you are logged in. If you set to **False**, Edit Menu (see page 246) and Add Menu (see page 246) options appear at the bottom of every menu and submenu.

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("Set to only render a single edit button (and no add button) when logged in.")]
[TypeConverter(typeof(BooleanConverter))]
[DefaultValue(false)]
public bool SingleEditButton;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("Set to only render a single edit button (and no add button) when logged in.")>
<TypeConverter(typeof(BooleanConverter))>
<DefaultValue(false)>
Public Property SingleEditButton() As bool
```

SmartMenu.StartCollapsed Property

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 246) 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
```

WebSearch Class

The Web Search (see page 279) server control lets you customize the behavior of the search (see page 514). You place this control on any Web form from which a site visitor can search (see page 514) your site. For information about the site visitor experience, see the Ektron CMS400.NET Administrator Manual section "Searching Your Web Site (see page 424)" > "Web Site (see page 424) Search (see page 279)."

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 WebSearch()
```

WebSearch Methods

The methods of the WebSearch class are listed here.

Public Methods

	Name	Description
	Fill (see page 312)	Method that can be overridden and used to grab data from the back end for each server control.
	LoadSearch (see page 312)	
V	SearchDisplayUI (see page 313)	
	Wrap (see page 313)	Method that can be overridden and used to wrap an Ektron toolbar corresponding to a particular Control around some Html text.

Legend

	Method
V	virtual

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.LoadSearch Method

C#

```
public int LoadSearch(SearchRequestData searchReqData);
```

Visual Basic

```
Public Function LoadSearch(searchReqData As SearchRequestData) As Integer
```

Parameters

Parameters	Description
SearchReq	

WebSearch.SearchDisplayUI Method

C#

```
public virtual string SearchDisplayUI(bool supressPostRedirect);
```

Visual Basic

```
Public virtual Function SearchDisplayUI(supressPostRedirect As bool) As string
```

Parameters

Parameters	Description
supressPostRedirect	

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 315)	Optional: If you want to display an image on the submit button, enter the server path to it.
	ButtonText (see page 316)	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.
	CustomSearch (see page 316)	Optional: If you want to display an image on the submit button, enter the server path to it.
	DisplayXslt (see page 316)	Determines the display of the search (see page 514) results page. • **None** - databind only • **ecmNavigation** - lists the title of every content (see page 449) item found by the search (see page 514) • **ecmTeaser** - lists the title and the summary of every content (see page 449) item found by the search (see page 514) • **ecmUnOrderedList** - sorts the list in no particular order. Shows the title and content (see page 449) summary • **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page
	EnableAdvancedLink (see page 317)	Set to **True** to display the link that makes the advanced properties visible.
	EndCount (see page 317)	Get the numerical record of the first item on a page.

	FolderID (see page 317)	The folder at which the search (see page 514) begins. The starting folder need not be the root folder. The **Recursive (see page 320)** property determines if the search (see page 514) examines this folder's subfolders.
	LinkTarget (see page 318)	Gets or sets the target window for the hyperlinks in the search (see page 514) result. The choices are: • **_blank** - loads the link in a new blank window. This window is not named. • **_self** - loads the link in the same window the anchor was clicked in. This is useful for overriding a globally assigned base target. • **_parent** - loads the link in the immediate frameset parent of the document. This defaults to acting like "_self" if the document has no parent. • **_top** - loads the link in the full body of the window. This defaults to acting like "_self" if the... more (see page 318)
	MarkupLanguage (see page 318)	Identify the template markup file that controls the display of the search (see page 514) 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 316) property value is ignored.
	MaxCharacters (see page 318)	The maximum number of characters the Search (see page 279) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 324) property to the same number.
	MaxTeaserLength (see page 319)	An optional result formatting parameter. If non-zero, then the matching content (see page 449) 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).
	OrderBy (see page 319)	The order of the search (see page 514) results. Choices are: • **Title** - The content (see page 449) title (alphabetical) • **ID** - The content (see page 449) ID number • **Date Created** - The date the content (see page 449) was created • **Date Modified** - The date the content (see page 449) was most recently modified • **Editor** - The user (see page 534) who last edited the content (see page 449) (alphabetical) • **Rank** - The rank assigned to the content (see page 449). See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 424)" > "Display of Search (see page 279) Results" > "Search (see page 279) Result Ranking"
	OrderDirection (see page 319)	Specify the order direction of the results. Choices are **Ascending** or **Descending**.
	Recursive (see page 320)	Select **True** if you wish to search (see page 514) sub-folders of the starting folder. The starting folder is identified in the `FolderID (see page 317)` property.
	RemoveTeaserHtml (see page 320)	An optional result formatting parameter. If set to **True**, any HTML in the content (see page 449) 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 320)	The number of search (see page 514) results. This is a read-only property.
	ResultDataTable (see page 321)	Allows a developer to get results from the search (see page 514) control as a DataTable
	ResultsPageSize (see page 321)	An optional result formatting parameter. If non-zero, then the matching content (see page 449) 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 321)	Lets you designate where search (see page 514) results appear. You can place search (see page 514) criteria in one area of a Web form and the results in another. For example, you have the following tag. `<span id="results"></span>` In this case, enter **results** for the property value.
	ResultXml (see page 322)	Allows a developer to get results from the search (see page 514) control as XML.

	SearchFor (see page 322)	Choose the type of content (see page 449) to search (see page 514). • All • HTML • Documents • Images • Multimedia • Discussion Forums
	SearchText (see page 322)	Get or set default search (see page 514) text. This is optional.
	ShowSearchBoxAlways (see page 323)	If set to **False**, the search (see page 514) box does not appear on PostBack. The default is True. **True** = Display search (see page 514) box on PostBack **False** = Do not display search (see page 514) box on PostBack
	ShowSearchOptions (see page 323)	If set to **True**, the following options appear above the Search (see page 279) box. • Site (see page 424) • Html • Documents • Images • Multimedia
	StartCount (see page 324)	Get the numerical record of the last item on a page.
	Stylesheet (see page 324)	Specify the location of the style sheet for the search (see page 514) 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.`
	TextBoxSize (see page 324)	The size of the Search (see page 279) text box for user (see page 534) input. The width is based on the number of characters.
	TimeTaken (see page 324)	Gets the amount of time, in seconds, it has taken to perform the search (see page 514).
	XmlDoc (see page 325)	This property is not supported by WebSearch (see page 311). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 316) property of the WebSearch (see page 311) server control. The XSLT filename is specified relative to the current directory.

Legend

	Property
	virtual

WebSearch.ButtonImgSrc Property

Optional: If you want to display an image on the submit button, enter the server path to it.

C#

```
[Bindable(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)>
<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)]
[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
```

WebSearch.CustomSearch Property

Optional: If you want to display an image on the submit button, enter the server path to it.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("")]
[Description("Optional: Custom directories to search.")]
public string CustomSearch;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("")>
<Description("Optional: Custom directories to search.")>
Public Property CustomSearch() As string
```

WebSearch.DisplayXslt Property

Determines the display of the search (see page 514) results page.

- **None** - databind only
- **ecmNavigation** - lists the title of every content (see page 449) item found by the search (see page 514)
- **ecmTeaser** - lists the title and the summary of every content (see page 449) item found by the search (see page 514)
- **ecmUnOrderedList** - sorts the list in no particular order. Shows the title and content (see page 449) summary
- **Path to Custom Xslt** - enter the path to an Xslt that determines the display of the page

C#

```
[Bindable(true)]
[Category("Ektron")]
[Description("The Xslt to use to render the Control (optional)")]
[TypeConverter(typeof(XsltTypeConverter))]
[DefaultValue("None - Databind Only")]
[Personalizable()]
public override string DisplayXslt;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<Description("The Xslt to use to render the Control (optional)")>
<TypeConverter(typeof(XsltTypeConverter))>
<DefaultValue("None - Databind Only")>
<Personalizable()>
Public override Property DisplayXslt() 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 (see page 514) begins. The starting folder need not be the root folder. The **Recursive (see page 320)** property determines if the search (see page 514) examines this folder's subfolders.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[Description("The ID of the folder you wish to select data from. Root is 0.")]
[Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))]
public int FolderID;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<Description("The ID of the folder you wish to select data from. Root is 0.")>
<Editor(typeof(DefaultFolderIDUITypeEditor), typeof(System.Drawing.Design.UITypeEditor))>
Public Property FolderID() As Integer
```

WebSearch.LinkTarget Property

Gets or sets the target window for the hyperlinks in the search (see page 514) 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), 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
```

WebSearch.MarkupLanguage Property

Identify the template markup file that controls the display of the search (see page 514) 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 316) property value is ignored.

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
```

WebSearch.MaxCharacters Property

The maximum number of characters the Search (see page 279) text box accepts. If you enter a value less than 50, it is a good idea to set the TextBoxSize (see page 324) property to the same number.

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
```

WebSearch.MaxTeaserLength Property

An optional result formatting parameter. If non-zero, then the matching content (see page 449) 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)]
[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)>
<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.OrderBy Property

The order of the search (see page 514) results. Choices are:

- **Title** - The content (see page 449) title (alphabetical)
- **ID** - The content (see page 449) ID number
- **Date Created** - The date the content (see page 449) was created
- **Date Modified** - The date the content (see page 449) was most recently modified
- **Editor** - The user (see page 534) who last edited the content (see page 449) (alphabetical)
- **Rank** - The rank assigned to the content (see page 449). See Also: Ektron CMS400.NET Administrator manual section "Searching Your Web Site (see page 424)" > "Display of Search (see page 279) Results" > "Search (see page 279) Result Ranking"

C#

```
[Browsable(true), Category("Ektron"), Description("Specify the order of the results. Choices
are: Rank, Title, User, DateCreated, DateModified")]
public WebSearchOrder OrderBy;
```

Visual Basic

```
<Browsable(true), Category("Ektron"), Description("Specify the order of the results. Choices
are: Rank, Title, User, DateCreated, DateModified")>
Public Property OrderBy() As WebSearchOrder
```

WebSearch.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
```

WebSearch.Recursive Property

Select **True** if you wish to search (see page 514) sub-folders of the starting folder. The starting folder is identified in the `FolderID` (see page 317) 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 (see page 449) 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)]
[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
```

WebSearch.ResultCount Property

The number of search (see page 514) results. This is a read-only property.

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue("0")]
[Browsable(true)]
[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)>
<Category("Ektron")>
<DefaultValue("0")>
<Browsable(true)>
<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 (see page 514) 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 (see page 449) 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)]
[Category("Ektron")]
[DefaultValue(10)]
[Description("An optional page size setting, defaults to web.config search provider page
size).")]
public int ResultsPageSize;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(10)>
<Description("An optional page size setting, defaults to web.config search provider page
size).")>
Public Property ResultsPageSize() As Integer
```

WebSearch.ResultTagId Property

Lets you designate where search (see page 514) results appear. You can place search (see page 514) 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)]
[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
```

WebSearch.ResultXml Property

Allows a developer to get results from the search (see page 514) 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 (see page 449) to search (see page 514).

- All
- HTML
- Documents
- Images
- Multimedia
- Discussion Forums

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(0)]
[Description("Choose the type of content to search. All - HTML - Documents - Images -
Multimedia - Discussion Forums")]
public SearchDocumentType SearchFor;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(0)>
<Description("Choose the type of content to search. All - HTML - Documents - Images -
Multimedia - Discussion Forums")>
Public Property SearchFor() As SearchDocumentType
```

WebSearch.SearchText Property

Get or set default search (see page 514) text. This is optional.

C#

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

Visual Basic

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

WebSearch.ShowSearchBoxAlways Property

If set to **False**, the search (see page 514) box does not appear on PostBack. The default is True.

True = Display search (see page 514) box on PostBack

False = Do not display search (see page 514) 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
```

WebSearch.ShowSearchOptions Property

If set to **True**, the following options appear above the Search (see page 279) box.

- Site (see page 424)
- Html
- Documents
- Images
- Multimedia

C#

```
[Bindable(true)]
[Category("Ektron")]
[DefaultValue(true)]
[Description("Select true to show type select over search box")]
public bool ShowSearchOptions;
```

Visual Basic

```
<Bindable(true)>
<Category("Ektron")>
<DefaultValue(true)>
<Description("Select true to show type select over search box")>
Public Property ShowSearchOptions() As bool
```

Description

public bool EnableCallback { get { return _enableCallback; } set { _enableCallback = value; } }

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 (see page 514) 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)]
[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
```

WebSearch.TextBoxSize Property

The size of the Search (see page 279) text box for user (see page 534) input. The width is based on the number of 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
```

WebSearch.TimeTaken Property

Gets the amount of time, in seconds, it has taken to perform the search (see page 514).

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 311). The underlying XML can be displayed using an XSLT specified by the DisplayXslt (see page 316) property of the WebSearch (see page 311) server control. The XSLT filename is specified relative to the current directory.

C#

```
[Bindable(true)]
[Category("Ektron")]
[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)>
<Category("Ektron")>
<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 (see page 363) 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 325)	This is namespace Ektron.Cms.Extensibility.

Ektron.Cms.Extensibility Namespace

This is namespace Ektron.Cms.Extensibility.

Classes

	Name	Description
	ContentInfo (see page 327)	Class which holds information about the content (see page 449) being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	ExtensionAuthor (see page 330)	Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in.
	ExtensionDescription (see page 332)	Custom attribute - Used at the method level to describe what a particular event does
	ExtensionEvent (see page 333)	Override the methods in this class to customize the CMS via event hooks.
	FolderInfo (see page 348)	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 350)	Class that describes a field in a form - name / value pair
	FormInfo (see page 351)	Class which describes a Form (see page 493) object
	RequestInfo (see page 353)	Reserved for future use.
	ReviewInfo (see page 353)	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 358)	Class which holds information about the site in which the event was fired.
	UserInfo (see page 359)	Class which holds information about the user (see page 534) 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.

Legend

	Class
	abstract

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	ContentInfo (see page 327)	Class which holds information about the content (see page 449) being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.
	ExtensionAuthor (see page 330)	Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in.
	ExtensionDescription (see page 332)	Custom attribute - Used at the method level to describe what a particular event does
	ExtensionEvent (see page 333)	Override the methods in this class to customize the CMS via event hooks.
	FolderInfo (see page 348)	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 350)	Class that describes a field in a form - name / value pair
	FormInfo (see page 351)	Class which describes a Form (see page 493) object
	RequestInfo (see page 353)	Reserved for future use.
	ReviewInfo (see page 353)	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 358)	Class which holds information about the site in which the event was fired.
	UserInfo (see page 359)	Class which holds information about the user (see page 534) 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.

Legend

	Class
	abstract

ContentInfo Class

Class which holds information about the content (see page 449) 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.ContentInfo

C#

```
public sealed class ContentInfo;
```

Visual Basic

```
Public sealed Class ContentInfo
```

File

CmsInfoObjects.cs

ContentInfo.ContentInfo Constructor

C#

```
public ContentInfo(XmlNode xml);
```

Visual Basic

```
Public Sub ContentInfo(xml As XmlNode)
```

Description

private string origXml;

ContentInfo Properties

The properties of the ContentInfo class are listed here.

Public Properties

	Name	Description
	AssetID (see page 328)	Gets a string containing the asset ID (see page 329) in the form of a GUID if the content (see page 449) is an asset. (ReadOnly)
	Comments (see page 328)	Gets a string containing the comments a user (see page 534) has entered during their operation. (ReadOnly)
	ErrorMessage (see page 328)	Gets or sets a string containing message to display to the end user (see page 534) if Plug-in returns false. (ReadWrite)
	FileAndPath (see page 328)	Gets a string containing the full physical path and file name on the server if content (see page 449) is an asset. (ReadOnly)
	FileExtension (see page 328)	Gets a string containing file extension if content (see page 449) is an asset. (ReadOnly)
	FileName (see page 329)	Gets a string containing file name if content (see page 449) is an asset. (ReadOnly)
	FolderID (see page 329)	Gets an integer representing the folder ID (see page 329) which this content (see page 449) is located. (ReadOnly)
	Html (see page 329)	Gets or Sets a string containing all the html displayed to end users for a piece of content (see page 449). (ReadWrite Note: This property only allows write access within OnBeforePublish.)
	ID (see page 329)	Gets an integer representing the content (see page 449) ID of the content (see page 449). (ReadOnly)
	Language (see page 329)	Gets an integer representing the language of the content (see page 449). (ReadOnly)

	MimeType (see page 329)	Gets a string containing the MIME type if content (see page 449) is an asset. (ReadOnly)
	Status (see page 330)	Gets a string representing the status of the content (see page 449), published, checkedOut. (ReadOnly)
	Title (see page 330)	Gets a string containing the title of the content (see page 449). (ReadOnly)
	Type (see page 330)	Gets an integer representing the content (see page 449) type. (ReadOnly)
	UserID (see page 330)	Gets an integer representing the user (see page 534) ID (see page 329) of the user (see page 534) performing the operation on the content (see page 449). (ReadOnly)

Legend

	Property

ContentInfo.AssetID Property

Gets a string containing the asset ID (see page 329) in the form of a GUID if the content (see page 449) 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 (see page 534) 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 (see page 534) if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

ContentInfo.FileAndPath Property

Gets a string containing the full physical path and file name on the server if content (see page 449) is an asset. (ReadOnly)

C#

```
public string FileAndPath;
```

Visual Basic

```
Public Property FileAndPath() As string
```

ContentInfo.FileExtension Property

Gets a string containing file extension if content (see page 449) is an asset. (ReadOnly)

C#

```
public string FileExtension;
```

Visual Basic

```
Public Property FileExtension() As string
```

ContentInfo.FileName Property

Gets a string containing file name if content (see page 449) is an asset. (ReadOnly)

C#

```
public string FileName;
```

Visual Basic

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

ContentInfo.FolderID Property

Gets an integer representing the folder ID (see page 329) which this content (see page 449) is located. (ReadOnly)

C#

```
public int FolderID;
```

Visual Basic

```
Public Property FolderID() As Integer
```

ContentInfo.Html Property

Gets or Sets a string containing all the html displayed to end users for a piece of content (see page 449). (ReadWrite Note: This property only allows write access within OnBeforePublish.)

C#

```
public string Html;
```

Visual Basic

```
Public Property Html() As string
```

ContentInfo.ID Property

Gets an integer representing the content (see page 449) ID of the content (see page 449). (ReadOnly)

C#

```
public int ID;
```

Visual Basic

```
Public Property ID() As Integer
```

ContentInfo.Language Property

Gets an integer representing the language of the content (see page 449). (ReadOnly)

C#

```
public int Language;
```

Visual Basic

```
Public Property Language() As Integer
```

ContentInfo.MimeType Property

Gets a string containing the MIME type if content (see page 449) is an asset. (ReadOnly)

C#

```
public string MimeType;
```

Visual Basic

```
Public Property MimeType() As string
```

ContentInfo.Status Property

Gets a string representing the status of the content (see page 449), published, checkedOut. (ReadOnly)

C#

```
public string Status;
```

Visual Basic

```
Public Property Status() As string
```

ContentInfo.Title Property

Gets a string containing the title of the content (see page 449). (ReadOnly)

C#

```
public string Title;
```

Visual Basic

```
Public Property Title() As string
```

ContentInfo.Type Property

Gets an integer representing the content (see page 449) type. (ReadOnly)

C#

```
public int Type;
```

Visual Basic

```
Public Property Type() As Integer
```

ContentInfo.UserID Property

Gets an integer representing the user (see page 534) ID (see page 329) of the user (see page 534) performing the operation on the content (see page 449). (ReadOnly)

C#

```
public int UserID;
```

Visual Basic

```
Public Property UserID() As Integer
```

ExtensionAuthor Class

Custom attribute - Used at the method and/or assembly level to indicate who wrote the plug-in.

Class Hierarchy

C#

```
[AttributeUsage(AttributeTargets.Method | AttributeTargets.Assembly)]
public class ExtensionAuthor : Attribute;
```

Visual Basic

```
<AttributeUsage(AttributeTargets.Method | AttributeTargets.Assembly)>
Public Class ExtensionAuthor
Inherits Attribute
```

File

ExtensibilityBase.cs

ExtensionAuthor.ExtensionAuthor Constructor

Sets the creator of the Extension.

C#

```
public ExtensionAuthor(string author);
```

Visual Basic

```
Public Sub ExtensionAuthor(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 331)	Converts author name to a string.

Legend

	Method

ExtensionAuthor.ToString Method

Converts author name to a string.

C#

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

Visual Basic

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

Returns

String represnting the author's name.

ExtensionAuthor Properties

The properties of the ExtensionAuthor class are listed here.

Public Properties

	Name	Description
	Author (see page 331)	Gets the creator of the Extension. (ReadOnly)

Legend

	Property

ExtensionAuthor.Author Property

Gets the creator of the Extension. (ReadOnly)

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 ExtensionDescription(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 332)	Converts the description to a string.

Legend

	Method

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 333)	Gets the description of the extension. (ReadOnly)

Legend

	Property

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 335)	Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 593) Web Service.
	GetAssetProxy (see page 335)	Returns an AssetProxy object to access the Asset (see page 450) Web Service.

	GetBlogProxy (see page 336)	Returns a BlogProxy object to access the Blog (see page 454) Web Service.
	GetBusinessRulesProxy (see page 336)	Returns a BusinessRulesProxy object to access the BusinessRules (see page 61) Web Service.
	GetCalendarEventTypeProxy (see page 336)	Returns a CalendarEventTypeProxy object to access the CalendarEventType (see page 449) Web Service.
	GetCalendarProxy (see page 336)	Returns a CalendarProxy object to access the Calendar (see page 447) Web Service.
	GetContentProxy (see page 337)	Returns a ContentProxy object to access the Content (see page 347) Web Service.
	GetContentRatingProxy (see page 337)	Returns a ContentRatingProxy object to access the ContentRating (see page 488) Web Service.
	GetCustomFieldsProxy (see page 337)	Returns a CustomFieldsProxy object to access the CustomFields (see page 367) Web Service.
	GetFolderProxy (see page 337)	Returns a FolderProxy object to access the Folder (see page 347) Web Service.
	GetFontProxy (see page 337)	Returns a FontProxy object to access the Font (see page 392) Web Service.
	GetFormProxy (see page 338)	Returns a FormProxy object to access the Form (see page 348) Web Service.
	GetLibraryProxy (see page 338)	Returns a LibraryProxy object to access the Library (see page 395) Web Service.
	GetMetadataProxy (see page 338)	Returns a MetadataProxy object to access the Metadata (see page 412) Web Service.
	GetObjectStateAsXml (see page 338)	Retrieves changes made by individual plug-ins
	GetPermissionsProxy (see page 339)	Returns a PermissionsProxy object to access the Permissions (see page 419) Web Service.
	GetSearchManagerProxy (see page 339)	Returns a SearchManagerProxy object to access the SearchManager (see page 515) Web Service.
	GetSiteMapProxy (see page 339)	Returns a SiteMapProxy object to access the SiteMap (see page 430) Web Service.
	GetSiteProxy (see page 339)	Returns a SiteProxy object to access the Site (see page 348) Web Service.
	GetTaskCategoryProxy (see page 340)	Returns a TaskCategoryProxy object to access the Task (see page 519) Category Web Service.
	GetTaskCategoryTypeProxy (see page 340)	Returns a TaskProxy object to access the Task (see page 519) Web Service.
	GetTaskProxy (see page 340)	Returns a TaskProxy object to access the Task (see page 519) Web Service.
	GetThreadedDiscussionProxy (see page 340)	Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 503) Web Service.
	GetUserProxy (see page 340)	Returns a UserProxy object to access the User (see page 348) Web Service.
	OnAfterContentAdd (see page 341)	This hook is called after content (see page 449) is added in the CMS.
	OnAfterContentDelete (see page 341)	This hook is called after a delete of content (see page 449) is completed in the CMS.
	OnAfterContentEdit (see page 341)	This hook is called after an edit is completed in the CMS.
	OnAfterFolderAdd (see page 341)	This hook is called after a folder is created in the CMS.
	OnAfterFolderDelete (see page 342)	This hook is called after a folder is deleted in the CMS.
	OnAfterFolderEdit (see page 342)	This hook is called after a folder edit is completed in the CMS.
	OnAfterFormSubmit (see page 342)	This hook is called after form data is submitted to the CMS database.
	OnAfterPublish (see page 342)	This hook is called after a publish is completed in the CMS.
	OnAfterReviewRejected (see page 343)	This hook is called after a review is rejected.
	OnAfterUserAdd (see page 343)	This hook is called after a user (see page 534) is created in the CMS.

	OnAfterUserDelete (see page 343)	This hook is called after a user (see page 534) is deleted from the CMS.
	OnAfterUserEdit (see page 343)	This hook is called after a user (see page 534) is edited in the CMS.
	OnAfterUserLogin (see page 344)	This hook is called after a user (see page 534) logs into in the CMS.
	OnBeforeContentAdd (see page 344)	This hook is called before content (see page 449) is added to the CMS.
	OnBeforeContentDelete (see page 344)	This hook is called before a delete of content (see page 449) in the CMS.
	OnBeforeContentEdit (see page 344)	This hook is called before a content (see page 449) is edited in the CMS.
	OnBeforeFolderAdd (see page 344)	This hook is called before a folder is created in the CMS.
	OnBeforeFolderDelete (see page 345)	This hook is called before a folder is deleted from the CMS.
	OnBeforeFolderEdit (see page 345)	This hook is called before a folder is edited in the CMS.
	OnBeforeFormSubmit (see page 345)	This hook is called before form data is submitted to the CMS database.
	OnBeforePublish (see page 345)	This hook is called before a publish takes place in the CMS.
	OnBeforeReviewRejected (see page 346)	This hook is called before a review is rejected.
	OnBeforeUserAdd (see page 346)	This hook is called before a user (see page 534) is created in the CMS.
	OnBeforeUserDelete (see page 346)	This hook is called before a user (see page 534) is delete from the CMS.
	OnBeforeUserEdit (see page 346)	This hook is called before a user (see page 534) is edited in the CMS.
	OnBeforeUserLogin (see page 347)	This hook is called before a user (see page 534) login is initiated in the CMS.

Legend

	Method
	virtual

ExtensionEvent.GetActiveDirectoryUserProxy Method

Returns an ActiveDirectoryUserProxy object to access the ActiveDirectoryUser (see page 593) 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 450) 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 454) 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 61) Web Service.

C#

```
public BusinessRulesProxy GetBusinessRulesProxy();
```

Visual Basic

```
Public Function GetBusinessRulesProxy() As BusinessRulesProxy
```

Returns

BusinessRulesProxy

ExtensionEvent.GetCalendarEventTypeProxy Method

Returns a CalendarEventTypeProxy object to access the CalendarEventType (see page 449) 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 447) Web Service.

C#

```
public CalendarProxy GetCalendarProxy();
```

Visual Basic

```
Public Function GetCalendarProxy() As CalendarProxy
```

Returns

CalendarProxy

ExtensionEvent.GetContentProxy Method

Returns a ContentProxy object to access the Content (see page 347) 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 488) 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 367) Web Service.

C#

```
public CustomFieldsProxy GetCustomFieldsProxy();
```

Visual Basic

```
Public Function GetCustomFieldsProxy() As CustomFieldsProxy
```

Returns

CustomFieldsProxy

ExtensionEvent.GetFolderProxy Method

Returns a FolderProxy object to access the Folder (see page 347) 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 392) 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 348) Web Service.

C#

```
public FormProxy GetFormProxy();
```

Visual Basic

```
Public Function GetFormProxy() As FormProxy
```

Returns

FormProxy

ExtensionEvent.GetLibraryProxy Method

Returns a LibraryProxy object to access the Library (see page 395) Web Service.

C#

```
public LibraryProxy GetLibraryProxy();
```

Visual Basic

```
Public Function GetLibraryProxy() As LibraryProxy
```

Returns

LibraryProxy

ExtensionEvent.GetMetadataProxy Method

Returns a MetadataProxy object to access the Metadata (see page 412) 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 419) 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 515) 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 430) 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 348) Web Service.

C#

```
public SiteProxy GetSiteProxy();
```

Visual Basic

```
Public Function GetSiteProxy() As SiteProxy
```

Returns

SiteProxy

ExtensionEvent.GetTaskCategoryProxy Method

Returns a TaskCategoryProxy object to access the Task (see page 519) 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 519) 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 519) Web Service.

C#

```
public TaskProxy GetTaskProxy();
```

Visual Basic

```
Public Function GetTaskProxy() As TaskProxy
```

Returns

TaskProxy

ExtensionEvent.GetThreadedDiscussionProxy Method

Returns a ThreadedDiscussionProxy object to access the ThreadedDiscussion (see page 503) 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 348) Web Service.

C#

```
public UserProxy GetUserProxy();
```

Visual Basic

```
Public Function GetUserProxy() As UserProxy
```

Returns

UserProxy

ExtensionEvent.OnAfterContentAdd Method

This hook is called after content (see page 449) is added in the CMS.

C#

```
public virtual bool OnAfterContentAdd();
```

Visual Basic

```
Public virtual Function OnAfterContentAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterContentDelete Method

This hook is called after a delete of content (see page 449) is completed in the CMS.

C#

```
public virtual bool OnAfterContentDelete();
```

Visual Basic

```
Public virtual Function OnAfterContentDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterContentEdit Method

This hook is called after an edit is completed in the CMS.

C#

```
public virtual bool OnAfterContentEdit();
```

Visual Basic

```
Public virtual Function OnAfterContentEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterFolderAdd Method

This hook is called after a folder is created in the CMS.

C#

```
public virtual bool OnAfterFolderAdd();
```

Visual Basic

```
Public virtual Function OnAfterFolderAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterFolderDelete Method

This hook is called after a folder is deleted in the CMS.

C#

```
public virtual bool OnAfterFolderDelete();
```

Visual Basic

```
Public virtual Function OnAfterFolderDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterFolderEdit Method

This hook is called after a folder edit is completed in the CMS.

C#

```
public virtual bool OnAfterFolderEdit();
```

Visual Basic

```
Public virtual Function OnAfterFolderEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterFormSubmit Method

This hook is called after form data is submitted to the CMS database.

C#

```
public virtual bool OnAfterFormSubmit();
```

Visual Basic

```
Public virtual Function OnAfterFormSubmit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnAfterPublish Method

This hook is called after a publish is completed in the CMS.

C#

```
public virtual bool OnAfterPublish();
```

Visual Basic

```
Public virtual Function OnAfterPublish() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.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 (see page 534) 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 (see page 534) 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 (see page 534) 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 (see page 534) logs into in the CMS.

C#

```
public virtual bool OnAfterUserLogin();
```

Visual Basic

```
Public virtual Function OnAfterUserLogin() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeContentAdd Method

This hook is called before content (see page 449) is added to the CMS.

C#

```
public virtual bool OnBeforeContentAdd();
```

Visual Basic

```
Public virtual Function OnBeforeContentAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeContentDelete Method

This hook is called before a delete of content (see page 449) in the CMS.

C#

```
public virtual bool OnBeforeContentDelete();
```

Visual Basic

```
Public virtual Function OnBeforeContentDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeContentEdit Method

This hook is called before a content (see page 449) is edited in the CMS.

C#

```
public virtual bool OnBeforeContentEdit();
```

Visual Basic

```
Public virtual Function OnBeforeContentEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeFolderAdd Method

This hook is called before a folder is created in the CMS.

C#

```
public virtual bool OnBeforeFolderAdd();
```

Visual Basic

```
Public virtual Function OnBeforeFolderAdd() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeFolderDelete Method

This hook is called before a folder is deleted from the CMS.

C#

```
public virtual bool OnBeforeFolderDelete();
```

Visual Basic

```
Public virtual Function OnBeforeFolderDelete() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeFolderEdit Method

This hook is called before a folder is edited in the CMS.

C#

```
public virtual bool OnBeforeFolderEdit();
```

Visual Basic

```
Public virtual Function OnBeforeFolderEdit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforeFormSubmit Method

This hook is called before form data is submitted to the CMS database.

C#

```
public virtual bool OnBeforeFormSubmit();
```

Visual Basic

```
Public virtual Function OnBeforeFormSubmit() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.OnBeforePublish Method

This hook is called before a publish takes place in the CMS.

C#

```
public virtual bool OnBeforePublish();
```

Visual Basic

```
Public virtual Function OnBeforePublish() As bool
```

Returns

A bool indicating success or failure. False indicates failure.

ExtensionEvent.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 (see page 534) 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 (see page 534) 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 (see page 534) 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 (see page 534) 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 347)	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 347)	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 348)	Returns object containing information about the Form, when the operation is form based.
	Site (see page 348)	Returns object containing information about the Site that fired the event.
	User (see page 348)	Returns object containing information about the User, when the operation is user based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

Legend

	Property
	protected

ExtensionEvent.Content Property

Returns object containing information about the Content, when the operation is content based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

C#

```
protected ContentInfo Content;
```

Visual Basic

```
Protected Property Content() As ContentInfo
```

ExtensionEvent.Folder Property

Returns object containing information about the Folder, when the operation is folder based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

C#

```
protected FolderInfo Folder;
```

Visual Basic

```
Protected Property Folder() As FolderInfo
```

ExtensionEvent.Form Property

Returns object containing information about the Form, when the operation is form based.

C#

```
protected FormInfo Form;
```

Visual Basic

```
Protected Property Form() As FormInfo
```

ExtensionEvent.Site Property

Returns object containing information about the Site that fired the event.

C#

```
protected SiteInfo Site;
```

Visual Basic

```
Protected Property Site() As SiteInfo
```

ExtensionEvent.User Property

Returns object containing information about the User, when the operation is user based. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this object. Data types default to the following: int = -1. string = "". bool = false.

C#

```
protected UserInfo User;
```

Visual Basic

```
Protected Property User() As UserInfo
```

FolderInfo Class

Class which holds information about the folder being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.FolderInfo

C#

```
public sealed class FolderInfo;
```

Visual Basic

```
Public sealed Class FolderInfo
```

File

CmsInfoObjects.cs

FolderInfo Properties

The properties of the FolderInfo class are listed here.

Public Properties

	Name	Description
	Description (see page 349)	Gets a string containing the description of the folder. (ReadOnly)
	ErrorMessage (see page 349)	Gets or sets a string containing message to display to the end user (see page 534) if Plug-in returns false. (ReadWrite)
	ID (see page 349)	Gets an integer representing the folder ID of the folder. (ReadOnly)
	Name (see page 349)	Gets a string containing the folder name. (ReadOnly)
	ParentID (see page 350)	Gets an integer representing the folder ID (see page 349) of the parent folder. (ReadOnly)
	ReplicationState (see page 350)	Gets an integer representing if replication is turned on for this folder. (ReadOnly)
	StyleSheet (see page 350)	Gets a string containing the style sheet to be applied to content (see page 449) in this folder. (ReadOnly)
	Type (see page 350)	Gets a string containing the folder type. (ReadOnly)

Legend

	Property

FolderInfo.Description Property

Gets a string containing the description of the folder. (ReadOnly)

C#

```
public string Description;
```

Visual Basic

```
Public Property Description() As string
```

FolderInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user (see page 534) if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

FolderInfo.ID Property

Gets an integer representing the folder ID of the folder. (ReadOnly)

C#

```
public int ID;
```

Visual Basic

```
Public Property ID() As Integer
```

FolderInfo.Name Property

Gets a string containing the folder name. (ReadOnly)

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

FolderInfo.ParentID Property

Gets an integer representing the folder ID (see page 349) of the parent folder. (ReadOnly)

C#

```
public int ParentID;
```

Visual Basic

```
Public Property ParentID() As Integer
```

FolderInfo.ReplicationState Property

Gets an integer representing if replication is turned on for this folder. (ReadOnly)

C#

```
public int ReplicationState;
```

Visual Basic

```
Public Property ReplicationState() As Integer
```

FolderInfo.StyleSheet Property

Gets a string containing the style sheet to be applied to content (see page 449) in this folder. (ReadOnly)

C#

```
public string StyleSheet;
```

Visual Basic

```
Public Property StyleSheet() As string
```

FolderInfo.Type Property

Gets a string containing the folder type. (ReadOnly)

C#

```
public string Type;
```

Visual Basic

```
Public Property Type() As string
```

FormField Class

Class that describes a field in a form - name / value pair

Class Hierarchy

Ektron.Cms.Extensibility.FormField

C#

```
public sealed class FormField;
```

Visual Basic

```
Public sealed Class FormField
```

File

CmsInfoObjects.cs

FormField.FormField Constructor

C#

```
public FormField(string FieldName, string FieldValue);
```

Visual Basic

```
Public Sub FormField(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 351)	Gets the name of the field as a string
	FieldValue (see page 351)	Gets the value of the field as a string

Legend

	Property

FormField.FieldName Property

Gets the name of the field as a string

C#

```
public string FieldName;
```

Visual Basic

```
Public Property FieldName() As string
```

FormField.FieldValue Property

Gets the value of the field as a string

C#

```
public string FieldValue;
```

Visual Basic

```
Public Property FieldValue() As string
```

FormInfo Class

Class which describes a Form (see page 493) 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 352)	Gets a single FormField (see page 350) object indexed by Field Name

Legend

	Indexer

FormInfo.this Indexer

Gets a single FormField (see page 350) 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 352)	Gets or sets a string containing message to display to the end user (see page 534) if plug-in returns false. (ReadWrite)
	FormDescription (see page 353)	Gets the description of the form as a string
	FormFields (see page 353)	Gets a collection of the Form (see page 493) Field objects as a System.Collections.Generic.List
	FormTitle (see page 353)	Gets the title for the form as a string

Legend

	Property

FormInfo.ErrorMessage Property

Gets or sets a string containing message to display to the end user (see page 534) 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 493) 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
```

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 354)	Gets an decimal representing the average rating. (ReadOnly)
	ContentHtml (see page 355)	Gets a string containing all the html displayed to end users for a piece of content (see page 449). (ReadOnly)
	ContentID (see page 355)	Gets an integer representing the content (see page 449) ID (see page 355) of the content (see page 449). (ReadOnly)
	ContentLanguage (see page 355)	Gets an integer representing the language of the content (see page 449). (ReadOnly)
	ContentTitle (see page 355)	Gets a string containing the title of the content (see page 449). (ReadOnly)
	ErrorMessage (see page 355)	Gets or sets a string containing message to display to the end user (see page 534) if Plug-in returns false. (ReadWrite)
	ID (see page 355)	Gets an integer representing the review ID. (ReadOnly)
	NumberofRatings (see page 356)	Gets an integer representing the number of ratings. (ReadOnly)
	Rating (see page 356)	Gets an integer representing the rating. (ReadOnly)
	RatingDate (see page 356)	Gets an datetime representing the rating time. (ReadOnly)
	State (see page 356)	Gets an integer representing the state. (ReadOnly)
	UserAvatar (see page 356)	Gets an string representing path of the user (see page 534)'s avatar. (ReadOnly)
	UserComments (see page 356)	Gets an string representing the review text. (ReadOnly)
	UserDisplayName (see page 357)	Gets an string representing the user (see page 534)'s display name. (ReadOnly)
	UserFirstName (see page 357)	Gets an string representing the user (see page 534)'s first name. (ReadOnly)
	UserID (see page 357)	Gets an integer representing the user (see page 534) ID (see page 355) of the user (see page 534). (ReadOnly)
	UserLastName (see page 357)	Gets an string representing the user (see page 534)'s last name. (ReadOnly)
	UserName (see page 357)	Gets an string representing the user (see page 534)'s CMS user (see page 534) name. (ReadOnly)
	VisitorID (see page 357)	Gets an string representing the cookie value of the visitor. (ReadOnly)

Legend

	Property

ReviewInfo.AverageRating Property

Gets an decimal representing the average rating. (ReadOnly)

C#

```
public decimal AverageRating;
```

Visual Basic

```
Public Property AverageRating() As decimal
```

Description

---------- review properties

ReviewInfo.ContentHtml Property

Gets a string containing all the html displayed to end users for a piece of content (see page 449). (ReadOnly)

C#

```
public string ContentHtml;
```

Visual Basic

```
Public Property ContentHtml() As string
```

Description

---------- content (see page 449) properties

ReviewInfo.ContentID Property

Gets an integer representing the content (see page 449) ID (see page 355) of the content (see page 449). (ReadOnly)

C#

```
public int ContentID;
```

Visual Basic

```
Public Property ContentID() As Integer
```

ReviewInfo.ContentLanguage Property

Gets an integer representing the language of the content (see page 449). (ReadOnly)

C#

```
public int ContentLanguage;
```

Visual Basic

```
Public Property ContentLanguage() As Integer
```

ReviewInfo.ContentTitle Property

Gets a string containing the title of the content (see page 449). (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 (see page 534) if Plug-in returns false. (ReadWrite)

C#

```
public string ErrorMessage;
```

Visual Basic

```
Public Property ErrorMessage() As string
```

ReviewInfo.ID Property

Gets an integer representing the review ID. (ReadOnly)

C#

```
public int ID;
```

Visual Basic

```
Public Property ID() As Integer
```

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 an 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 (see page 534)'s avatar. (ReadOnly)

C#

```
public string UserAvatar;
```

Visual Basic

```
Public Property UserAvatar() As string
```

Description

---------- user (see page 534) properties

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 (see page 534)'s display name. (ReadOnly)

C#

```
public string UserDisplayName;
```

Visual Basic

```
Public Property UserDisplayName() As string
```

ReviewInfo.UserFirstName Property

Gets an string representing the user (see page 534)'s first name. (ReadOnly)

C#

```
public string UserFirstName;
```

Visual Basic

```
Public Property UserFirstName() As string
```

ReviewInfo.UserID Property

Gets an integer representing the user (see page 534) ID (see page 355) of the user (see page 534). (ReadOnly)

C#

```
public int UserID;
```

Visual Basic

```
Public Property UserID() As Integer
```

ReviewInfo.UserLastName Property

Gets an string representing the user (see page 534)'s last name. (ReadOnly)

C#

```
public string UserLastName;
```

Visual Basic

```
Public Property UserLastName() As string
```

ReviewInfo.UserName Property

Gets an string representing the user (see page 534)'s CMS user (see page 534) 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 358)	Gets a string containing the full application path of asp.net appication (ReadOnly)
	IsSSL (see page 358)	Gets a bool indicating if calling site uses SSL. (ReadOnly)
	Name (see page 359)	Gets a string containing the name of the calling site. (ReadOnly)
	PhysicalSitePath (see page 359)	Gets a string containing the physical path of the application (ReadOnly)
	Port (see page 359)	Gets an integer representing the port the calling site is running on. (ReadOnly)
	Url (see page 359)	Gets a string containing the URL of the calling site. (ReadOnly)

Legend

	Property

SiteInfo.AppPath Property

Gets a string containing the full application path of asp.net appication (ReadOnly)

C#

```
public string AppPath;
```

Visual Basic

```
Public Property AppPath() As string
```

SiteInfo.IsSSL Property

Gets a bool indicating if calling site uses SSL. (ReadOnly)

C#

```
public bool IsSSL;
```

Visual Basic

```
Public Property IsSSL() As bool
```

SiteInfo.Name Property

Gets a string containing the name of the calling site. (ReadOnly)

C#

```
public string Name;
```

Visual Basic

```
Public Property Name() As string
```

SiteInfo.PhysicalSitePath Property

Gets a string containing the physical path of the application (ReadOnly)

C#

```
public string PhysicalSitePath;
```

Visual Basic

```
Public Property PhysicalSitePath() As string
```

SiteInfo.Port Property

Gets an integer representing the port the calling site is running on. (ReadOnly)

C#

```
public int Port;
```

Visual Basic

```
Public Property Port() As Integer
```

SiteInfo.Url Property

Gets a string containing the URL of the calling site. (ReadOnly)

C#

```
public string Url;
```

Visual Basic

```
Public Property Url() As string
```

UserInfo Class

Class which holds information about the user (see page 534) being processed. Certain hooks do not expose all information from the CMS, default values are assigned to the properties of this class. Data types default to the following: int = -1. string = "". bool = false.

Class Hierarchy

Ektron.Cms.Extensibility.UserInfo

C#

```
public sealed class UserInfo;
```

Visual Basic

```
Public sealed Class UserInfo
```

File

CmsInfoObjects.cs

UserInfo Properties

The properties of the UserInfo class are listed here.

Public Properties

	Name	Description
	DisplayName (see page 360)	Gets a string containing the display name of the user (see page 534) being added, edited or deleted. (ReadOnly)
	Editor (see page 360)	Gets a string containing the editor preference of the user (see page 534) being added, edited or deleted. (ReadOnly)
	EmailAddress (see page 360)	Gets a string containing the email address of the user (see page 534) being added, edited or deleted. (ReadOnly)
	ErrorMessage (see page 361)	Gets or sets a string containing message to display to the end user (see page 534) if Plug-in returns false. (ReadWrite)
	FirstName (see page 361)	Gets a string containing the first name of the user (see page 534) being added, edited or deleted. (ReadOnly)
	ID (see page 361)	Gets an integer representing the user (see page 534) ID of the being added, edited or deleted. (ReadOnly)
	Language (see page 361)	Gets a string containing the language preference of the user (see page 534) being added, edited or deleted. (ReadOnly)
	LastName (see page 361)	Gets a string containing the last name of the user (see page 534) being added, edited or deleted. (ReadOnly)
	Notifications (see page 361)	Gets an integer representing if the user (see page 534) has turned on email notifications of the user (see page 534) being added, edited or deleted. (ReadOnly)
	Type (see page 362)	Integer representing the user (see page 534) type of the user (see page 534) being added, edited or deleted. Only valid with OnBeforeUserAdd and OnAfterUserAdd. (ReadOnly)
	UserName (see page 362)	Gets a string containing the login name of the user (see page 534) being added, edited or deleted. (ReadOnly)

Legend

	Property

UserInfo.DisplayName Property

Gets a string containing the display name of the user (see page 534) 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 (see page 534) 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 (see page 534) 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 (see page 534) 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 (see page 534) being added, edited or deleted. (ReadOnly)

C#

```
public string FirstName;
```

Visual Basic

```
Public Property FirstName() As string
```

UserInfo.ID Property

Gets an integer representing the user (see page 534) ID of the being added, edited or deleted. (ReadOnly)

C#

```
public int ID;
```

Visual Basic

```
Public Property ID() As Integer
```

UserInfo.Language Property

Gets a string containing the language preference of the user (see page 534) 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 (see page 534) being added, edited or deleted. (ReadOnly)

C#

```
public string LastName;
```

Visual Basic

```
Public Property LastName() As string
```

UserInfo.Notifications Property

Gets an integer representing if the user (see page 534) has turned on email notifications of the user (see page 534) being added, edited or deleted. (ReadOnly)

C#

```
public int Notifications;
```

Visual Basic

```
Public Property Notifications() As Integer
```

UserInfo.Type Property

Integer representing the user (see page 534) type of the user (see page 534) 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 (see page 534) 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 363) Namespace.

- Ektron.Cms.API.BusinessRule (see page 365)
- Ektron.Cms.API.CustomFields (see page 367)
- Ektron.Cms.API.DateTimeSelector (see page 377)
- Ektron.Cms.API.Folder (see page 383)
- Ektron.Cms.API.Font (see page 392)
- Ektron.Cms.API.Library (see page 395)
- Ektron.Cms.API.Localization (see page 403)
- Ektron.Cms.API.Metadata (see page 412)
- Ektron.Cms.API.Permissions (see page 419)
- Ektron.Cms.API.Site (see page 424)
- Ektron.Cms.API.SiteMap (see page 430)
- Ektron.Cms.API.Subscription (see page 435)
- Ektron.Cms.API.Utilities (see page 441)

The rest of the subsections contain different Namespaces with classes below them.

- Ektron.Cms.API.Analytics.Analytics (see page 443)
- Ektron.Cms.API.Calendar.Calendar (see page 448)
- Ektron.Cms.API.Calendar.CalendarEvent (see page 448)
- Ektron.Cms.API.Calendar.CalendarEventType (see page 449)
- Ektron.Cms.API.Content.Asset (see page 450)
- Ektron.Cms.API.Content.Blog (see page 454)
- Ektron.Cms.API.Content.Content (see page 468)
- Ektron.Cms.API.Content.ContentRating (see page 488)
- Ektron.Cms.API.Content.Form (see page 493)
- Ektron.Cms.API.Content.Taxonomy (see page 498)
- Ektron.Cms.API.Content.ThreadedDiscussion (see page 503)
- Ektron.Cms.API.Search.SearchManager (see page 515)
- Ektron.Cms.API.Task.Task (see page 520)
- Ektron.Cms.API.Task.TaskCategory (see page 530)
- Ektron.Cms.API.Task.TaskCategoryType (see page 532)
- Ektron.Cms.API.User.ADUser (see page 534)
- Ektron.Cms.API.User.User (see page 542)

Namespaces

Name	Description
API (see page 363)	The API Namespace contains classes for programmatically accessing features typically found in the Workarea.

API Namespace

The API Namespace contains classes for programmatically accessing features typically found in the Workarea.

Classes

	Name	Description
	BusinessRule (see page 365)	Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example: • **Condition** - a site visitor visits your site for the first time • **Action** - add a cookie to the site visitor's system and pop up a welcome message With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 365)
	CustomFields (see page 367)	This is class API.CustomFields.
	DateTimeSelector (see page 377)	This is class API.DateTimeSelector.
	Folder (see page 383)	This is class API.Folder.
	Font (see page 392)	This is class API.Font.
	Library (see page 395)	This is class API.Library.
	Localization (see page 403)	This is class API.Localization.
	Metadata (see page 412)	This is class API.Metadata.

	Permissions (see page 419)	This is class API.Permissions.
	Site (see page 424)	This is class API.Site.
	SiteMap (see page 430)	This is class API.SiteMap.
	Subscription (see page 435)	This is class API.Subscription.
	Utilities (see page 441)	This is class API.Utilities.

Legend

	Class

Namespaces

Name	Description
Analytics (see page 442)	This is namespace API.Analytics.
Calendar (see page 447)	This is namespace API.Calendar.
Content (see page 449)	This is namespace API.Content.
Search (see page 514)	The Search Developer API enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 424), if you chose to install it, has excellent examples of programmatically using the SearchManager (see page 515). 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 (see page 449) that contains the word *cms*. • Retrieve content (see page 449) that contains *research* and has a searchable metadata *wellness* set to Cancer. • Retrieve images that contain *RotatingGraphic1* in their title. • Retrieve geo-content (see page 449) from CMS400.Net.
Task (see page 519)	This is namespace API.Task.
User (see page 534)	This is namespace API.User.

Classes

The following table lists %CATEGORYL% in this documentation.

Classes

	Name	Description
	BusinessRule (see page 365)	Business Rules allow you to define conditions for Web pages on your site. You choose an action based on whether the conditions are true or false when a site visitor browses your site. For example: • **Condition** - a site visitor visits your site for the first time • **Action** - add a cookie to the site visitor's system and pop up a welcome message With Business Rules, you do not write code to set conditions and initiate actions. You simply complete the information requested in the Business Rules wizard within the workarea, have your developer add a Business Rules server control... more (see page 365)
	CustomFields (see page 367)	This is class API.CustomFields.
	DateTimeSelector (see page 377)	This is class API.DateTimeSelector.
	Folder (see page 383)	This is class API.Folder.
	Font (see page 392)	This is class API.Font.
	Library (see page 395)	This is class API.Library.
	Localization (see page 403)	This is class API.Localization.
	Metadata (see page 412)	This is class API.Metadata.
	Permissions (see page 419)	This is class API.Permissions.
	Site (see page 424)	This is class API.Site.

	SiteMap (see page 430)	This is class API.SiteMap.
	Subscription (see page 435)	This is class API.Subscription.
	Utilities (see page 441)	This is class API.Utilities.

Legend

	Class

BusinessRule Class

Class Hierarchy

C#

```
public class BusinessRule : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class BusinessRule
Inherits Ektron.Cms.API.Common
```

File

BusinessRule.vb

Description

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 363) allows you to work with results of a Business Rule. The BusinessRules (see page 61) server control API (see page 363) allows you to customize how and where a Business Rule is used on a Web page.

To learn about the BusinessRules (see page 61) server control API (see page 363), click here (see page 61).

BusinessRule Methods

The methods of the BusinessRule class are listed here.

Public Methods

	Name	Description
	GetBusinessRules (see page 366)	Method to get the results from a business rule. This method is overloaded.

Legend

	Method

GetBusinessRules Method

Method to get the results from a business rule. This method is overloaded.

Overload List

	Name	Description
	BusinessRule.GetBusinessRules (String, String, String, String, String, String, String, Integer) (see page 366)	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 366)	Method to get the results from a business rule by identifier. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control.

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

Method to get the results from a business rule by ID. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset.

C#

```
public String GetBusinessRules(String httpHost, String urlRest, String cookies, String
formValues, String queryValues, String serverValues, String paramValues, int iRuleID);
```

Visual Basic

```
Public Function GetBusinessRules(ByVal httpHost As String, ByVal urlRest As String, ByVal
cookies As String, ByVal formValues As String, ByVal queryValues As String, ByVal serverValues
As String, ByVal paramValues As String, ByVal iRuleID As Integer) As String
```

Parameters

Parameters	Description
`httpHost`	The HTTP host the request is made on.
`urlRest`	not used
`cookies`	Cookie Value Collection
`formValues`	Form (see page 493) Value Collection
`queryValues`	Querystring Value Collection
`serverValues`	Server Variables Collection
`paramValues`	Custom Value Collection
`iRuleID`	Integer representing the ID of the Ruleset

Returns

A string representing the output of the evaluated ruleset.

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

Method to get the results from a business rule by identifier. Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc. Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control.

C#

```
public String GetBusinessRules(String httpHost, String urlRest, String cookies, String
formValues, String queryValues, String serverValues, String paramValues, String sRuleID);
```

Visual Basic

```
Public Function GetBusinessRules(ByVal httpHost As String, ByVal urlRest As String, ByVal
```

```
cookies As String, ByVal formValues As String, ByVal queryValues As String, ByVal serverValues
As String, ByVal paramValues As String, ByVal sRuleID As String) As String
```

Parameters

Parameters	Description
`httpHost`	The HTTP host the request is made on.
`urlRest`	not used
`cookies`	Cookie Value Collection
`formValues`	Form (see page 493) Value Collection
`queryValues`	Querystring Value Collection
`serverValues`	Server Variables Collection
`paramValues`	Custom Value Collection
`sRuleID`	String representing the identifier of the Ruleset. The identifier is a combination of the path/page/uniqueID of the business rules control.

Returns

A string representing the output of the evaluated ruleset.

CustomFields Class

Class Hierarchy

C#

```
public class CustomFields : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class CustomFields
Inherits Ektron.Cms.API.Common
```

File

CustomFields.vb

Description

This is class API.CustomFields.

CustomFields Methods

The methods of the CustomFields class are listed here.

Public Methods

	Name	Description
	AddNewEditableCustomFieldAssignments (see page 369)	Returns the HTML string for the add/edit custom fields.
S	BoolToYesNo (see page 369)	Returns labels for yes or no based on locale.

	GetEditableCustomFieldAssignments (see page 369)	Returns the editable custom fields. Break Inheritance checkbox: • Visible if not at root. • Checked if any fields assigned without inheritance. • If unchecked & form saved, removes all assignments. • If checked & form saved and no assignments chosen, creates NULL assignment (CustomFieldID -1). • When checked, enables all assignment checkboxes (iterate through all of this tables elements). • When unchecked, disables all assignment checkboxes (iterate through all of this tables elements). • Assignment Checkboxes: • When clicked: if 'Assignment' button is going unchecked, then ensure required is unchecked. • When clicked: if 'Required' button, don't accept checks if assigned is unchecked. • When form submitted, copy checked... more (see page 369)
S	GetEnhancedMetadataArea (see page 370)	Returns the HTML needed to render the EnhancedMetadata selection area.
S	GetEnhancedMetadataScript (see page 370)	Returns support Javascript code for enhanced metadata.
	GetFieldsByFolder (see page 370)	Returns all of the custom attributes by folder.
S	GetSearchProperties (see page 371)	Returns the search's properties. This method is overloaded.
S	Populate_AssetRequestObjectFromForm (see page 372)	Returns asset request object using form data.
	ProcessCustomFields (see page 373)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form (see page 493) processor: • If break inheritance checked, and any items in hidden field, send contents to db. • If break inheritance checked, and hidden field empty, create dummy assignment fieldID = -1. • If break inheritance NOT checked, remove all assignments for this folder.
S	SearchAllAssets (see page 373)	Returns the search (see page 514) result based on assets.
S	WriteFilteredMetadataForEdit (see page 373)	Returns an HTML string for filtered metadata.
S	WriteFilteredMetadataForView (see page 374)	Returns an HTML string for metadata view.
S	WriteLibrarySearchExtended (see page 374)	Returns an HTML string for the library search (see page 514) custom metadata properties.
S	WriteMetadataDefaultForEdit (see page 374)	Returns an HTML string for the default metadata.
S	WriteMetadataTypeForView (see page 375)	Returns a metadata type for view.
S	WritePublicSearch (see page 375)	Returns WritepublicsearchEx (see page 376).
S	WritePublicSearchEx (see page 376)	Returns an HTML string for extended the search (see page 514) attributes.

Legend

	Method
S	static

CustomFields.AddNewEditableCustomFieldAssignments Method

Returns the HTML string for the add/edit custom fields.

C#

```
public String AddNewEditableCustomFieldAssignments(int parentfolderid);
```

Visual Basic

```
Public Function AddNewEditableCustomFieldAssignments(ByVal parentfolderid As Integer) As String
```

Parameters

Parameters	Description
parentfolderid	The parent folder's ID.

Returns

String

CustomFields.BoolToYesNo Method

Returns labels for yes or no based on locale.

C#

```
public Shared String BoolToYesNo(Boolean bValue);
```

Visual Basic

```
Public Shared Function BoolToYesNo(ByVal bValue As Boolean) As String
```

Parameters

Parameters	Description
bValue	True or False

Returns

String

CustomFields.GetEditableCustomFieldAssignments Method

Returns the editable custom fields.

Break Inheritance checkbox:

- Visible if not at root.
- Checked if any fields assigned without inheritance.
- If unchecked & form saved, removes all assignments.
- If checked & form saved and no assignments chosen, creates NULL assignment (CustomFieldID -1).
- When checked, enables all assignment checkboxes (iterate through all of this tables elements).
- When unchecked, disables all assignment checkboxes (iterate through all of this tables elements).
- Assignment Checkboxes:
- When clicked: if 'Assignment' button is going unchecked, then ensure required is unchecked.
- When clicked: if 'Required' button, don't accept checks if assigned is unchecked.
- When form submitted, copy checked items to hidden field.
- Form (see page 493) 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 (see page 534).

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, Long LanguageID);
```

Visual Basic

```
Public Function GetFieldsByFolder(ByVal FolderID As Long, ByVal LanguageID As Long) As
Collection
```

Parameters

Parameters	Description
`FolderID`	The folder's ID.
`LanguageID`	The language's ID.

Returns

Microsoft.VisualBasic.Collection

GetSearchProperties Method

Returns the search's properties. This method is overloaded.

Overload List

	Name	Description
	CustomFields.GetSearchProperties (Ektron.Cms.Common.SearchAssetDispayRequest) (see page 371)	Returns an HTML string for the search (see page 514) properties display.
	CustomFields.GetSearchProperties (String, String, String, String, Collection, String, String, String) (see page 371)	Returns an HTML string to display search (see page 514) properties.
	CustomFields.GetSearchProperties (System.Collections.Specialized.NameValueCollection, Ektron.Cms.Common.SearchItemCustom,), Ektron.Cms.Common.SearchItemCustom,)) (see page 372)	Returns the search (see page 514)'s properties.

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

Returns an HTML string for the search (see page 514) 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 (see page 514) 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 (see page 514) 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 (see page 514)'s properties.

C#

```
public Shared GetSearchProperties(System.Collections.Specialized.NameValueCollection FormReq, Ektron.Cms.Common.SearchItemCustom CustomProps, ), Ektron.Cms.Common.SearchItemCustom AssetProps, ));
```

Visual Basic

```
Public Shared Sub GetSearchProperties(ByVal FormReq As System.Collections.Specialized.NameValueCollection, ByRef CustomProps As Ektron.Cms.Common.SearchItemCustom,  As ), ByRef AssetProps As Ektron.Cms.Common.SearchItemCustom,  As ))
```

Parameters

Parameters	Description
FormReq	Form (see page 493) field values.
CustomProps	Custom properties value.
AssetProps	Asset (see page 450) properties value.

CustomFields.Populate_AssetRequestObjectFromForm Method

Returns asset request object using form data.

C#

```
public Shared Ektron.Cms.Common.SearchAssetRequest Populate_AssetRequestObjectFromForm(System.Collections.Specialized.NameValueCollection PostBackData);
```

Visual Basic

```
Public Shared Function Populate_AssetRequestObjectFromForm(Optional ByVal PostBackData As System.Collections.Specialized.NameValueCollection = Nothing) As Ektron.Cms.Common.SearchAssetRequest
```

Returns

Ektron.Cms.Common.SearchAssetRequest

CustomFields.ProcessCustomFields Method

Process Folders' Custom-Field (and metadata) assignments:

When form submitted, copy checked items to hidden field.

Form (see page 493) 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 (see page 514) 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 514) parameters. (Required)

Returns

System.Text.StringBuilder

CustomFields.WriteFilteredMetadataForEdit Method

Returns an HTML string for filtered metadata.

C#

```
public Shared System.Text.StringBuilder
WriteFilteredMetadataForEdit(Ektron.Cms.ContentMetaData cMetadataTypes, Boolean
bSearchableOnly, String ty, Long FolderId, int Count, Hashtable metaPermsObj);
```

Visual Basic

```
Public Shared Function WriteFilteredMetadataForEdit(ByVal cMetadataTypes() As
```

```
Ektron.Cms.ContentMetaData, ByVal bSearchableOnly As Boolean, ByVal ty As String, ByVal
FolderId As Long, Optional ByRef Count As Integer = 0, Optional ByVal metaPermsObj As
Hashtable = Nothing) As System.Text.StringBuilder
```

Parameters

Parameters	Description
cMetadataTypes	Metadata (see page 412) 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 412) types.
FolderId	The folder's Id.
bSearchableOnly	True or False

Returns

String

CustomFields.WriteLibrarySearchExtended Method

Returns an HTML string for the library search (see page 514) custom metadata properties.

C#

```
public Shared System.Text.StringBuilder WriteLibrarySearchExtended(int folderId);
```

Visual Basic

```
Public Shared Function WriteLibrarySearchExtended(ByVal folderId As Integer) As
System.Text.StringBuilder
```

Parameters

Parameters	Description
folderId	The folder's ID.

Returns

System.Text.StringBuilder

CustomFields.WriteMetadataDefaultForEdit Method

Returns an HTML string for the default metadata.

C#

```
public Shared System.Text.StringBuilder WriteMetadataDefaultForEdit(Ektron.Cms.ContentMetaData
cMetadataType, int DataType, Hashtable metaPermsObj);
```

Visual Basic

```
Public Shared Function WriteMetadataDefaultForEdit(ByVal cMetadataType As
Ektron.Cms.ContentMetaData, ByVal DataType As Integer, Optional ByVal metaPermsObj As
Hashtable = Nothing) As System.Text.StringBuilder
```

Parameters

Parameters	Description
cMetadataType	Metadata (see page 412) values.
DataType	Data types

Returns

System.Text.StringBuilder

CustomFields.WriteMetadataTypeForView Method

Returns a metadata type for view.

C#

```
public Shared String WriteMetadataTypeForView(String DataType, String Caption, String Value);
```

Visual Basic

```
Public Shared Function WriteMetadataTypeForView(ByVal DataType As String, ByVal Caption As
String, ByVal Value As String) As String
```

Parameters

Parameters	Description
DataType	Data types.
Caption	Caption value.
Value	Value. (Required)

Returns

String

CustomFields.WritePublicSearch Method

Returns WritepublicsearchEx (see page 376).

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 514) asset data. (Required)

Returns

System.Text.StringBuilder

CustomFields.WritePublicSearchEx Method

Returns an HTML string for extended the search (see page 514) attributes.

C#

```
public Shared System.Text.StringBuilder WritePublicSearchEx(SearchAssetDispayRequest sadReq,
Boolean EnableAdvancedLink, Boolean EnableContentBtn, Boolean EnableContent, Boolean
EnableFormsBtn, Boolean EnableForms, Boolean EnableDMSAssetsBtn, Boolean EnableDMSAssets,
Boolean EnableLibImagesBtn, Boolean EnableLibImages, Boolean EnableLibFilesBtn, Boolean
EnableLibFiles, Boolean EnableLibHyperlinksBtn, Boolean EnableLibHyperlinks, Boolean
EnableArchivedBtn, Boolean EnableArchived, String TargetFolderDescriptionList, Boolean
EnableContentFieldset, String ContentFieldsetLegend, Boolean EnableLibraryFieldset, String
LibraryFieldsetLegend, Boolean EnableFolderFieldset, String FolderFieldsetLegend, Boolean
EnableCallBack, Boolean RemoveTeaserHtml, int MaxTeaserLength);
```

Visual Basic

```
Public Shared Function WritePublicSearchEx(ByVal sadReq As SearchAssetDispayRequest, ByVal
EnableAdvancedLink As Boolean, ByVal EnableContentBtn As Boolean, ByVal EnableContent As
Boolean, ByVal EnableFormsBtn As Boolean, ByVal EnableForms As Boolean, ByVal
EnableDMSAssetsBtn As Boolean, ByVal EnableDMSAssets As Boolean, ByVal EnableLibImagesBtn As
Boolean, ByVal EnableLibImages As Boolean, ByVal EnableLibFilesBtn As Boolean, ByVal
EnableLibFiles As Boolean, ByVal EnableLibHyperlinksBtn As Boolean, ByVal EnableLibHyperlinks
As Boolean, ByVal EnableArchivedBtn As Boolean, ByVal EnableArchived As Boolean, ByVal
TargetFolderDescriptionList As String, Optional ByVal EnableContentFieldset As Boolean =
False, Optional ByVal ContentFieldsetLegend As String = "", Optional ByVal
EnableLibraryFieldset As Boolean = False, Optional ByVal LibraryFieldsetLegend As String = "",
Optional ByVal EnableFolderFieldset As Boolean = False, Optional ByVal FolderFieldsetLegend As
String = "", Optional ByVal EnableCallBack As Boolean = False, Optional ByVal RemoveTeaserHtml
As Boolean = True, Optional ByVal MaxTeaserLength As Integer = 100) As
System.Text.StringBuilder
```

Parameters

Parameters	Description
sadReq	Search (see page 514) asset properties. (Required)
EnableAdvancedLink	True or False
EnableContentBtn	True or False
EnableContent	True or False
EnableFormsBtn	True or False
EnableForms	True or False
EnableDMSAssetsBtn	True or False
EnableDMSAssets	True or False
EnableLibImagesBtn	True or False
EnableLibImages	True or False
EnableLibFilesBtn	True or False
EnableLibFiles	True or False
EnableLibHyperlinksBtn	True or False
EnableLibHyperlinks	True or False
EnableArchivedBtn	True or False
EnableArchived	True or False
TargetFolderDescriptionList	True or False
EnableContentFieldset	True or False
ContentFieldsetLegend	True or False
EnableLibraryFieldset	True or False
LibraryFieldsetLegend	True or False
EnableFolderFieldset	True or False
FolderFieldsetLegend	True or False

Returns

System.Text.StringBuilder

DateTimeSelector Class

Class Hierarchy

C#

```
public class DateTimeSelector : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class DateTimeSelector
Inherits Ektron.Cms.API.Common
```

File

DateTimeSelector.vb

Description

This is class API.DateTimeSelector.

DateTimeSelector Methods

The methods of the DateTimeSelector class are listed here.

Public Methods

	Name	Description
	DisplayCultureDate (see page 377)	Returns server culture date using culture object.
	DisplayCultureDateTime (see page 378)	Returns server culture date and time.
	DisplayCultureTime (see page 378)	Returns the server culture time.
	RoundMinutes (see page 378)	Math function to round the minute.

Legend

	Method

DateTimeSelector.DisplayCultureDate Method

Returns server culture date using culture object.

C#

```
public String DisplayCultureDate(Boolean isEditable, String inSpanID, String inFormElemName);
```

Visual Basic

```
Public Function DisplayCultureDate(ByVal isEditable As Boolean, Optional ByVal inSpanID As
String = "", Optional ByVal inFormElemName As String = "") As String
```

Parameters

Parameters	Description
isEditable	True for editable span tag, otherwise false.
inSpanID	ID value for the span tag.
inFormElemName	Name value for the span tag.

Returns

String

DateTimeSelector.DisplayCultureDateTime Method

Returns server culture date and time.

C#

```
public String DisplayCultureDateTime(Boolean isEditable, String inSpanID, String
inFormElemName);
```

Visual Basic

```
Public Function DisplayCultureDateTime(ByVal isEditable As Boolean, Optional ByVal inSpanID As
String = "", Optional ByVal inFormElemName As String = "") As String
```

Parameters

Parameters	Description
isEditable	True for editable span tag, otherwise false.
inSpanID	Id value for the span tag.
inFormElemName	Name value for the span tag.

Returns

String

DateTimeSelector.DisplayCultureTime Method

Returns the server culture time.

C#

```
public String DisplayCultureTime(Boolean isEditable, String inSpanID, String inFormElemName);
```

Visual Basic

```
Public Function DisplayCultureTime(ByVal isEditable As Boolean, Optional ByVal inSpanID As
String = "", Optional ByVal inFormElemName As String = "") As String
```

Parameters

Parameters	Description
isEditable	True for editable span tag, otherwise false.
inSpanID	ID value for the span tag.
inFormElemName	Name value for the span tag.

DateTimeSelector.RoundMinutes Method

Math function to round the minute.

C#

```
public Date RoundMinutes(Date dateToRound, int minuteIncrement);
```

Visual Basic

```
Public Function RoundMinutes(ByVal dateToRound As Date, ByVal minuteIncrement As Integer) As
Date
```

Parameters

Parameters	Description
dateToRound	Date value
minuteIncrement	Minute value

Returns

Date

DateTimeSelector Properties

The properties of the DateTimeSelector class are listed here.

Public Properties

	Name	Description
R	ClientDateSeparator (see page 379)	Returns client date separator value.
R	ClientLongDatePattern (see page 379)	Returns the client date in long date format.
R	ClientLongTimePattern (see page 380)	Returns the clients time in long format.
R	ClientShortDatePattern (see page 380)	Returns the clients date in short format.
R	ClientShortTimePattern (see page 380)	Returns the client time in short format.
R	ClientTimeSeparator (see page 380)	Returns the clients time separator.
	ExtendedMeta (see page 381)	Returns the value of meta extended or not meta extended.
	FormElement (see page 381)	Sets or gets the form element value.
	FormName (see page 381)	Sets or gets the form name.
	RequestInfo (see page 381)	Gets or sets the RequestInfo object.
R	ServerDateSeparator (see page 382)	Returns the server side date separator.
R	ServerShortDatePattern (see page 382)	Returns the server side short date pattern.
R	ServerShortTimePattern (see page 382)	Returns the server short time pattern.
R	ServerTimeSeparator (see page 382)	Returns the server side date separator.
	SpanAttributes (see page 383)	Gets or sets the span attributes.
	SpanId (see page 383)	Gets or sets span for calendar (see page 447) control.
	TargetDate (see page 383)	Sets or gets the target date.
	ValidationType (see page 383)	Gets or sets the validation type.

Legend

	Property
R	read only

DateTimeSelector.ClientDateSeparator Property

Returns client date separator value.

C#

```
public ReadOnly String ClientDateSeparator;
```

Visual Basic

```
Public ReadOnly Property ClientDateSeparator() As String
```

Returns

String

DateTimeSelector.ClientLongDatePattern Property

Returns the client date in long date format.

C#

```
public ReadOnly String ClientLongDatePattern;
```

Visual Basic

```
Public ReadOnly Property ClientLongDatePattern() As String
```

Returns

String

DateTimeSelector.ClientLongTimePattern Property

Returns the clients time in long format.

C#

```
public ReadOnly String ClientLongTimePattern;
```

Visual Basic

```
Public ReadOnly Property ClientLongTimePattern() As String
```

Returns

String

DateTimeSelector.ClientShortDatePattern Property

Returns the clients date in short format.

C#

```
public ReadOnly String ClientShortDatePattern;
```

Visual Basic

```
Public ReadOnly Property ClientShortDatePattern() As String
```

Returns

String

DateTimeSelector.ClientShortTimePattern Property

Returns the client time in short format.

C#

```
public ReadOnly String ClientShortTimePattern;
```

Visual Basic

```
Public ReadOnly Property ClientShortTimePattern() As String
```

Returns

String

DateTimeSelector.ClientTimeSeparator Property

Returns the clients time separator.

C#

```
public ReadOnly String ClientTimeSeparator;
```

Visual Basic

```
Public ReadOnly Property ClientTimeSeparator() As String
```

Returns

String

DateTimeSelector.ExtendedMeta Property

Returns the value of meta extended or not meta extended.

C#

```
public Boolean ExtendedMeta;
```

Visual Basic

```
Public Property ExtendedMeta() As Boolean
```

Returns

Boolean

DateTimeSelector.FormElement Property

Sets or gets the form element value.

C#

```
public String FormElement;
```

Visual Basic

```
Public Property FormElement() As String
```

Returns

String

DateTimeSelector.FormName Property

Sets or gets the form name.

C#

```
public String FormName;
```

Visual Basic

```
Public Property FormName() As String
```

Returns

String

DateTimeSelector.RequestInfo Property

Gets or sets the RequestInfo object.

C#

```
public EkRequestInformation RequestInfo;
```

Visual Basic

```
Public Property RequestInfo() As EkRequestInformation
```

Returns

Ektron.Cms.Common.EkRequestInformation

DateTimeSelector.ServerDateSeparator Property

Returns the server side date separator.

C#

```
public ReadOnly String ServerDateSeparator;
```

Visual Basic

```
Public ReadOnly Property ServerDateSeparator() As String
```

Returns

String

DateTimeSelector.ServerShortDatePattern Property

Returns the server side short date pattern.

C#

```
public ReadOnly String ServerShortDatePattern;
```

Visual Basic

```
Public ReadOnly Property ServerShortDatePattern() As String
```

Returns

String

DateTimeSelector.ServerShortTimePattern Property

Returns the server short time pattern.

C#

```
public ReadOnly String ServerShortTimePattern;
```

Visual Basic

```
Public ReadOnly Property ServerShortTimePattern() As String
```

Returns

String

DateTimeSelector.ServerTimeSeparator Property

Returns the server side date separator.

C#

```
public ReadOnly String ServerTimeSeparator;
```

Visual Basic

```
Public ReadOnly Property ServerTimeSeparator() As String
```

Returns

String

DateTimeSelector.SpanAttributes Property

Gets or sets the span attributes.

C#

```
public String SpanAttributes;
```

Visual Basic

```
Public Property SpanAttributes() As String
```

Returns

String

DateTimeSelector.SpanId Property

Gets or sets span for calendar (see page 447) control.

C#

```
public String SpanId;
```

Visual Basic

```
Public Property SpanId() As String
```

Returns

String

DateTimeSelector.TargetDate Property

Sets or gets the target date.

C#

```
public Date TargetDate;
```

Visual Basic

```
Public Property TargetDate() As Date
```

Returns

Date

DateTimeSelector.ValidationType Property

Gets or sets the validation type.

C#

```
public int ValidationType;
```

Visual Basic

```
Public Property ValidationType() As Integer
```

Returns

Integer

Folder Class

Class Hierarchy

C#

```
public class Folder : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Folder
Inherits Ektron.Cms.API.Common
```

File

Folder.vb

Description

This is class API.Folder.

Folder Methods

The methods of the Folder class are listed here.

Public Methods

	Name	Description
	AddCommunityFolder (see page 384)	Creates CMS Community folder (see page 383).
	AddFolder (see page 385)	Creates a new folder (see page 383) in the CMS. By default the new folder (see page 383) inherits subscription properties of parent folder (see page 383). To override, use the BreakInheritance property of the FolderRequest.SubscriptionProperties (see page 778) object (set it to True). *See Also:* The Example section.
V	CreateFolder (see page 386)	Creates a folder (see page 383).
	DeleteFolderById (see page 386)	Delete the folder (see page 383) by its ID. *See Also:* The Example section.
	GetChildFolders (see page 387)	Loads all the subfolders for a given folder (see page 383) ID. *See Also:* The Example section.
	GetFolder (see page 389)	
	GetFolderDataWithPermission (see page 390)	Retrieves the folder (see page 383)'s details with permission information.
	GetFolderId (see page 390)	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.
V	GetFolderInfo (see page 391)	Returns the folder (see page 383) detail.
	GetPath (see page 391)	Gets the path of a folder (see page 383) in the Workarea by using the folder (see page 383)'s ID.
	RenameFolder (see page 392)	Rename the folder (see page 383) with new name.
	UpdateFolder (see page 392)	Updates the CMS folder (see page 383)'s information.

Legend

	Method
V	virtual

Folder.AddCommunityFolder Method

Creates CMS Community folder (see page 383).

C#

```
public AddCommunityFolder(FolderRequest request);
```

Visual Basic

```
Public Sub AddCommunityFolder(ByVal request As FolderRequest)
```

Parameters

Parameters	Description
`request`	The Folder (see page 383)'s properties as a FolderRequest (see page 770) object.

Folder.AddFolder Method

Creates a new folder (see page 383) in the CMS. By default the new folder (see page 383) inherits subscription properties of parent folder (see page 383). To override, use the BreakInheritance property of the FolderRequest.SubscriptionProperties (see page 778) 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 383)'s properties as a FolderRequest (see page 770) object.

Example

The following example shows how to create a Web page from which a folder (see page 383) 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 (see page 383). 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 383) API (see page 363).

```
            Dim folderApi As New Ektron.Cms.API.Folder
```

3. Create a Try/Catch around the AddFolder method. This allows you to catch any exceptions and helps when debugging.
4. Create a FolderRequest (see page 770) 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 (see page 383) 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 (see page 534) know what happened.

```
            Catch ex As Exception
                lblResult.Text = ex.Message
            End Try

    End Sub
```

Folder.CreateFolder Method

Creates a folder (see page 383).

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 (see page 383) by its ID.

See Also: The Example section.

C#

```
public Boolean DeleteFolderById(int Id);
```

Visual Basic

```
Public Function DeleteFolderById(ByVal Id As Integer) As Boolean
```

Parameters

Parameters	Description
Id	The folder (see page 383)'s ID.

Returns

False = successfully deleted. Otherwise, returns True.

Example

The following example shows how to create a Web page from which a folder (see page 383) 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 (see page 383). 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 (see page 383) 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 383) API (see page 363).
```
            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 (see page 534) know.
```
              Catch ex As Exception
                  lblResult.Text = "Error deleting folder"
              End Try

      End Sub
```

Folder.GetChildFolders Method

Loads all the subfolders for a given folder (see page 383) ID.

See Also: The Example section.

C#

```
public FolderData[] GetChildFolders(int Id, Boolean Recursive, String OrderBy);
```

Visual Basic

```
Public Function GetChildFolders(ByVal Id As Integer, Optional ByVal Recursive As Boolean =
False, Optional ByVal OrderBy As String = "name") As FolderData()
```

Parameters

Parameters	Description
`Id`	The folder (see page 383)'s ID
`Recursive`	True = recursive look up. False = none.
`OrderBy`	An order by clause

Returns

Ektron.Cms.FolderData (see page 755)()

Example

The following example shows how to create a Web page from which a folder (see page 383) 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 (see page 383) 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 383) API (see page 363).
```
Dim folderApi As New Ektron.Cms.API.Folder
```
3. Check to make sure the folder (see page 383) 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 (see page 755).
```
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 (see page 383)'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

Overload List

	Name	Description
	Folder.GetFolder (Integer) (see page 389)	Retrieves the folder (see page 383)'s details.
	Folder.GetFolder (Integer, Boolean) (see page 390)	Retrieves the folder (see page 383)'s details with taxonomy information.

Folder.GetFolder Method (Integer)

Retrieves the folder (see page 383)'s details.

C#

```
public FolderData GetFolder(int Id);
```

Visual Basic

```
Public Function GetFolder(ByVal Id As Integer) As FolderData
```

Parameters

Parameters	Description
Id	The Folder (see page 383)'s ID.

Returns

Ektron.Cms.FolderData (see page 755)

Remarks

Read permission required

Folder.GetFolder Method (Integer, Boolean)

Retrieves the folder (see page 383)'s details with taxonomy information.

C#

```
public FolderData GetFolder(int Id, Boolean GetTaxonomy);
```

Visual Basic

```
Public Function GetFolder(ByVal Id As Integer, ByVal GetTaxonomy As Boolean) As FolderData
```

Parameters

Parameters	Description
Id	The Folder (see page 383)'s ID.
GetTaxonomy	If true, gets the Taxonomy (see page 498) associated with the folder (see page 383) information.

Returns

Ektron.Cms.FolderData (see page 755)

Folder.GetFolderDataWithPermission Method

Retrieves the folder (see page 383)'s details with permission information.

C#

```
public FolderData GetFolderDataWithPermission(int Id);
```

Visual Basic

```
Public Function GetFolderDataWithPermission(ByVal Id As Integer) As FolderData
```

Parameters

Parameters	Description
Id	The folder (see page 383)'s integer ID

Returns

Ektron.Cms.FolderData (see page 755)

GetFolderId Method

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 (Integer) (see page 390)	Finds the folder (see page 383) ID by using a content (see page 449)'s ID.
	Folder.GetFolderId (String, Integer) (see page 391)	Finds the folder (see page 383) ID by specifing its folder (see page 383) name and the parent folder (see page 383) ID.

Folder.GetFolderId Method (Integer)

Finds the folder (see page 383) ID by using a content (see page 449)'s ID.

C#

```
public int GetFolderId(int Id);
```

Visual Basic

```
Public Function GetFolderId(ByVal Id As Integer) As Integer
```

Parameters

Parameters	Description
Id	The content (see page 449)'s integer ID.

Returns

Integer representing the folder (see page 383) ID.

Folder.GetFolderId Method (String, Integer)

Finds the folder (see page 383) ID by specifing its folder (see page 383) name and the parent folder (see page 383) ID.

C#

```
public int GetFolderId(String FolderName, int ParentId);
```

Visual Basic

```
Public Function GetFolderId(ByVal FolderName As String, ByVal ParentId As Integer) As Integer
```

Parameters

Parameters	Description
FolderName	Name of the folder (see page 383).
ParentId	The integer ID of the parent folder (see page 383).

Returns

Returns the folder (see page 383) ID. Returns 0 if it does not exist.

Folder.GetFolderInfo Method

Returns the folder (see page 383) 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 (see page 383)'s path.

Returns

String()

Folder.GetPath Method

Gets the path of a folder (see page 383) in the Workarea by using the folder (see page 383)'s ID.

C#

```
public String GetPath(int Id);
```

Visual Basic

```
Public Function GetPath(ByVal Id As Integer) As String
```

Parameters

Parameters	Description
Id	The folder (see page 383)'s ID.

Returns

Returns the folder (see page 383) path as a string. For example: `\~SiteRoot~\Products\Medication Management`.

Folder.RenameFolder Method

Rename the folder (see page 383) with new name.

C#

```
public Boolean RenameFolder(int cId, String title);
```

Visual Basic

```
Public Function RenameFolder(ByVal cId As Integer, ByVal title As String) As Boolean
```

Parameters

Parameters	Description
cId	The folder (see page 383)'s ID
title	The new folder (see page 383)'s name

Returns

False = successfully renamed. Otherwise, True.

Folder.UpdateFolder Method

Updates the CMS folder (see page 383)'s information.

C#

```
public UpdateFolder(FolderRequest request);
```

Visual Basic

```
Public Sub UpdateFolder(ByVal request As FolderRequest)
```

Parameters

Parameters	Description
request	The Folder (see page 383)'s properties as a FolderRequest (see page 770) object.

Font Class

Class Hierarchy

C#

```
public class Font : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Font
Inherits Ektron.Cms.API.Common
```

File

Font.vb

Description

This is class API.Font.

Font Methods

The methods of the Font class are listed here.

Public Methods

	Name	Description
	AddFont (see page 393)	Adds a new font (see page 392) to the system.
	DeleteFont (see page 393)	Deletes a font (see page 392) from the system.
	GetAllFonts (see page 393)	Loads all of the fonts as a array of FontData.
	GetFont (see page 394)	Loads the required font (see page 392) using the font (see page 392)'s ID.
	GetFontConfigList (see page 394)	Returns all of the font (see page 392) details in a drop down control.
	UpdateFont (see page 394)	Updates the font (see page 392).

Legend

	Method

Font.AddFont Method

Adds a new font (see page 392) to the system.

C#

```
public AddFont(Collection data);
```

Visual Basic

```
Public Sub AddFont(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Font (see page 392) data as a value key pair

Font.DeleteFont Method

Deletes a font (see page 392) from the system.

C#

```
public DeleteFont(Collection data);
```

Visual Basic

```
Public Sub DeleteFont(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Required font (see page 392) data (for example, font (see page 392) ID)

Font.GetAllFonts Method

Loads all of the fonts as a array of FontData.

C#

```
public FontData[] GetAllFonts();
```

Visual Basic

```
Public Function GetAllFonts() As FontData()
```

Returns

Ektron.Cms.FontData()

Font.GetFont Method

Loads the required font (see page 392) using the font (see page 392)'s ID.

C#

```
public FontData GetFont(int Id);
```

Visual Basic

```
Public Function GetFont(ByVal Id As Integer) As FontData
```

Parameters

Parameters	Description
Id	The font (see page 392)'s ID

Returns

Integer

Font.GetFontConfigList Method

Returns all of the font (see page 392) details in a drop down control.

C#

```
public String GetFontConfigList();
```

Visual Basic

```
Public Function GetFontConfigList() As String
```

Returns

String

Font.UpdateFont Method

Updates the font (see page 392).

C#

```
public UpdateFont(Collection data);
```

Visual Basic

```
Public Sub UpdateFont(ByVal data As Collection)
```

Parameters

Parameters	Description
data	The font (see page 392)'s data

Library Class

Class Hierarchy

C#

```
public class Library : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Library
Inherits Ektron.Cms.API.Common
```

File

Library.vb

Description

This is class API.Library.

Library Methods

The methods of the Library class are listed here.

Public Methods

	Name	Description
	AddLibraryItem (see page 396)	Adds a library (see page 395) item's information to the database. It does not actually upload the item to the library (see page 395).
	AddLoadBalanceItem (see page 396)	Adds a load balance path for files and images.
	DeleteLibraryItem (see page 396)	Deletes the library (see page 395) item by extension.
	DeleteLibraryItemById (see page 396)	Deletes the library (see page 395) items from the system.
	DeleteLoadBalanceItem (see page 397)	Deletes the load balance item from the system.
	GetAllChildLibItems (see page 397)	Loads all of the library (see page 395) items by type. Using the OrderBy parameter allows you to sort the library (see page 395) items. The currentPageNum, pageSize and totalPages parameters are used with paging. Setting currentPageNum and pageSize to zero causes the API (see page 363) To retrieve all items on a single page. **Note**: The caller of API (see page 363) must actually call the API (see page 363) with the correct page number to implement paging. The API (see page 363) does not provide any inherent UI for paging purposes.
	GetAllLoadBalancePaths (see page 399)	Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.
	GetLibraryItem (see page 400)	Loads the library item's data. This method is overloaded.
	GetLibrarySettings (see page 401)	Loads the library (see page 395) settings.
	GetLibraryTypes (see page 401)	Loads the library (see page 395) types.
	GetLoadBalancePath (see page 401)	Gets the load balance path by ID.
	InternalLibrarySearch (see page 402)	Searches library (see page 395) items.
	UpdateLibrary (see page 402)	Updates a library (see page 395) item.
	UpdateLibrarySettings (see page 402)	Update the library (see page 395) settings.
	UpdateLoadBalanceSettings (see page 403)	Updates the load balance settings.

	UpdateQlinkTemplates (see page 403)	Updates the Quicklink template.

Legend

	Method

Library.AddLibraryItem Method

Adds a library (see page 395) item's information to the database. It does not actually upload the item to the library (see page 395).

C#

```
public int AddLibraryItem(LibraryData libraryData);
```

Visual Basic

```
Public Function AddLibraryItem(ByRef libraryData As LibraryData) As Integer
```

Parameters

Parameters	Description
libraryData	Details of the new library (see page 395) 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 (see page 395) item by extension.

C#

```
public DeleteLibraryItem(Collection pagedata);
```

Visual Basic

```
Public Sub DeleteLibraryItem(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Library (see page 395) item data

Library.DeleteLibraryItemById Method

Deletes the library (see page 395) 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 395) item data

Library.DeleteLoadBalanceItem Method

Deletes the load balance item from the system.

C#

```
public DeleteLoadBalanceItem(int Id);
```

Visual Basic

```
Public Sub DeleteLoadBalanceItem(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	The load balance item ID

Library.GetAllChildLibItems Method

Loads all of the library (see page 395) items by type. Using the OrderBy parameter allows you to sort the library (see page 395) items. The currentPageNum, pageSize and totalPages parameters are used with paging. Setting currentPageNum and pageSize to zero causes the API (see page 363) To retrieve all items on a single page.

Note: The caller of API (see page 363) must actually call the API (see page 363) with the correct page number to implement paging. The API (see page 363) does not provide any inherent UI for paging purposes.

C#

```
public LibraryData[] GetAllChildLibItems(String Type, int ParentId, String OrderBy, int
currentPageNum, int pageSize, int totalPages);
```

Visual Basic

```
Public Function GetAllChildLibItems(ByVal Type As String, ByVal ParentId As Integer, ByVal
OrderBy As String, ByVal currentPageNum As Integer, ByVal pageSize As Integer, ByRef
totalPages As Integer) As LibraryData()
```

Parameters

Parameters	Description
Type	The library (see page 395) item's type. (Required) Use one of the following: • images • files • hyperlinks • quicklinks • forms
ParentId	The folder's ID.

`OrderBy`	The order to load the library (see page 395) items. You can pass: • librarytitle • libraryid • libraryfilename • date
`currentPageNum`	The current page number to display.
`pageSize`	The page size set by the caller of the API (see page 363).
`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 (see page 395) 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 (see page 395) API (see page 363).
```
            Dim libraryApi As New Ektron.Cms.API.Library
```
4. Create an array to contain the library (see page 395) 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 (see page 395) has images for the folder.

```
If (Not data Is Nothing) AndAlso (data.Length > 0) Then
```

7. If so, create an object for each library (see page 395) 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 (see page 395) 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

Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.

Overload List

	Name	Description
	Library.GetAllLoadBalancePaths (Integer, String) (see page 399)	Returns an array of LoadBalanceData.
	Library.GetAllLoadBalancePaths (String) (see page 400)	Returns all the load balance paths as a array of LoadBalanceData.

Library.GetAllLoadBalancePaths Method (Integer, String)

Returns an array of LoadBalanceData.

C#

```
public LoadBalanceData[] GetAllLoadBalancePaths(int Id, String Type);
```

Visual Basic

```
Public Function GetAllLoadBalancePaths(ByVal Id As Integer, ByVal Type As String) As
LoadBalanceData()
```

Parameters

Parameters	Description
Id	Load balance ID
Type	Use images or files

Returns

Ektron.Cms.LoadBalanceData()

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

Loads the library item's data. This method is overloaded.

Overload List

	Name	Description
	Library.GetLibraryItem (Integer) (see page 400)	Loads the library (see page 395) item data.
	Library.GetLibraryItem (Integer, Integer) (see page 400)	Loads the library (see page 395) item's data.

Library.GetLibraryItem Method (Integer)

Loads the library (see page 395) item data.

C#

```
public LibraryData GetLibraryItem(int LibID);
```

Visual Basic

```
Public Function GetLibraryItem(ByVal LibID As Integer) As LibraryData
```

Parameters

Parameters	Description
LibID	The library (see page 395) ID

Returns

Ektron.Cms.LibraryData

Library.GetLibraryItem Method (Integer, Integer)

Loads the library (see page 395) item's data.

C#

```
public LibraryData GetLibraryItem(int Id, int ParentId);
```

Visual Basic

```
Public Function GetLibraryItem(ByVal Id As Integer, ByVal ParentId As Integer) As LibraryData
```

Parameters

Parameters	Description
Id	The library (see page 395)'s ID
ParentId	The folder's ID

Returns

Ektron.Cms.LibraryData

Library.GetLibrarySettings Method

Loads the library (see page 395) settings.

C#

```
public LibraryConfigData GetLibrarySettings(int Id);
```

Visual Basic

```
Public Function GetLibrarySettings(Optional ByVal Id As Integer = -1) As LibraryConfigData
```

Parameters

Parameters	Description
Id	The folder's ID

Returns

Ektron.Cms.LibraryConfigData

Library.GetLibraryTypes Method

Loads the library (see page 395) types.

C#

```
public LibraryTypeData[] GetLibraryTypes(int Id, String OrderBy);
```

Visual Basic

```
Public Function GetLibraryTypes(ByVal Id As Integer, ByVal OrderBy As String) As
LibraryTypeData()
```

Parameters

Parameters	Description
Id	The folder's ID
OrderBy	Order by clause

Returns

Object

Library.GetLoadBalancePath Method

Gets the load balance path by ID.

C#

```
public LoadBalanceData GetLoadBalancePath(int Id);
```

Visual Basic

```
Public Function GetLoadBalancePath(ByVal Id As Integer) As LoadBalanceData
```

Parameters

Parameters	Description
Id	the load balance path's ID

Returns

Ektron.Cms.LoadBalanceData

Library.InternalLibrarySearch Method

Searches library (see page 395) items.

C#

```
public SearchResultData[] InternalLibrarySearch(Collection pagedata,
Ektron.Cms.Common.SearchAssetRequest sar);
```

Visual Basic

```
Public Function InternalLibrarySearch(ByVal pagedata As Collection, Optional ByVal sar As
Ektron.Cms.Common.SearchAssetRequest = Nothing) As SearchResultData()
```

Parameters

Parameters	Description
pagedata	Search (see page 514) criteria
sar	Search (see page 514) asset criteria

Returns

Ektron.Cms.SearchResultData()

Library.UpdateLibrary Method

Updates a library (see page 395) item.

C#

```
public UpdateLibrary(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateLibrary(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Library (see page 395) item data

Library.UpdateLibrarySettings Method

Update the library (see page 395) settings.

C#

```
public UpdateLibrarySettings(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateLibrarySettings(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Library (see page 395) 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 API.Localization.

Localization Methods

The methods of the Localization class are listed here.

Public Methods

	Name	Description
	GetAddViewLanguage (see page 404)	Returns an array of all the active content (see page 449) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content (see page 449) ID or whether it does not exist in that language (Type="ADD").
	GetAddViewLanguageForAllContent (see page 405)	Returns an array of all the active content (see page 449) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content (see page 449) block or whether no content (see page 449) exists in that language (Type="ADD").
	GetAddViewLanguageForMenus (see page 405)	Returns an array of all the active content (see page 449) 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 405)	Returns a list of content (see page 449) items (i.e., XLIFF skeletons) associated with a given job.
	GetFile (see page 406)	Returns the specified file record.
	GetFiles (see page 406)	Returns a list of files associated with a given job.
	GetFilesInZipFile (see page 406)	Returns a list of files in a given ZIP file.
	GetFlagUrl (see page 407)	Returns a URL to a small image of a national flag for the country and language specified by LanguageID or Data. This method is overloaded.
	GetInterchangeFile (see page 408)	Returns the interchange (i.e., XLIFF) file record specified by ID.
	GetInterchangeFiles (see page 408)	Returns a list of interchange files (i.e., XLIFF files) associated with a given content (see page 449) item (i.e., XLIFF Skeleton).
	GetInterchangeFilesInZipFile (see page 409)	Returns a list of interchange files (i.e., XLIFF files) contained within a given ZIP file.
	GetJobs (see page 409)	Gets a list of top level jobs or a list of the sub jobs of the given job. This method is overloaded.
	GetLocalizationUrl (see page 410)	Returns a URL to the base directory where localization (see page 403) files are stored. `For example, /~SiteRoot~/uploadedfiles/localization (see page 403).`
	GetTranslationUploadDirectory (see page 410)	Returns the server directory path to upload XLIFF and ZIP files for importing. For example, `C:\inetpub\wwwroot\~SiteRoot~\workarea\uploadedfiles\localization (see page 403).` See Also: StartImportTranslation (see page 412).
	StartExportContentForTranslation (see page 411)	Starts the process of exporting the content (see page 449) 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 412) method.
	StartExportFolderForTranslation (see page 411)	Starts the process of exporting all the content (see page 449) specified by `ContentType` in the specified folder. Optionally, all the content (see page 449) 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 412).
	StartExportMenusForTranslation (see page 412)	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 412) method.
	StartImportTranslation (see page 412)	Starts the process of importing XLIFF files that have been translated. The files should be uploaded to server directory specified by GetTranslationUploadDirectory (see page 410). Files may be XLIFF files (.xlf extension) or XLIFF files that are compressed in ZIP (.zip extension) files.

Legend

	Method

Localization.GetAddViewLanguage Method

Returns an array of all the active content (see page 449) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") for the specified content (see page 449) ID or whether it does not exist in that language (Type="ADD").

C#

```
public LanguageData[] GetAddViewLanguage(int Id);
```

Visual Basic

```
Public Function GetAddViewLanguage(ByVal Id As Integer) As LanguageData()
```

Parameters

Parameters	Description
`Id`	Content (see page 449) ID

Returns

Ektron.Cms.LanguageData()

Localization.GetAddViewLanguageForAllContent Method

Returns an array of all the active content (see page 449) languages where the LanguageData.Type indicates whether the language exists (Type="VIEW") in at least one content (see page 449) block or whether no content (see page 449) 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 (see page 449) 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 (see page 449) items (i.e., XLIFF skeletons) associated with a given job.

C#

```
public DS.LocalizationSkeletonDataTable GetContentItems(int JobID);
```

Visual Basic

```
Public Function GetContentItems(ByVal JobID As Integer) As DS.LocalizationSkeletonDataTable
```

Parameters

Parameters	Description
`JobID`	Specifies the job.

Returns

A LocalizationJobDataSet.LocalizationSkeletonDataTable of content (see page 449) (i.e., skeleton) items.

Localization.GetFile Method

Returns the specified file record.

C#

```
public DS.LocalizationJobFileRow GetFile(int FileID);
```

Visual Basic

```
Public Function GetFile(ByVal FileID As Integer) As DS.LocalizationJobFileRow
```

Parameters

Parameters	Description
FileID	Specifies a file (e.g., ZIP file)

Returns

A LocalizationJobDataSet.LocalizationJobFileRow" record of a job file or Nothing if not found.

Localization.GetFiles Method

Returns a list of files associated with a given job.

C#

```
public DS.LocalizationJobFileDataTable GetFiles(int JobID);
```

Visual Basic

```
Public Function GetFiles(ByVal JobID As Integer) As DS.LocalizationJobFileDataTable
```

Parameters

Parameters	Description
JobID	Specifies the job.

Returns

A LocalizationJobDataSet.LocalizationJobFileDataTable of files.

Remarks

Files are typically ZIP files for export jobs and ZIP and XLIFF files for import jobs.

Localization.GetFilesInZipFile Method

Returns a list of files in a given ZIP file.

C#

```
public DS.LocalizationJobFileDataTable GetFilesInZipFile(int FileID);
```

Visual Basic

```
Public Function GetFilesInZipFile(ByVal FileID As Integer) As DS.LocalizationJobFileDataTable
```

Parameters

Parameters	Description
FileID	Specifies a file (e.g., ZIP file)

Returns

A LocalizationJobDataSet.LocalizationJobFileDataTable of files.

Remarks

Files are typically ZIP and XLIFF files for import jobs. The export process usually will not generate ZIP files with other ZIP files in them.

GetFlagUrl Method

Returns a URL to a small image of a national flag for the country and language specified by LanguageID or Data. This method is overloaded.

Overload List

	Name	Description
	Localization.GetFlagUrl (Integer) (see page 407)	Returns a URL to a small image of a national flag for the country and language specified by LanguageID.
	Localization.GetFlagUrl (LanguageData) (see page 407)	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 408)	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(int ID);
```

Visual Basic

```
Public Function GetInterchangeFile(ByVal ID As Integer) As DS.LocalizationTargetRow
```

Parameters

Parameters	Description
ID	Specifies the interchange (i.e., XLIFF) file record.

Returns

A LocalizationJobDataSet.LocalizationTargetRow data record of the selected XLIFF file or Nothing if it does not exist.

Localization.GetInterchangeFiles Method

Returns a list of interchange files (i.e., XLIFF files) associated with a given content (see page 449) item (i.e., XLIFF Skeleton).

C#

```
public DS.LocalizationTargetDataTable GetInterchangeFiles(int ItemID);
```

Visual Basic

```
Public Function GetInterchangeFiles(ByVal ItemID As Integer) As DS.LocalizationTargetDataTable
```

Parameters

Parameters	Description
ItemID	Specifies the content (see page 449) 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 (see page 449) item it was generated from (the original language) and a new skeleton content (see page 449) item generated by merging the XLIFF with the original skeleton to produce a skeleton in the target language. A skeleton consists of the content (see page 449), typically the main HTML content (see page 449) block, a summary (HTML), title, meta data, etc.

Localization.GetInterchangeFilesInZipFile Method

Returns a list of interchange files (i.e., XLIFF files) contained within a given ZIP file.

C#

```
public DS.LocalizationTargetDataTable GetInterchangeFilesInZipFile(int FileID);
```

Visual Basic

```
Public Function GetInterchangeFilesInZipFile(ByVal FileID As Integer) As
DS.LocalizationTargetDataTable
```

Parameters

Parameters	Description
FileID	Specifies the ZIP file.

Returns

A LocalizationJobDataSet.LocalizationTargetDataTable of XLIFF files.

Remarks

XLIFF is a standard to interchange localization (see page 403) information for translation of text. The XLIFF files are compressed together into one or more ZIP files for convenience.

GetJobs Method

Gets a list of top level jobs or a list of the sub jobs of the given job. This method is overloaded.

Overload List

	Name	Description
	Localization.GetJobs () (see page 410)	Returns a list of the top-level jobs, export and import jobs. The jobs listed depend on the user (see page 534) id, unauthenticated users (e.g., translators) see only publicly available jobs (e.g., ZIP files for download). Administrators can view all jobs, non-administrators may only view jobs they created. This function is overloaded. See Also: GetJobs (Integer)
	Localization.GetJobs (Integer) (see page 410)	Returns a list of the sub jobs of the given job. This function is overloaded. See Also: GetJobs (see page 410)

Localization.GetJobs Method ()

Returns a list of the top-level jobs, export and import jobs. The jobs listed depend on the user (see page 534) id, unauthenticated users (e.g., translators) see only publicly available jobs (e.g., ZIP files for download). Administrators can view all jobs, non-administrators may only view jobs they created.

This function is overloaded. See Also: GetJobs (Integer)

C#

```
public DS.LocalizationJobDataTable GetJobs();
```

Visual Basic

```
Public Function GetJobs() As DS.LocalizationJobDataTable
```

Returns

A LocalizationJobDataSet.LocalizationJobDataTable of top-level jobs.

Localization.GetJobs Method (Integer)

Returns a list of the sub jobs of the given job.

This function is overloaded. See Also: GetJobs (see page 410)

C#

```
public DS.LocalizationJobDataTable GetJobs(int ParentJobID);
```

Visual Basic

```
Public Function GetJobs(ByVal ParentJobID As Integer) As DS.LocalizationJobDataTable
```

Parameters

Parameters	Description
`ParentJobID`	Specifies the job.

Returns

A LocalizationJobDataSet.LocalizationJobDataTable data table of jobs.

Remarks

Localization.GetLocalizationUrl Method

Returns a URL to the base directory where localization (see page 403) files are stored. For example, /~SiteRoot~/uploadedfiles/localization (see page 403).

C#

```
public String GetLocalizationUrl();
```

Visual Basic

```
Public Function GetLocalizationUrl() As String
```

Returns

A URL to the base localization (see page 403) 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 page 403).

See Also: StartImportTranslation (see page 412).

C#

```
public String GetTranslationUploadDirectory();
```

Visual Basic

```
Public Function GetTranslationUploadDirectory() As String
```

Returns

The server directory path (as a String) to store XLIFF and ZIP files that are to be imported.

Remarks

This directory path should not be displayed on a web page.

Localization.StartExportContentForTranslation Method

Starts the process of exporting the content (see page 449) 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 412) method.

C#

```
public StartExportContentForTranslation(String ListOfContentIDs, String ListOfLanguageIDs);
```

Visual Basic

```
Public Sub StartExportContentForTranslation(ByVal ListOfContentIDs As String, Optional ByVal
ListOfLanguageIDs As String = "")
```

Parameters

Parameters	Description
`ListOfContentIDs`	Comma-delimited list of content (see page 449) IDs. e.g., "`11,12`"
`ListOfLanguageIDs`	Comma-delimited list of language IDs. e.g., "`1031,1036`"

Remarks

To export all content (see page 449) in a folder, use StartExportFolderForTranslation (see page 411) (Integer Integer Boolean String). See http://www.xliff.org for more information about XLIFF.

Localization.StartExportFolderForTranslation Method

Starts the process of exporting all the content (see page 449) specified by `ContentType` in the specified folder. Optionally, all the content (see page 449) 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 412).

C#

```
public StartExportFolderForTranslation(int FolderID, int ContentType, Boolean
IncludeSubfolders, String ListOfLanguageIDs);
```

Visual Basic

```
Public Sub StartExportFolderForTranslation(ByVal FolderID As Integer, ByVal ContentType As
Integer, ByVal IncludeSubfolders As Boolean, Optional ByVal ListOfLanguageIDs As String = "")
```

Parameters

Parameters	Description
`FolderID`	Specifies the folder containing the content (see page 449) to export.

`ContentType`	Specifies which type or types of content (see page 449) to export.
`IncludeSubfolders`	Specifies whether to export just the specified folder or all subfolders as well.
`ListOfLanguageIDs`	Comma-delimited list of language IDs. e.g., "`1031,1036`"

Remarks

To export just selected content (see page 449), use StartExportFolderForTranslation (Integer 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 412) method.

C#

```
public StartExportMenusForTranslation(String ListOfLanguageIDs);
```

Visual Basic

```
Public Sub StartExportMenusForTranslation(Optional ByVal ListOfLanguageIDs As String = "")
```

Parameters

Parameters	Description
`ListOfLanguageIDs`	Comma-delimited list of language IDs. e.g., "`1031,1036`"

Remarks

See http://www.xliff.org for more information about XLIFF.

Localization.StartImportTranslation Method

Starts the process of importing XLIFF files that have been translated. The files should be uploaded to server directory specified by GetTranslationUploadDirectory (see page 410). Files may be XLIFF files (.xlf extension) or XLIFF files that are compressed in ZIP (.zip extension) files.

C#

```
public StartImportTranslation();
```

Visual Basic

```
Public Sub StartImportTranslation()
```

Remarks

See http://www.xliff.org for more information about XLIFF.

Metadata Class

Class Hierarchy

C#

```
public class Metadata : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Metadata
Inherits Ektron.Cms.API.Common
```

File

Metadata.vb

Description

This is class API.Metadata.

Metadata Methods

The methods of the Metadata class are listed here.

Public Methods

	Name	Description
	AddMetaDataType (see page 413)	
	DeleteMetadataType (see page 415)	Deletes a metadata (see page 412) type from the system.
	GetContentMetadataList (see page 416)	
	GetIntegratedSearchFolder (see page 418)	Loads all custom folder data.
	GetIntegratedSearchFolders (see page 418)	Loads all the custom folders.
	GetMetadataType (see page 418)	Loads all content (see page 449) metadata (see page 412).
	GetMetaDataTypes (see page 419)	Loads all the metadata (see page 412) types.
	UpdateIntegratedSearchFolder (see page 419)	Updates the metadata (see page 412) type.
	UpdateMetadataType (see page 419)	Updates the metadata (see page 412) type.

Legend

	Method

AddMetaDataType Method

Overload List

	Name	Description
	Metadata.AddMetaDataType (Collection) (see page 413)	Adds a new metadata (see page 412) type to the system.
	Metadata.AddMetaDataType (ContentMetaData) (see page 414)	Adds a new metadata (see page 412) type to the system.

Metadata.AddMetaDataType Method (Collection)

Adds a new metadata (see page 412) 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 412) items as value key pair.

Metadata.AddMetaDataType Method (ContentMetaData)

Adds a new metadata (see page 412) type to the system.

C#

```
public AddMetaDataType(ContentMetaData metaDataItem);
```

Visual Basic

```
Public Sub AddMetaDataType(ByRef metaDataItem As ContentMetaData)
```

Parameters

Parameters	Description
metaDataItem	ContentMetadata object containing metadata (see page 412) definition details.

Example

The following example shows how to create a Web page from which you can add a new metadata (see page 412) 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 (see page 412) 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 (see page 412) definition types. These are the only types available. Make sure you add the numeric value. This corresponds to metadata (see page 412) 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 (see page 412) 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 (see page 534) know what happened.
```
        Catch ex As Exception
           lblResult.Text = ex.Message
        End Try
    End Sub
```

Metadata.DeleteMetadataType Method

Deletes a metadata (see page 412) type from the system.

C#

```
public DeleteMetadataType(int Id);
```

Visual Basic

```
Public Sub DeleteMetadataType(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	Metatype ID

GetContentMetadataList Method

Overload List

	Name	Description
	Metadata.GetContentMetadataList (Integer) (see page 416)	Loads all of the metadata (see page 412) for the given content (see page 449).
	Metadata.GetContentMetadataList (Integer, Boolean) (see page 417)	Loads all of the metadata (see page 412) for the given content (see page 449).

Metadata.GetContentMetadataList Method (Integer)

Loads all of the metadata (see page 412) for the given content (see page 449).

C#

```
public CustomAttributeList GetContentMetadataList(int ContentId);
```

Visual Basic

```
Public Function GetContentMetadataList(ByVal ContentId As Integer) As CustomAttributeList
```

Parameters

Parameters	Description
ContentId	The contents ID.

Returns

Ektron.Cms.CustomAttributeList

Example

The following example shows how to create a Web page from which all metadata (see page 412) for a specific content (see page 449) 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 Integer) method. This method uses the InternalAdmin to retrieve the metadata (see page 412) for the content (see page 449) 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 412) API (see page 363), 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 (see page 449) 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 (see page 412) 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 (see page 412) 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 (Integer, Boolean)

Loads all of the metadata (see page 412) for the given content (see page 449).

C#

```
public CustomAttributeList GetContentMetadataList(int ContentId, Boolean preview);
```

Visual Basic

```
Public Function GetContentMetadataList(ByVal ContentId As Integer, ByVal preview As Boolean)
```

```
As CustomAttributeList
```

Parameters

Parameters	Description
ContentId	The contents ID.
preview	True implies get metadata (see page 412) from staging content (see page 449), else use published content (see page 449).

Returns

Ektron.Cms.CustomAttributeList

Metadata.GetIntegratedSearchFolder Method

Loads all custom folder data.

C#

```
public IntegratedFolderData GetIntegratedSearchFolder(int Id);
```

Visual Basic

```
Public Function GetIntegratedSearchFolder(ByVal Id As Integer) As IntegratedFolderData
```

Parameters

Parameters	Description
Id	The integrated search (see page 514) 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.GetMetadataType Method

Loads all content (see page 449) metadata (see page 412).

C#

```
public ContentMetaData GetMetadataType(int Id);
```

Visual Basic

```
Public Function GetMetadataType(ByVal Id As Integer) As ContentMetaData
```

Parameters

Parameters	Description
Id	The metadata (see page 412) type ID

Returns

Ektron.Cms.ContentMetaData

Metadata.GetMetaDataTypes Method

Loads all the metadata (see page 412) 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.UpdateIntegratedSearchFolder Method

Updates the metadata (see page 412) type.

C#

```
public UpdateIntegratedSearchFolder(IntegratedFolderData integratedFolder);
```

Visual Basic

```
Public Sub UpdateIntegratedSearchFolder(ByVal integratedFolder As IntegratedFolderData)
```

Parameters

Parameters	Description
pagedata	Metadata (see page 412) types

Metadata.UpdateMetadataType Method

Updates the metadata (see page 412) type.

C#

```
public UpdateMetadataType(Collection pagedata);
```

Visual Basic

```
Public Sub UpdateMetadataType(ByVal pagedata As Collection)
```

Parameters

Parameters	Description
pagedata	Metadata (see page 412) 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 API.Permissions.

Permissions Methods

The methods of the Permissions class are listed here.

Public Methods

	Name	Description
	AddItemPermission (see page 420)	Adds the item permissions (see page 419) for a user (see page 534) or a usergroup.
	DeleteItemPermission (see page 420)	Deletes a user (see page 534) or a group from the item's permissions (see page 419)
	DisableContentInheritance (see page 421)	Disable permissions (see page 419) inheritance on the content (see page 449) and copies current permissions (see page 419) to the content (see page 449).
	DisableFolderInheritance (see page 421)	Disable permissions (see page 419) inheritance on the folder and copies current permissions (see page 419) to the folder.
	EnableContentInheritance (see page 421)	Deletes current permissions (see page 419) from the content (see page 449) and enable permissions (see page 419) inheritance on the content (see page 449).
	EnableFolderInheritance (see page 422)	Deletes current permissions (see page 419) from the folder and enable permissions (see page 419) inheritance on the folder.
	GetUserPermissions (see page 422)	
	LoadPermissions (see page 423)	Loads the permission for given object, such as content (see page 449), folder, task (see page 519), etc.
	UpdateItemPermission (see page 424)	Updates the item permissions (see page 419) for a user (see page 534) or a usergroup.

Legend

	Method

Permissions.AddItemPermission Method

Adds the item permissions (see page 419) for a user (see page 534) 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 534)'s Permissions (see page 419) Data

Returns

False if it's successful

Permissions.DeleteItemPermission Method

Deletes a user (see page 534) or a group from the item's permissions (see page 419)

C#

```
public Boolean DeleteItemPermission(int id, Boolean IsGroupId, int itemId,
Ektron.Cms.Common.EkEnumeration.CMSObjectTypes itemType);
```

Visual Basic

```
Public Function DeleteItemPermission(ByVal id As Integer, ByVal IsGroupId As Boolean, ByVal
itemId As Integer, ByVal itemType As Ektron.Cms.Common.EkEnumeration.CMSObjectTypes) As Boolean
```

Parameters

Parameters	Description
id	User (see page 534) or Group Id
IsGroupId	Given ID is a user (see page 534) id or a group id?
itemId	ContentId or FolderId
itemType	Given item id is EkEnumeration.CMSObjectTypes.Content (see page 449) or EkEnumeration.CMSObjectTypes.Folder (see page 383)?

Returns

False if it's successful.

Permissions.DisableContentInheritance Method

Disable permissions (see page 419) inheritance on the content (see page 449) and copies current permissions (see page 419) to the content (see page 449).

C#

```
public Boolean DisableContentInheritance(int id);
```

Visual Basic

```
Public Function DisableContentInheritance(ByVal id As Integer) As Boolean
```

Parameters

Parameters	Description
id	Content (see page 449) Id

Returns

False if it's successful.

Permissions.DisableFolderInheritance Method

Disable permissions (see page 419) inheritance on the folder and copies current permissions (see page 419) to the folder.

C#

```
public Boolean DisableFolderInheritance(int id);
```

Visual Basic

```
Public Function DisableFolderInheritance(ByVal id As Integer) As Boolean
```

Parameters

Parameters	Description
id	folder Id

Returns

False if it's successful.

Permissions.EnableContentInheritance Method

Deletes current permissions (see page 419) from the content (see page 449) and enable permissions (see page 419) inheritance on the content (see page 449).

C#

```
public Boolean EnableContentInheritance(int id);
```

Visual Basic

```
Public Function EnableContentInheritance(ByVal id As Integer) As Boolean
```

Parameters

Parameters	Description
id	Content (see page 449) Id

Returns

False if it's successful.

Permissions.EnableFolderInheritance Method

Deletes current permissions (see page 419) from the folder and enable permissions (see page 419) inheritance on the folder.

C#

```
public Boolean EnableFolderInheritance(int id);
```

Visual Basic

```
Public Function EnableFolderInheritance(ByVal id As Integer) As Boolean
```

Parameters

Parameters	Description
id	folder Id

Returns

False if it's successful.

GetUserPermissions Method

Overload List

	Name	Description
	Permissions.GetUserPermissions (Integer, String, Integer, String) (see page 422)	Loads the user (see page 534) permissions (see page 419) as a array of UserPermissionData.
	Permissions.GetUserPermissions (Integer, String, Integer, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) (see page 423)	Loads the user (see page 534) permissions (see page 419) as a array of UserPermissionData.

Permissions.GetUserPermissions Method (Integer, String, Integer, String)

Loads the user (see page 534) permissions (see page 419) as a array of UserPermissionData.

C#

```
public UserPermissionData[] GetUserPermissions(int Id, String ItemType, int UserId, String
UserGroupList);
```

Visual Basic

```
Public Function GetUserPermissions(ByVal Id As Integer, ByVal ItemType As String, ByVal UserId
```

```
As Integer, ByVal UserGroupList As String) As UserPermissionData()
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user (see page 534)'s ID
UserGroupList	the group list

Returns

Ektron.Cms.UserPermissionData()

Permissions.GetUserPermissions Method (Integer, String, Integer, String, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType)

Loads the user (see page 534) permissions (see page 419) as a array of UserPermissionData.

C#

```
public UserPermissionData[] GetUserPermissions(int Id, String ItemType, int UserId, String
UserGroupList, ContentAPI.PermissionUserType PermissionType, ContentAPI.PermissionRequestType
PermissionRequest);
```

Visual Basic

```
Public Function GetUserPermissions(ByVal Id As Integer, ByVal ItemType As String, ByVal UserId
As Integer, ByVal UserGroupList As String, ByVal PermissionType As
ContentAPI.PermissionUserType, ByVal PermissionRequest As ContentAPI.PermissionRequestType) As
UserPermissionData()
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user (see page 534)'s ID
UserGroupList	the group list
PermissionType	PermissionUserType
PermissionRequest	PermissionRequestType

Returns

Ektron.Cms.UserPermissionData()

Permissions.LoadPermissions Method

Loads the permission for given object, such as content (see page 449), folder, task (see page 519), etc.

C#

```
public PermissionData LoadPermissions(int Id, String RequestType,
ContentAPI.PermissionResultType Type);
```

Visual Basic

```
Public Function LoadPermissions(ByVal Id As Integer, ByVal RequestType As String, Optional
ByVal Type As ContentAPI.PermissionResultType = ContentAPI.PermissionResultType.Common) As
PermissionData
```

Parameters

Parameters	Description
Id	The object ID
RequestType	The object type := content (see page 449) | folder | collections | tasks

Type	The permission result type

Returns

Ektron.Cms.PermissionData

Remarks

User (see page 534) login with administrator privileges are required to perform this operation.

Permissions.UpdateItemPermission Method

Updates the item permissions (see page 419) for a user (see page 534) 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 534)'s Permissions (see page 419) Data

Returns

False if it's successful

Site Class

Class Hierarchy

C#

```
public class Site : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Site
Inherits Ektron.Cms.API.Common
```

File

Site.vb

Description

This is class API.Site.

Site Methods

The methods of the Site class are listed here.

Public Methods

	Name	Description
	BreakTaskPermissionInherit (see page 425)	Removes the task (see page 519) permissions for a given user (see page 534).
	DeleteCookie (see page 425)	Removes the cookie from the site (see page 424).
	GetAllActiveLanguages (see page 425)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language or Settings area.

	GetAllLanguages (see page 426)	
	GetClientDayNames (see page 426)	Returns the day name of a given content (see page 449) language (or site (see page 424) language).
	GetClientMonthNames (see page 427)	Returns the month name of a given content (see page 449) language (or site (see page 424) language)
	GetCookieValue (see page 427)	Reads the cookie value.
	GetLanguage (see page 427)	Returns the details about the given language ID.
	GetLanguages (see page 428)	Gets a list of all languages that meet a given state, for example, active.
	GetSiteVariables (see page 428)	Returns all of the settings data.
	GetTaskPermission (see page 428)	Returns 1 if the user (see page 534) has one of the task (see page 519) permissions, otherwise 0.
	IsTaskPermissionInherited (see page 429)	Returns true if a user (see page 534) has already granted one of the task (see page 519) permissions.
	SetCookieValue (see page 429)	Resets the cookie value.
	SetTaskPermission (see page 429)	Updates the task (see page 519) permission(s) for the given user (see page 534) or group.
	UpdateSiteEnabledLanguages (see page 429)	Updates the languages in the given list to be enabled on the site (see page 424). Languages not in the list are disabled.
	UpdateSiteVariables (see page 430)	Update the site (see page 424) settings

Legend

	Method

Site.BreakTaskPermissionInherit Method

Removes the task (see page 519) permissions for a given user (see page 534).

C#

```
public BreakTaskPermissionInherit(int Id);
```

Visual Basic

```
Public Sub BreakTaskPermissionInherit(ByVal Id As Integer)
```

Parameters

Parameters	Description
`Id`	User (see page 534) ID

Site.DeleteCookie Method

Removes the cookie from the site (see page 424).

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

Overload List

	Name	Description
	Site.GetAllLanguages () (see page 426)	Gets a list of all the available languages sorted by language name in ascending order.
	Site.GetAllLanguages (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) (see page 426)	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 (see page 449) language (or site (see page 424) language).

C#

```
public String GetClientDayNames();
```

Visual Basic

```
Public Function GetClientDayNames() As String
```

Returns

String

Site.GetClientMonthNames Method

Returns the month name of a given content (see page 449) language (or site (see page 424) language)

C#

```
public String GetClientMonthNames();
```

Visual Basic

```
Public Function GetClientMonthNames() As String
```

Returns

String

Site.GetCookieValue Method

Reads the cookie value.

C#

```
public String GetCookieValue(String Keyname);
```

Visual Basic

```
Public Function GetCookieValue(ByVal Keyname As String) As String
```

Parameters

Parameters	Description
Keyname	The Keyname. (Required)

Returns

String

Site.GetLanguage Method

Returns the details about the given language ID.

C#

```
public LanguageData GetLanguage(int Id);
```

Visual Basic

```
Public Function GetLanguage(ByVal Id As Integer) As LanguageData
```

Parameters

Parameters	Description
Id	Valid non negative language ID (refer to the language_type table for all the languages)

Returns

Ektron.Cms.LanguageData

Site.GetLanguages Method

Gets a list of all languages that meet a given state, for example, active.

C#

```
public LanguageDataSet.LanguageDataTable GetLanguages(LanguageDataSet.LanguageState State,
LanguageDataSet.SortBy SortBy, LanguageDataSet.SortDirection SortDirection);
```

Visual Basic

```
Public Function GetLanguages(ByVal State As LanguageDataSet.LanguageState, Optional ByVal
SortBy As LanguageDataSet.SortBy = LanguageDataSet.SortBy.LanguageState, Optional ByVal
SortDirection As LanguageDataSet.SortDirection = LanguageDataSet.SortDirection.Ascending) As
LanguageDataSet.LanguageDataTable
```

Parameters

Parameters	Description
State	The state that the language must meet. For example, defined, active, site (see page 424) enabled.
SortBy	The field on which to sort the languages in the data table.
SortDirection	The direction, ascending or descending, in which to sort the field values.

Returns

LanguageDataTable of languages that meet the given state.

Site.GetSiteVariables Method

Returns all of the settings data.

C#

```
public SettingsData GetSiteVariables(int Id);
```

Visual Basic

```
Public Function GetSiteVariables(Optional ByVal Id As Integer = -1) As SettingsData
```

Parameters

Parameters	Description
Id	User (see page 534) ID

Returns

Ektron.Cms.SettingsData

Remarks

If user (see page 534) ID is passed it returns the additional value of MaxUsersAllowed, AdInstalled and AdAllowed properties.

Site.GetTaskPermission Method

Returns 1 if the user (see page 534) has one of the task (see page 519) permissions, otherwise 0.

C#

```
public int GetTaskPermission(int Id, int Type);
```

Visual Basic

```
Public Function GetTaskPermission(ByVal Id As Integer, ByVal Type As Integer) As Integer
```

Parameters

Parameters	Description
Id	User (see page 534) or group ID
Type	User (see page 534) or group (1 = user (see page 534), 2 = group).

Returns

Integer

Site.IsTaskPermissionInherited Method

Returns true if a user (see page 534) has already granted one of the task (see page 519) permissions.

C#

```
public Boolean IsTaskPermissionInherited(int Id);
```

Visual Basic

```
Public Function IsTaskPermissionInherited(ByVal Id As Integer) As Boolean
```

Parameters

Parameters	Description
Id	User (see page 534) 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 (see page 519) permission(s) for the given user (see page 534) or group.

C#

```
public SetTaskPermission(int Id, int UserType, int PermissionType);
```

Visual Basic

```
Public Sub SetTaskPermission(ByVal Id As Integer, ByVal UserType As Integer, ByVal
PermissionType As Integer)
```

Parameters

Parameters	Description
Id	User (see page 534) ID or Group ID
UserType	User (see page 534) (1) or Group (2)
PermissionType	Ektron.Cms.Common.TaskPermission (1 = Create Task (see page 519), 2 = Redirect Task (see page 519), 3 = Delete Task (see page 519), 4 = Edit Alias)

Site.UpdateSiteEnabledLanguages Method

Updates the languages in the given list to be enabled on the site (see page 424). 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 (see page 424) 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 API.SiteMap.

SiteMap Methods

The methods of the SiteMap class are listed here.

Public Methods

	Name	Description
	GetFolderBreadcrumbPath (see page 431)	Get breadcrumb path of the folder in which content (see page 449) block exists.
	GetWebsiteMap (see page 432)	Creates the site map from folder breadcrumb.
	RenderFolderBreadcrumbHtml (see page 434)	Render folder breadcrumb HTML.
	RenderWebsiteMap (see page 434)	Create HTML to render on the page.

Legend

	Method

SiteMap.GetFolderBreadcrumbPath Method

Get breadcrumb path of the folder in which content (see page 449) block exists.

C#

```
public Ektron.Cms.Common.SitemapPath[] GetFolderBreadcrumbPath(int id, Boolean isFolder);
```

Visual Basic

```
Public Function GetFolderBreadcrumbPath(ByVal id As Integer, Optional ByVal isFolder As
Boolean = False) As Ektron.Cms.Common.SitemapPath()
```

Parameters

Parameters	Description
id	Content (see page 449) or Folder (see page 383) 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 (see page 449)'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 424) Map (see page 232) API (see page 363).

```
        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(int folderId, Boolean getSameLevelNodes);
```

Visual Basic

```
Public Function GetWebsiteMap(Optional ByVal folderId As Integer = 0, Optional ByVal
getSameLevelNodes As Boolean = False) As WebsiteMap
```

Parameters

Parameters	Description
`folderId`	Build the web sitemap (see page 430) from the folder specified.
`getSameLevelNodes`	Do you want to retrieve all nodes at the same level as the one you specify? **True** = include all sites at the same level as the `folderId` property. **False** = Do not include the same level nodes.

Example

The following example shows how to create a Web page that displays the site map for a given folder. This example uses some standard drag and drop controls and a small section of VB code utilizing the GetWebsiteMap method. This method uses the InternalAdmin to access and return the site map information. In the first section, we will deal with the standard drag and drop controls. In the second, we'll deal with the code behind.

1. Between the form tags, add the following label:
```
<asp:Label ID="Label6" runat="server" Text="Folder ID:" Width="66px"></asp:Label>
```
2. Add text box where the folder ID can be entered.
```
<asp:TextBox ID="txtFolderId" runat="server" Width="55px"></asp:TextBox>
```
3. From the Validation control menu, add a RequiredFieldValidator control. This validates that something appears in the Folder ID box.
```
<asp:RequiredFieldValidator ID="rfvId" runat="server" ControlToValidate="txtFolderId"
   ErrorMessage="Please enter folder ID" Width="179px"></asp:RequiredFieldValidator>
```
4. Add two breaks for spacing purposes.
```
<br /><br />
```
5. Add a button to submit the folder ID and retrieve the site map information. We will set the button click event in the code behind.
```
<asp:Button ID="btnGet" runat="server" Text="Get site map" Width="91px" />
```
6. Add two more breaks for spacing purposes.
```
<br /><br />
```
7. Add a label to display the site map.
```
<asp:Label ID="lblResult" runat="server" Width="502px" Height="100px"></asp:Label>
```

Add the following information to the code behind of your aspx.vb page.

1. Add a button click event.
```
    Protected Sub btnGet_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles
btnGet.Click
```
2. Check to make sure the folder ID is numeric.
```
        If Not IsNumeric(txtFolderId.Text) Then
            lblResult.Text = "Invalid Folder ID"
            Exit Sub
        End If
```
3. Create an object for the site map API (see page 363).
```
        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? **True** = hyperlink breadcrumb items. **False** = breadcrumb items appear as plain text.
linkTarget	Determines the type of window that appears on this Web form when a user (see page 534) 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.
displayVertical	Do you want the breadcrumb path to display vertically? **True** = display the breadcrumb trail vertically. **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, String className,
int maxLevel, int startLevel, String PageUrl);
```

Visual Basic

```
Public Function RenderWebsiteMap(ByVal webSitemap As WebsiteMap, ByVal displayType As String,
Optional ByVal className As String = "ektSitemap", Optional ByVal maxLevel As Integer = 0,
Optional ByVal startLevel As Integer = 0, Optional ByVal PageUrl As String = "") As String
```

Parameters

Parameters	Description
webSitemap	GetWebSiteMap()
displayType	Determines how to produce HTML.
className	Stylesheet class to format the HTML. It must exists in [site]/workarea/csslib/sitemap.css. Leave blank to accept default.
maxLevel	Set the max level of the sitemap (see page 430). Set it to 0 for unlimited.
startLevel	Set the starting level of the sitemap (see page 430). Set it to 0 for start from root.

Returns

HTML/XML

Remarks

Supported displayType: ListWithDescription | List | RawXml

Subscription Class

Class Hierarchy

C#

```
public class Subscription : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Subscription
Inherits Ektron.Cms.API.Common
```

File

Subscription.vb

Description

This is class API.Subscription.

Subscription Methods

The methods of the Subscription class are listed here.

Public Methods

	Name	Description
	AddEmailFrom (see page 436)	Adds a subscription (see page 435) email address.
	AddSubscription (see page 436)	Adds a subscription (see page 435).
	AddSubscriptionMessage (see page 436)	Adds subscription (see page 435) message(s)
	DeleteEmailFrom (see page 437)	Deletes a subscription (see page 435) email address from system.
	DeleteSubscription (see page 437)	Deletes a subscription (see page 435) from the system.
	DeleteSubscriptionMessage (see page 437)	Deletes the subscription (see page 435) message from the system.
	GetAllActiveSubscriptions (see page 437)	Loads all of the active subscriptions.
	GetAllEmailFrom (see page 438)	Returns all of the email as an array of EmailFromData.
	GetAllSubscriptionMessages (see page 438)	Loads all the subscription (see page 435) messages.
	GetAllSubscriptions (see page 438)	Loads all the subscriptions.
	GetFolderInheritedFrom (see page 438)	Retrieves the inherited parent ID
	GetSubscription (see page 439)	Retrieves a subscription (see page 435) by using its ID.
	GetSubscriptionMessage (see page 439)	Retrieves a subscription (see page 435) message by using the subscription (see page 435) ID

	GetSubscriptionMessagesForType (see page 439)	Retrieve subscription (see page 435) message for type.
	GetSubscriptionPropertiesForContent (see page 439)	Retrieves the subscription (see page 435) properties for content (see page 449) using the content (see page 449)'s ID.
	GetSubscriptionPropertiesForFolder (see page 440)	Retrieves the subscription (see page 435) properties for a given folder.
	GetSubscriptionsForContent (see page 440)	Loads the subscriptions for the given content (see page 449).
	GetSubscriptionsForFolder (see page 440)	Loads the subscriptions for a given folder.
	UpdateEmailFrom (see page 441)	Update the email address.
	UpdateSubscription (see page 441)	Updates the subscription (see page 435).
	UpdateSubscriptionMessage (see page 441)	Updates the subscription (see page 435) message.

Legend

	Method

Subscription.AddEmailFrom Method

Adds a subscription (see page 435) email address.

C#

```
public AddEmailFrom(Collection data);
```

Visual Basic

```
Public Sub AddEmailFrom(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Email address as a collection

Subscription.AddSubscription Method

Adds a subscription (see page 435).

C#

```
public AddSubscription(Collection data);
```

Visual Basic

```
Public Sub AddSubscription(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Subscription (see page 435) name

Subscription.AddSubscriptionMessage Method

Adds subscription (see page 435) message(s)

C#

```
public AddSubscriptionMessage(Collection data);
```

Visual Basic

```
Public Sub AddSubscriptionMessage(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Subscription (see page 435) message data

Subscription.DeleteEmailFrom Method

Deletes a subscription (see page 435) 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 (see page 435) from the system.

C#

```
public DeleteSubscription(Collection data);
```

Visual Basic

```
Public Sub DeleteSubscription(ByVal data As Collection)
```

Parameters

Parameters	Description
data	Subscription (see page 435) data

Subscription.DeleteSubscriptionMessage Method

Deletes the subscription (see page 435) 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 435) 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 (see page 435) 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 int GetFolderInheritedFrom(int FolderId);
```

Visual Basic

```
Public Function GetFolderInheritedFrom(ByVal FolderId As Integer) As Integer
```

Parameters

Parameters	Description
`FolderId`	The folder's ID

Returns

Integer

Subscription.GetSubscription Method

Retrieves a subscription (see page 435) by using its ID.

C#

```
public SubscriptionData GetSubscription(int Id);
```

Visual Basic

```
Public Function GetSubscription(ByVal Id As Integer) As SubscriptionData
```

Parameters

Parameters	Description
Id	The subscription (see page 435)'s ID

Returns

Ektron.Cms.SubscriptionData

Subscription.GetSubscriptionMessage Method

Retrieves a subscription (see page 435) message by using the subscription (see page 435) ID

C#

```
public EmailMessageData GetSubscriptionMessage(int Id);
```

Visual Basic

```
Public Function GetSubscriptionMessage(ByVal Id As Integer) As EmailMessageData
```

Parameters

Parameters	Description
Id	The subscription (see page 435)'s ID

Returns

Ektron.Cms.EmailMessageData

Subscription.GetSubscriptionMessagesForType Method

Retrieve subscription (see page 435) message for type.

C#

```
public EmailMessageData[]
GetSubscriptionMessagesForType(Ektron.Cms.Common.EkEnumeration.EmailMessageTypes MessageType);
```

Visual Basic

```
Public Function GetSubscriptionMessagesForType(ByVal MessageType As
Ektron.Cms.Common.EkEnumeration.EmailMessageTypes) As EmailMessageData()
```

Parameters

Parameters	Description
MessageType	The message type

Returns

Ektron.Cms.EmailMessageData()

Subscription.GetSubscriptionPropertiesForContent Method

Retrieves the subscription (see page 435) properties for content (see page 449) using the content (see page 449)'s ID.

C#

```
public SubscriptionPropertiesData GetSubscriptionPropertiesForContent(int ContentId);
```

Visual Basic

```
Public Function GetSubscriptionPropertiesForContent(ByVal ContentId As Integer) As
SubscriptionPropertiesData
```

Parameters

Parameters	Description
ContentId	The content (see page 449)'s ID

Returns

Ektron.Cms.SubscriptionPropertiesData

Subscription.GetSubscriptionPropertiesForFolder Method

Retrieves the subscription (see page 435) properties for a given folder.

C#

```
public SubscriptionPropertiesData GetSubscriptionPropertiesForFolder(int FolderId);
```

Visual Basic

```
Public Function GetSubscriptionPropertiesForFolder(ByVal FolderId As Integer) As
SubscriptionPropertiesData
```

Parameters

Parameters	Description
FolderId	The folder's ID

Returns

Ektron.Cms.SubscriptionPropertiesData

Subscription.GetSubscriptionsForContent Method

Loads the subscriptions for the given content (see page 449).

C#

```
public SubscriptionData[] GetSubscriptionsForContent(int ContentId);
```

Visual Basic

```
Public Function GetSubscriptionsForContent(ByVal ContentId As Integer) As SubscriptionData()
```

Parameters

Parameters	Description
ContentId	The content (see page 449)'s ID

Returns

Ektron.Cms.SubscriptionData()

Subscription.GetSubscriptionsForFolder Method

Loads the subscriptions for a given folder.

C#

```
public SubscriptionData[] GetSubscriptionsForFolder(int FolderId);
```

Visual Basic

```
Public Function GetSubscriptionsForFolder(ByVal FolderId As Integer) As SubscriptionData()
```

Parameters

Parameters	Description
FolderId	The folder's ID

Returns

Ektron.Cms.SubscriptionData()

Subscription.UpdateEmailFrom Method

Update the email address.

C#

```
public UpdateEmailFrom(Collection data);
```

Visual Basic

```
Public Sub UpdateEmailFrom(ByVal data As Collection)
```

Parameters

Parameters	Description
data	email data

Subscription.UpdateSubscription Method

Updates the subscription (see page 435).

C#

```
public UpdateSubscription(Collection data);
```

Visual Basic

```
Public Sub UpdateSubscription(ByVal data As Collection)
```

Parameters

Parameters	Description
data	The subscription (see page 435) details

Subscription.UpdateSubscriptionMessage Method

Updates the subscription (see page 435) message.

C#

```
public UpdateSubscriptionMessage(Collection data);
```

Visual Basic

```
Public Sub UpdateSubscriptionMessage(ByVal data As Collection)
```

Parameters

Parameters	Description
data	The subscription (see page 435) message details.

Utilities Class

Class Hierarchy

C#

```
public class Utilities : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Utilities
Inherits Ektron.Cms.API.Common
```

File

Utilities.vb

Description

This is class API.Utilities.

Utilities Methods

The methods of the Utilities class are listed here.

Public Methods

	Name	Description
	LoadPermissions (see page 442)	Loads the permission for given object, such as content (see page 449), folder, task (see page 519), etc.

Legend

	Method

Utilities.LoadPermissions Method

Loads the permission for given object, such as content (see page 449), folder, task (see page 519), etc.

C#

```
[Obsolete("Use LoadPermissions from the Ektron.CMS.API.Permissions namespace.")]
public PermissionData LoadPermissions(int Id, String RequestType, PermissionResultType Type);
```

Visual Basic

```
<Obsolete("Use LoadPermissions from the Ektron.CMS.API.Permissions namespace.")>
Public Function LoadPermissions(ByVal Id As Integer, ByVal RequestType As String, Optional
ByVal Type As PermissionResultType = PermissionResultType.Common) As PermissionData
```

Parameters

Parameters	Description
`Id`	The object ID
`RequestType`	The object type := content (see page 449) | folder | collections | tasks
`Type`	The permission result type

Returns

Ektron.Cms.PermissionData

Remarks

Use LoadPermissions from the Ektron.CMS.API.Permissions (see page 419) namespace.

API.Analytics Namespace

This is namespace API.Analytics.

Classes

	Name	Description
	Analytics (see page 443)	A class for managing and querying CMS400.NET site activity information

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Analytics (see page 443)	A class for managing and querying CMS400.NET site activity information

Legend

	Class

Analytics Class

A class for managing and querying CMS400.NET site activity information

Class Hierarchy

C#

```
public class Analytics : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Analytics
Inherits Ektron.Cms.API.Common
```

File

Analytics.vb

Analytics Methods

The methods of the Analytics class are listed here.

Public Methods

	Name	Description
	GetSessionRank (see page 444)	Gets the Session rank of the visiting user (see page 534). A session rank of 0 is the first page a new user (see page 534) has visited. A session rank of 1 is the first page a returning user (see page 534) has visited. A session rank of 2 is any other page.
	HardDumpAnalyticsBuffer (see page 444)	Forces the analytics (see page 443) 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 444)	Provides a simple way of querying the database for analytics (see page 443) data. Returns the result as a weakly typed DataSet. Note that the connection is to the Analytics (see page 443) database and not to the standard CMS database.
	SoftDumpAnalyticsBuffer (see page 444)	Conditionally writes the analytics (see page 443) 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 445)	Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer. This method is overloaded.

Legend

	Method

Analytics.GetSessionRank Method

Gets the Session rank of the visiting user (see page 534). A session rank of 0 is the first page a new user (see page 534) has visited. A session rank of 1 is the first page a returning user (see page 534) has visited. A session rank of 2 is any other page.

C#

```
public int GetSessionRank(System.Web.UI.Page page);
```

Visual Basic

```
Public Function GetSessionRank(ByVal page As System.Web.UI.Page) As Integer
```

Parameters

Parameters	Description
page	

Analytics.HardDumpAnalyticsBuffer Method

Forces the analytics (see page 443) buffer to write to the database. This surpasses the checks to see if the time before write or number of records have passed.

C#

```
public HardDumpAnalyticsBuffer();
```

Visual Basic

```
Public Sub HardDumpAnalyticsBuffer()
```

Analytics.QueryAnalytics Method

Provides a simple way of querying the database for analytics (see page 443) data. Returns the result as a weakly typed DataSet. Note that the connection is to the Analytics (see page 443) database and not to the standard CMS database.

C#

```
public DataSet QueryAnalytics(String selectCommands);
```

Visual Basic

```
Public Function QueryAnalytics(ByVal selectCommands() As String) As DataSet
```

Parameters

Parameters	Description
selectCommands	An array of SQL queries in string format.

Returns

A weakly typed dataset with the result of the query.

Analytics.SoftDumpAnalyticsBuffer Method

Conditionally writes the analytics (see page 443) buffer in memory to write to the database. If the time before write or number of records before write conditions have been met or exceeded, this function will force the write to the database.

C#

```
public Boolean SoftDumpAnalyticsBuffer();
```

Visual Basic

```
Public Function SoftDumpAnalyticsBuffer() As Boolean
```

TrackVisit Method

Tracks a visit to a page, and places it into the analytics buffer. This method calls SoftDumpAnalyticsBuffer. This method is overloaded.

Overload List

	Name	Description
	Analytics.TrackVisit (Integer, Integer, String, Integer, String, String) (see page 445)	Tracks a visit to a page, and places it into the analytics (see page 443) buffer. This method calls SoftDumpAnalyticsBuffer (see page 444).
	Analytics.TrackVisit (System.Web.UI.Page, Integer) (see page 445)	Tracks a visit to a page, and places it into the analytics (see page 443) buffer. This method calls SoftDumpAnalyticsBuffer (see page 444).

Analytics.TrackVisit Method (Integer, Integer, String, Integer, String, String)

Tracks a visit to a page, and places it into the analytics (see page 443) buffer. This method calls SoftDumpAnalyticsBuffer (see page 444).

C#

```
public TrackVisit(int content_id, int user_id, String visitor_id, int session_rank, String
referring_url, String url);
```

Visual Basic

```
Public Sub TrackVisit(ByVal content_id As Integer, ByVal user_id As Integer, ByVal visitor_id
As String, ByVal session_rank As Integer, ByVal referring_url As String, ByVal url As String)
```

Parameters

Parameters	Description
content_id	The ID of a content (see page 449) block that a visitor has viewed.
user_id	The CMS user (see page 534) ID of either a CMS User (see page 534) 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 (see page 534) has visited. A session rank of 1 is the first page a returning user (see page 534) has visited. A session rank of 2 is any other page.
referring_url	An external URL a user (see page 534) has arrived from.
url	The URL the user (see page 534) is visiting.

Analytics.TrackVisit Method (System.Web.UI.Page, Integer)

Tracks a visit to a page, and places it into the analytics (see page 443) buffer. This method calls SoftDumpAnalyticsBuffer (see page 444).

C#

```
public TrackVisit(System.Web.UI.Page page, int content_id);
```

Visual Basic

```
Public Sub TrackVisit(ByVal page As System.Web.UI.Page, ByVal content_id As Integer)
```

Parameters

Parameters	Description
`page`	The System.Web.UI.Page object of a page visit. Will extract information from it such as visitor id, user (see page 534) id, session rank, and so forth.
`content_id`	The ID of a content (see page 449) block that a visitor has viewed.

Analytics Properties

The properties of the Analytics class are listed here.

Public Properties

	Name	Description
R	AnalyticsEnabled (see page 446)	Indicates if Analytics (see page 443) has been enabled through a setting in the web.config file. If disabled, no analytics (see page 443) 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 17) server control.
R	AppendToIISLog (see page 446)	Indicates whether or not the web.config has been set to append user_id and content_id information to the IIS log.
R	InMemoryContentHitsData (see page 447)	The dataset that stores all the page/content (see page 449) view information in memory until it reaches its record or time threshold and writes back to the database. This is a strongly typed dataset of type 'ContentHitsData'. The underlying static member is initialized on demand.
R	RecordsBeforeWrite (see page 447)	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 447)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.
R	TimeBeforeWrite (see page 447)	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 447)' property are evaluated and, if any value has been exceeded, the buffered data will be written to the database.

Legend

	Property
R	read only

Analytics.AnalyticsEnabled Property

Indicates if Analytics (see page 443) has been enabled through a setting in the web.config file. If disabled, no analytics (see page 443) 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 17) server control.

C#

```
public ReadOnly Boolean AnalyticsEnabled;
```

Visual Basic

```
Public ReadOnly Property AnalyticsEnabled() As Boolean
```

Description

True indicates that analytics (see page 443) data is being tracked. False indicates that no analytics (see page 443) data is being tracked.

Analytics.AppendToIISLog Property

Indicates whether or not the web.config has been set to append user_id and content_id information to the IIS log.

C#

```
public ReadOnly Boolean AppendToIISLog;
```

Visual Basic

```
Public ReadOnly Property AppendToIISLog() As Boolean
```

Description

Boolean

Analytics.InMemoryContentHitsData Property

The dataset that stores all the page/content (see page 449) 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 447)' 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 447)' 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
```

API.Calendar Namespace

This is namespace API.Calendar.

Classes

	Name	Description
	Calendar (see page 448)	This is class API.Calendar.Calendar.
	CalendarEvent (see page 448)	This is class API.Calendar.CalendarEvent.
	CalendarEventType (see page 449)	This is class API.Calendar.CalendarEventType.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Calendar (see page 448)	This is class API.Calendar.Calendar.
	CalendarEvent (see page 448)	This is class API.Calendar.CalendarEvent.
	CalendarEventType (see page 449)	This is class API.Calendar.CalendarEventType.

Legend

	Class

Calendar Class

Class Hierarchy

C#

```
public class Calendar : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Calendar
Inherits Ektron.Cms.API.Common
```

File

Calendar.vb

Description

This is class API.Calendar.Calendar.

CalendarEvent Class

Class Hierarchy

C#

```
public class CalendarEvent : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class CalendarEvent
Inherits Ektron.Cms.API.Common
```

File

CalendarEvent.vb

Description

This is class API.Calendar.CalendarEvent.

CalendarEventType Class

Class Hierarchy

C#

```
public class CalendarEventType : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class CalendarEventType
Inherits Ektron.Cms.API.Common
```

File

CalendarEventType.vb

Description

This is class API.Calendar.CalendarEventType.

API.Content Namespace

This is namespace API.Content.

Classes

	Name	Description
	Asset (see page 450)	This is class API.Content.Asset.
	Blog (see page 454)	This is class API.Content.Blog.
	Content (see page 468)	This is class API.Content.Content.
	ContentRating (see page 488)	This is class API.Content.ContentRating.
	Form (see page 493)	This is class API.Content.Form.
	Taxonomy (see page 498)	This is class API.Content.Taxonomy.
	ThreadedDiscussion (see page 503)	This is class API.Content.ThreadedDiscussion.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Asset (see page 450)	This is class API.Content.Asset.
	Blog (see page 454)	This is class API.Content.Blog.
	Content (see page 468)	This is class API.Content.Content.
	ContentRating (see page 488)	This is class API.Content.ContentRating.
	Form (see page 493)	This is class API.Content.Form.
	Taxonomy (see page 498)	This is class API.Content.Taxonomy.
	ThreadedDiscussion (see page 503)	This is class API.Content.ThreadedDiscussion.

Legend

	Class

Asset Class

Class Hierarchy

C#

```
public class Asset : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Asset
Inherits Ektron.Cms.API.Common
```

File

Asset.vb

Description

This is class API.Content.Asset.

Asset Methods

The methods of the Asset class are listed here.

Public Methods

	Name	Description
	GetAssetMgtConfigInfo (see page 450)	Loads all of the registered plug-in configuration details. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	GetAssetMgtInfo (see page 451)	Loads all of the registered plug-in type details. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	GetAssetSupertypes (see page 451)	Loads all of the asset (see page 450) supertypes supported by the documents add-in. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	GetCreateDragDropSnippet (see page 451)	Loads the drag and drop client snippet used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	GetCreateEktronExplorerSnippet (see page 452)	Loads the explorer snippet used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	GetCreateSnippet (see page 452)	Loads the create document snippet used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	GetDragDropWOSnippet (see page 452)	Loads the edit snippet for the given document when working offline. Used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	GetEditSnippet (see page 453)	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 450)
	GetMyAssetMgtID (see page 453)	Retrieves the Document Management functionality client ID. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	GetViewUrl (see page 453)	Returns a URL associated with the current asset (see page 450). This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	MakeCopyDMSAsset (see page 454)	Create a copy of a DMS Asset (see page 450). This method is a member of Ektron.Cms.API.Content.Asset (see page 450)
	SetAssetMgtConfigInfo (see page 454)	Updates the Document Management configuration information. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

Legend

	Method

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

450)

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 450)

C#

```
public AssetInfoData[] GetAssetMgtInfo();
```

Visual Basic

```
Public Function GetAssetMgtInfo() As AssetInfoData()
```

Returns

Ektron.Cms.AssetInfoData()

Asset.GetAssetSupertypes Method

Loads all of the asset (see page 450) supertypes supported by the documents add-in. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

C#

```
public AssetInfoData[] GetAssetSupertypes();
```

Visual Basic

```
Public Function GetAssetSupertypes() As AssetInfoData()
```

Returns

Ektron.Cms.AssetInfoData()

Asset.GetCreateDragDropSnippet Method

Loads the drag and drop client snippet used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

C#

```
public String GetCreateDragDropSnippet();
```

Visual Basic

```
Public Function GetCreateDragDropSnippet() As String
```

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetCreateEktronExplorerSnippet Method

Loads the explorer snippet used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

C#

```
public String GetCreateEktronExplorerSnippet();
```

Visual Basic

```
Public Function GetCreateEktronExplorerSnippet() As String
```

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetCreateSnippet Method

Loads the create document snippet used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

C#

```
public String GetCreateSnippet(int ContentType);
```

Visual Basic

```
Public Function GetCreateSnippet(ByVal ContentType As Integer) As String
```

Parameters

Parameters	Description
ContentType	The type of the content (see page 449), such as office documents or managed files or * (all).

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetDragDropWOSnippet Method

Loads the edit snippet for the given document when working offline. Used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

C#

```
public String GetDragDropWOSnippet(String AssetId, int ContentType);
```

Visual Basic

```
Public Function GetDragDropWOSnippet(ByVal AssetId As String, ByVal ContentType As Integer) As
String
```

Parameters

Parameters	Description
AssetId	The document's ID (Required)
ContentType	The type of the content (see page 449), such as office documents or managed files or * (all).

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetEditSnippet Method

Loads the edit snippet for the given document. Used with the Document Management functionality. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

C#

```
public String GetEditSnippet(String AssetId, int ContentType);
```

Visual Basic

```
Public Function GetEditSnippet(ByVal AssetId As String, ByVal ContentType As Integer) As String
```

Parameters

Parameters	Description
AssetId	The document's ID (Required)
ContentType	The type of the content (see page 449), 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 450)

C#

```
public String GetMyAssetMgtID();
```

Visual Basic

```
Public Function GetMyAssetMgtID() As String
```

Returns

String

Asset.GetViewUrl Method

Returns a URL associated with the current asset (see page 450). This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

C#

```
public String GetViewUrl(String AssetId, int ContentType);
```

Visual Basic

```
Public Function GetViewUrl(ByVal AssetId As String, ByVal ContentType As Integer) As String
```

Parameters

Parameters	Description
AssetId	The document's ID (Required)
ContentType	The type of the content (see page 449), 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 450). This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

C#

```
public Boolean MakeCopyDMSAsset(String strAssetID, String strAssetVersion);
```

Visual Basic

```
Public Function MakeCopyDMSAsset(ByRef strAssetID As String, ByRef strAssetVersion As String)
As Boolean
```

Parameters

Parameters	Description
strAssetID	ID associated with the asset (see page 450)
strAssetVersion	Version of the asset (see page 450)

Returns

boolean

Asset.SetAssetMgtConfigInfo Method

Updates the Document Management configuration information. This method is a member of Ektron.Cms.API.Content.Asset (see page 450)

C#

```
public Boolean SetAssetMgtConfigInfo(AssetConfigInfo asset_config, ));
```

Visual Basic

```
Public Function SetAssetMgtConfigInfo(ByVal asset_config As AssetConfigInfo,  As )) As Boolean
```

Parameters

Parameters	Description
asset_config	The Document Management configuration data

Returns

Boolean

Blog Class

Class Hierarchy

C#

```
public class Blog : Content;
```

Visual Basic

```
Public Class Blog
Inherits Content
```

File

Blog.vb

Description

This is class API.Content.Blog.

Blog Methods

The methods of the Blog class are listed here.

Public Methods

	Name	Description
	AddBlog (see page 455)	Adds a blog (see page 454) to the ParentFolderID. Accepts standard blog (see page 454) properties. Returns the ID of the newly created blog (see page 454).
	AddCommentForPost (see page 456)	Adds a post comment to a blog (see page 454) 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 456)	Adds a post to a given blog (see page 454). Accepts a ContentData (see page 738) class as the post content (see page 449). Return new post ID. *See Also:* The Example section.
	DeleteBlog (see page 458)	Deletes a blog (see page 454) with a given ID. This will delete the blog (see page 454) as well as all posts and comments for the blog (see page 454). BlogID must be greater than 0.
	DeletePost (see page 458)	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 459)	Returns a blank BlogPostData class. Used Primarily in the Workarea.
	GetBlog (see page 459)	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.
	GetBlogData (see page 462)	Overloaded Method to obtain a BlogData Class.
	GetBlogDataForPost (see page 465)	Returns the blog (see page 454) information for a blog (see page 454) post. This contains information about categories, whether comments are enabled, etc. Accepts the post ID.
	GetBlogPostData (see page 466)	Gets the Post data for a specific blog (see page 454) post. Accepts the ID of the post. Will return tags, trackback, pingback, and categories for the post.
	GetBlogRoll (see page 466)	Returns a populated blog roll class. This blog roll is the list of links for the blog. See the BlogRoll class for the properties and usage of the blog roll. This is an overloaded method.
	GetPost (see page 467)	Gets a blog (see page 454) post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData (see page 738) Class.
	GetPostbyID (see page 468)	Gets a blog (see page 454) post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class.

Legend

	Method

Blog.AddBlog Method

Adds a blog (see page 454) to the ParentFolderID. Accepts standard blog (see page 454) properties. Returns the ID of the newly created blog (see page 454).

C#

```
public int AddBlog(int ParentFolderID, String BlogName, String BlogTitle, String
BlogDescription, Ektron.Cms.Common.EkEnumeration.BlogVisibility Visibility, Boolean
```

```
EnableComments, Boolean ModerateComments, Boolean RequireAuthentication, String Categories, ),
BlogRoll RollLinks);
```

Visual Basic

```
Public Function AddBlog(ByVal ParentFolderID As Integer, ByVal BlogName As String, ByVal
BlogTitle As String, ByVal BlogDescription As String, ByVal Visibility As
Ektron.Cms.Common.EkEnumeration.BlogVisibility, ByVal EnableComments As Boolean, ByVal
ModerateComments As Boolean, ByVal RequireAuthentication As Boolean, ByVal Categories As
String,  As ), ByVal RollLinks As BlogRoll) As Integer
```

Parameters

Parameters	Description
ParentFolderID	
BlogName	Name of the Blog (see page 454).
BlogTitle	Title of the Blog (see page 454).
BlogDescription	Description of the Blog (see page 454).
Visibility	Is the blog (see page 454) 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 (see page 454).
RollLinks	The Blog (see page 454)'s roll links.

Returns

Integer value of the blog (see page 454) ID that is greater than 0 (zero).

Blog.AddCommentForPost Method

Adds a post comment to a blog (see page 454) 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 (see page 454). Accepts a ContentData (see page 738) class as the post content (see page 449). Return new post ID.

See Also: The Example section.

C#

```
public int AddPost(int blogID, ContentData PostContent, String Categories, ), Boolean
Pingback, String Tags, String TrackbackURL);
```

Visual Basic

```
Public Function AddPost(ByVal blogID As Integer, ByVal PostContent As ContentData, ByVal
Categories As String,  As ), ByVal Pingback As Boolean, ByVal Tags As String, ByVal
```

```
TrackbackURL As String) As Integer
```

Parameters

Parameters	Description
blogID	Integer ID of the blog (see page 454) to which the post is added.
PostContent	ContentData (see page 738) 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

Integer ID of the post created.

Example

The following example shows how to create a Web page from which a user (see page 534) can add a blog (see page 454) post. A user (see page 534) enters the blog (see page 454)'s ID, post title and content (see page 449) 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 (see page 454) 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 (see page 454) 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 468) of the blog (see page 454) post.

```
         postContentData.Title = txtPostTitle.text
         postContentData.Html = txtPostContent.Text
```

6. Create a place holder for blog (see page 454) post's ID using the AddPost method.

```
         Dim postId As Integer
         postId = blogApi.AddPost(CInt(txtBlogId.Text), postContentData, categories, False,
"", "")
```

7. Let the user (see page 534) 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 (see page 534) know what happened.

```
           Catch ex As Exception
              lblResult.Text = ex.Message
           End Try

    End Sub
```

Blog.DeleteBlog Method

Deletes a blog (see page 454) with a given ID. This will delete the blog (see page 454) as well as all posts and comments for the blog (see page 454). BlogID must be greater than 0.

C#

```
public DeleteBlog(int BlogID);
```

Visual Basic

```
Public Sub DeleteBlog(ByVal BlogID As Integer)
```

Parameters

Parameters	Description
BlogID	The integer ID of the blog (see page 454). This corresponds to the folder ID of the blog (see page 454).

Blog.DeletePost Method

Deletes a post by the post's ID This method will delete all comments associated with the post as well. Returns boolean value.

C#

```
public Boolean DeletePost(int post_id);
```

Visual Basic

```
Public Function DeletePost(ByVal post_id As Integer) As Boolean
```

Parameters

Parameters	Description
`post_id`	Integer value of the post ID to delete.

Returns

Boolean

Blog.GetBlankBlogPostData Method

Returns a blank BlogPostData class. Used Primarily in the Workarea.

C#

```
public BlogPostData GetBlankBlogPostData();
```

Visual Basic

```
Public Function GetBlankBlogPostData() As BlogPostData
```

Returns

A blank, but not null, BlogPostData class.

GetBlog Method

Returns a BlogData class for a given blog, 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 (Integer, String, Integer, Integer) (see page 459)	Returns a BlogData class for a given blog (see page 454) ID, with parameters. This class contains the content (see page 449) for a blog (see page 454), the post info, the roll, and the categories information. See the BlogData class for more information.
	Blog.GetBlog (String, String, Integer, Integer) (see page 462)	Returns a BlogData class for a given path, with parameters. This class contains the content (see page 449) for a blog (see page 454), the post info, the roll, and the categories information. See the BlogData class for more information.

Blog.GetBlog Method (Integer, String, Integer, Integer)

Returns a BlogData class for a given blog (see page 454) ID, with parameters. This class contains the content (see page 449) for a blog (see page 454), the post info, the roll, and the categories information. See the BlogData class for more information.

C#

```
public BlogData GetBlog(int ID, String StartDate, int MaxEntries, int LanguageID);
```

Visual Basic

```
Public Function GetBlog(ByVal ID As Integer, ByVal StartDate As String, ByVal MaxEntries As
Integer, ByVal LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
`ID`	The ID of the blog (see page 454). This is a integer value that corresponds to the folder ID of the blog (see page 454).
`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. If set to 0, the `postsvisible` property of the blog (see page 454) is used. If set to -1, the content (see page 449) is not returned, only the post info for the content (see page 449).
`LanguageID`	The language ID for the content (see page 449) to get. An integer value.

Returns

For a given blog (see page 454) 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 (see page 534) can retrieve blog (see page 454) information by specifying a blog (see page 454) ID, a date and how to handle post information. The sample returns BlogData for the Title, Tagline, Categories, Blog (see page 454) 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 (see page 454) 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 (see page 454).

```
blogDataItem = blogApi.GetBlog(CInt(txtBlogId.Text), txtDate.Text, iPostNum, 1033)
```

Note: To get the blog (see page 454) 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 (see page 454) exists.

```
If (blogDataItem.Id = 0) Then
    lblResult.Text = "Error getting blog"
    Exit Sub
End If
```

9. Display the results for the user (see page 534).

```
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 (see page 534) 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 (see page 449) for a blog (see page 454), 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 (see page 454). This is a string value that corresponds to the folder path of the blog (see page 454).
`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. If set to 0, the `postsvisible` property of the blog (see page 454) is used. If set to -1, the content (see page 449) is not returned, only the post info for the content (see page 449).
`LanguageID`	The language ID for the content (see page 449). An integer value.

Returns

For a given path, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

Remarks

Developers are advised to check for null prior to using.

GetBlogData Method

Overloaded Method to obtain a BlogData Class.

Overload List

	Name	Description
	Blog.GetBlogData (FolderData) (see page 463)	Returns the corresponding blog (see page 454) (if applicable) for a complete folderData object. If there is code that already leverages our FolderData (see page 755) class, this can piggyback atop it. **Returns only if the folder is a blog (see page 454).** This function is overloaded. See Also: • BlogObject(Integer DateTime) • BlogObject(Integer DateTime Integer Integer) • BlogObject(Integer)
	Blog.GetBlogData (Integer) (see page 464)	Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog (see page 454). Returns either a populated BlogData class or null (nothing). This function is overloaded. See Also: • BlogObject(Integer DateTime) • BlogObject(Integer DateTime Integer Integer) • BlogObject(FolderData (see page 755))
	Blog.GetBlogData (Integer, DateTime) (see page 464)	Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog (see page 454), 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(Integer) • BlogObject(Integer DateTime Integer Integer) • BlogObject(FolderData (see page 755))
	Blog.GetBlogData (Integer, DateTime, Integer, Integer) (see page 465)	Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog (see page 454), 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(Integer) • BlogObject(Integer DateTime) • BlogObject(FolderData (see page 755))

Blog.GetBlogData Method (FolderData)

Returns the corresponding blog (see page 454) (if applicable) for a complete folderData object. If there is code that already leverages our FolderData (see page 755) class, this can piggyback atop it. **Returns only if the folder is a blog (see page 454).**

This function is overloaded. See Also:

- BlogObject(Integer DateTime)
- BlogObject(Integer DateTime Integer Integer)
- BlogObject(Integer)

C#

```
public BlogData GetBlogData(FolderData folder_data);
```

Visual Basic

```
Public Function GetBlogData(ByVal folder_data As FolderData) As BlogData
```

Parameters

Parameters	Description
folder_data	A populated FolderData (see page 755) class. (see FolderData (see page 755))

Returns

A populated BlogData class or nothing

Remarks

Blog.GetBlogData Method (Integer)

Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog (see page 454). Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

- BlogObject(Integer DateTime)
- BlogObject(Integer DateTime Integer Integer)
- BlogObject(FolderData (see page 755))

C#

```
public BlogData GetBlogData(int BlogID);
```

Visual Basic

```
Public Function GetBlogData(ByVal BlogID As Integer) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog (see page 454). This corresponds to the ID of the blog (see page 454) folder.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogData Method (Integer, DateTime)

Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog (see page 454), 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(Integer)
- BlogObject(Integer DateTime Integer Integer)
- BlogObject(FolderData (see page 755))

C#

```
public BlogData GetBlogData(int BlogID, DateTime StartDate);
```

Visual Basic

```
Public Function GetBlogData(ByVal BlogID As Integer, ByVal StartDate As DateTime) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog (see page 454). This corresponds to the ID of the blog (see page 454) 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 (Integer, DateTime, Integer, Integer)

Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog (see page 454), 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(Integer)
- BlogObject(Integer DateTime)
- BlogObject(FolderData (see page 755))

C#

```
public BlogData GetBlogData(int BlogID, DateTime StartDate, int Postsvisible, int LanguageID);
```

Visual Basic

```
Public Function GetBlogData(ByVal BlogID As Integer, ByVal StartDate As DateTime, ByVal
Postsvisible As Integer, ByVal LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog (see page 454). This corresponds to the ID of the blog (see page 454) 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 (see page 454) 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 (see page 454) information for a blog (see page 454) post. This contains information about categories, whether comments are enabled, etc. Accepts the post ID.

C#

```
public BlogData GetBlogDataForPost(int post_id);
```

Visual Basic

```
Public Function GetBlogDataForPost(ByVal post_id As Integer) As BlogData
```

Parameters

Parameters	Description
post_id	Integer value of the post ID

Returns

BlogData class

Blog.GetBlogPostData Method

Gets the Post data for a specific blog (see page 454) post. Accepts the ID of the post. Will return tags, trackback, pingback, and categories for the post.

C#

```
public BlogPostData GetBlogPostData(int id);
```

Visual Basic

```
Public Function GetBlogPostData(ByVal id As Integer) As BlogPostData
```

Parameters

Parameters	Description
id	Integer value of the post ID.

Returns

BlogPostData Class.

GetBlogRoll Method

Returns a populated blog roll class. This blog roll is the list of links for the blog. See the BlogRoll class for the properties and usage of the blog roll.

This is an overloaded method.

Overload List

	Name	Description
	Blog.GetBlogRoll (Integer) (see page 466)	Returns a populated blog (see page 454) roll class. Accepts a blog (see page 454) ID integer value. See the BlogRoll (see page 46) class for more information.
	Blog.GetBlogRoll (String) (see page 467)	Loads a blog (see page 454) roll for a given path. The path is a string of the blog (see page 454) path, ex: `\Media\Blogs\MyBlog`. This blog (see page 454) roll is the list of links for the blog (see page 454). See the BlogRoll (see page 46) class for the properties and usage of the blog (see page 454) roll. This is an overloaded method. See Also: GetBlogRoll(Integer).

Blog.GetBlogRoll Method (Integer)

Returns a populated blog (see page 454) roll class. Accepts a blog (see page 454) ID integer value. See the BlogRoll (see page 46) class for more information.

C#

```
public BlogRoll GetBlogRoll(int blog_id);
```

Visual Basic

```
Public Function GetBlogRoll(ByVal blog_id As Integer) As BlogRoll
```

Parameters

Parameters	Description
blog_id	Integer ID of the blog (see page 454). This corresponds to the folder ID of the blog (see page 454).

Returns

A populated BlogRoll (see page 46) 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 (see page 454) roll for a given path. The path is a string of the blog (see page 454) path, ex: `\Media\Blogs\MyBlog`. This blog (see page 454) roll is the list of links for the blog (see page 454). See the BlogRoll (see page 46) class for the properties and usage of the blog (see page 454) roll.

This is an overloaded method. See Also: GetBlogRoll(Integer).

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 (see page 454). This is a string value that corresponds to the folder path of the blog (see page 454).

Returns

This method returns a populated BlogRoll (see page 46) class or null (nothing).

Remarks

When using, developers are advised to check for null prior to using.

Blog.GetPost Method

Gets a blog (see page 454) post for a given ID. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData (see page 738) Class.

C#

```
public ContentData GetPost(int post_id, BlogPostData PostInfo);
```

Visual Basic

```
Public Function GetPost(ByVal post_id As Integer, ByRef PostInfo As BlogPostData) As
ContentData
```

Parameters

Parameters	Description
post_id	Integer ID for the post.
PostInfo	BlogPostData class representing the blog (see page 454) specific post info. This is also in the metadata.

Returns

ContentData (see page 738) class with a ByRef BlogPostData

Blog.GetPostbyID Method

Gets a blog (see page 454) post for a given ID. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class.

C#

```
public BlogPostData GetPostbyID(int post_id);
```

Visual Basic

```
Public Function GetPostbyID(ByVal post_id As Integer) As BlogPostData
```

Parameters

Parameters	Description
post_id	Integer ID for the post.

Returns

BlogPostData class

Content Class

Class Hierarchy

C#

```
public class Content : Ektron.Cms.API.Common;
```

Visual Basic

```
Public Class Content
Inherits Ektron.Cms.API.Common
```

File

Content.vb

Description

This is class API.Content.Content.

Content Methods

The methods of the Content class are listed here.

Public Methods

	Name	Description
	AddContent (see page 470)	Adds Content to the CMS. This method is overloaded with three AddContent methods. The first AddContent (see page 470) method allows you to add content and have the CMS assign it a content ID. The second AddContent (see page 472) method allows you to add content and assign it a content ID. The third AddContent (see page 473) allows you to add content and assign it an XML ID and Template ID, while the CMS assigns it a content ID
	AddXmlConfiguration (see page 474)	Adds an XML configuration for XML oriented content (see page 468).
	CheckOutContent (see page 474)	This will check the content (see page 468) out to the current user (see page 534) if the content (see page 468) is currently checked in and the user (see page 534) has privileges to check it out.

	CopyContentToFolder (see page 474)	Copy the content (see page 468) into required folder.
	DeleteContentItem (see page 475)	Delete the content (see page 468) by the content (see page 468)'s ID. *See Also:* The Example section.
	DeleteXmlConfiguration (see page 476)	Deletes an XML configuration.
	GetAddViewLanguage (see page 476)	Loads all the languages from the system for a given content (see page 468) ID and populates the Add/View drop down box with flag set on each item.
	GetAllComments (see page 477)	Loads all of the comments.
	GetAllTemplates (see page 477)	Loads all of the templates.
	GetAllUnassignedItemApprovals (see page 477)	Returns all of the unassigned user (see page 534)/groups for a given content (see page 468) ID.
	GetAllXmlConfigurations (see page 478)	Reads all of the XML configurations.
	GetChildContent (see page 478)	Loads all of the contents for the given folder.
	GetContent (see page 478)	Returns list of content. This function is overloaded.
	GetContentByHistory (see page 481)	Retrieves the corresponding content (see page 468) by the history ID.
	GetContentDisplayLink (see page 481)	Function takes in the required information about the string then will decide the best link to use to display the content (see page 468). It decides between Linkit, a generated quicklink or just the template file.
	GetContentForEditing (see page 482)	Loads the content (see page 468) details with checkout mode.
	GetContentReport (see page 482)	Loads the content (see page 468) data for the given constraints.
	GetContentState (see page 482)	Loads the content (see page 468)'s details.
	GetContentStatus (see page 483)	This will return a single letter string which is the status of the piece of content (see page 468). The letters returned are the same letters used in the Workarea view.
	GetDomain (see page 483)	Gets web domain for a piece of content (see page 468).
	GetExpireContent (see page 483)	Loads all of the expired content (see page 468).
	MoveContentToFolder (see page 483)	Moves content (see page 468) to a folder.
	PublishContent (see page 484)	Function takes in information about the content (see page 468) to be checked in and published. This goes through the normal approval chain. The content (see page 468) must be in a checked out state to the current logged in user (see page 534) for this to succeed.
	RelicateXmlConfiguration (see page 484)	**This method is obsolete. The recommended alternative is ReplicateXmlConfiguration (see page 485).** Replicate the XML configuration.
	RenameContent (see page 485)	Renames the content (see page 468).
	ReplicateXmlConfiguration (see page 485)	Replicate the XML configuration.
	RestoreHistoryContent (see page 485)	Restores the content (see page 468) to a particular version of content (see page 468).
	SaveContent (see page 486)	Updates the checked out content (see page 468) 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 486)	Delete the content (see page 468) by the content (see page 468)'s Id and folder Id.

	UndoCheckout (see page 487)	This undoes the checkout for a piece of content (see page 468) which is checked out to the current user (see page 534).
	UpdateContentMetaData (see page 487)	Updates a single piece of content (see page 468) metadata
	UpdatexmlCollectionPackage (see page 487)	Updates the XML collection package.
	UpdateXmlConfiguration (see page 487)	Updates the XML configuration details.

Legend

	Method

AddContent Method

Adds Content to the CMS. This method is overloaded with three AddContent methods.

The first AddContent (see page 470) method allows you to add content and have the CMS assign it a content ID.

The second AddContent (see page 472) method allows you to add content and assign it a content ID.

The third AddContent (see page 473) 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, Integer, Object, Object, Object) (see page 470)	Adds Content (see page 468) to the CMS. Upon successful completion, the CMS returns a Content (see page 468) ID. When content (see page 468) is added using this method, Web Alerts are **not** sent. *See Also:* The Example section.
	Content.AddContent (String, String, String, String, String, String, Integer, Object, Object, Object, Integer) (see page 472)	Adds Content (see page 468) to the CMS and allows you to assign it a Content (see page 468) ID. Once the content (see page 468) is added the new ID is returned. When content (see page 468) is added using this method, Web Alerts are **not** sent.
	Content.AddContent (String, String, String, String, String, String, Integer, Object, Object, Object, Integer, Integer) (see page 473)	Adds Content (see page 468) to the CMS and allows you to assign the content (see page 468) an XML ID and Template ID. Once the content (see page 468) is added the new Content (see page 468) ID is returned. When content (see page 468) is added using this method, Web Alerts are **not** sent.

Content.AddContent Method (String, String, String, String, String, String, Integer, Object, Object, Object)

Adds Content (see page 468) to the CMS. Upon successful completion, the CMS returns a Content (see page 468) ID. When content (see page 468) is added using this method, Web Alerts are **not** sent.

See Also: The Example section.

C#

```
public int AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, int FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml);
```

Visual Basic

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Integer, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object) As Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 468) to add.
ContentComment	Comments about the content (see page 468) being added.
ContentHtml	Content (see page 468) HTML to display.
SearchText	Search (see page 514) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank.
SummaryHtml	A summary of the HTML content (see page 468).
ContentLanguage	The language to which the content (see page 468) is assigned.
FolderId	Folder (see page 383) in which to place the content (see page 468).
GoLive	The date the content (see page 468) goes live on the site.
EndDate	The date the content (see page 468) stops appearing on the site.
MetaInfoXml	Meta Info XML

Returns

Integer - ID of newly created content (see page 468).

Example

The following example shows how to create a Web page from which a user (see page 534) can add content (see page 468) 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 (see page 468). 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 (see page 468) API (see page 363).
```
       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 (see page 468) ID of 0 (zero). When the content (see page 468) 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 (see page 534) know whether the content (see page 468) was added and the ID of the content (see page 468).
```
               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 (see page 534) 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, Integer, Object, Object, Object, Integer)

Adds Content (see page 468) to the CMS and allows you to assign it a Content (see page 468) ID. Once the content (see page 468) is added the new ID is returned. When content (see page 468) is added using this method, Web Alerts are **not** sent.

C#

```
public int AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, int FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, int ContentID);
```

Visual Basic

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Integer, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object, ByVal ContentID As Integer) As Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 468) to add.
ContentComment	Comments about the content (see page 468) being added.
ContentHtml	Content (see page 468) HTML to display.
SearchText	Search (see page 514) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank.
SummaryHtml	A summary of the HTML content (see page 468).
ContentLanguage	The language to which the content (see page 468) is assigned.
FolderId	Folder (see page 383) in which to place the content (see page 468).
GoLive	The date the content (see page 468) goes live on the site.
EndDate	The date the content (see page 468) stops appearing on the site.
MetaInfoXml	Meta Info XML
ContentID	The ID you would like to assign the content (see page 468).

Returns

Integer - ID of newly created content (see page 468).

Content.AddContent Method (String, String, String, String, String, String, Integer, Object, Object, Object, Integer, Integer)

Adds Content (see page 468) to the CMS and allows you to assign the content (see page 468) an XML ID and Template ID. Once the content (see page 468) is added the new Content (see page 468) ID is returned. When content (see page 468) is added using this method, Web Alerts are **not** sent.

C#

```
public int AddContent(String ContentTitle, String ContentComment, String ContentHtml, String
SearchText, String SummaryHtml, String ContentLanguage, int FolderId, Object GoLive, Object
EndDate, Object MetaInfoXml, int XmlID, int TemplateID);
```

Visual Basic

```
Public Function AddContent(ByVal ContentTitle As String, ByVal ContentComment As String, ByVal
ContentHtml As String, ByVal SearchText As String, ByVal SummaryHtml As String, ByVal
ContentLanguage As String, ByVal FolderId As Integer, ByVal GoLive As Object, ByVal EndDate As
Object, ByVal MetaInfoXml As Object, ByVal XmlID As Integer, ByVal TemplateID As Integer) As
Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 468) to add.
ContentComment	Comments about the content (see page 468) being added.
ContentHtml	Content (see page 468) HTML to display.
SearchText	Search (see page 514) Text. This parameter allows for upgrades from CMS300. If you did not upgrade from CMS300, leave blank.
SummaryHtml	A summary of the HTML content (see page 468).
ContentLanguage	The language to which the content (see page 468) is assigned.

FolderId	Folder (see page 383) in which to place the content (see page 468).
GoLive	The date the content (see page 468) goes live on the site.
EndDate	The date the content (see page 468) stops appearing on the site.
MetaInfoXml	Meta Info XML
XmlID	XmlID
TemplateID	Template ID

Returns

Integer - ID of newly created content (see page 468).

Content.AddXmlConfiguration Method

Adds an XML configuration for XML oriented content (see page 468).

C#

```
public int AddXmlConfiguration(Collection pagedata);
```

Visual Basic

```
Public Function AddXmlConfiguration(ByVal pagedata As Collection) As Integer
```

Parameters

Parameters	Description
pagedata	XML configuration data

Returns

Integer

Content.CheckOutContent Method

This will check the content (see page 468) out to the current user (see page 534) if the content (see page 468) is currently checked in and the user (see page 534) has privileges to check it out.

C#

```
public Boolean CheckOutContent(int ContentId);
```

Visual Basic

```
Public Function CheckOutContent(ByVal ContentId As Integer) As Boolean
```

Parameters

Parameters	Description
ContentId	Id of the content (see page 468) to check out

Returns

A Boolean representing the success of the check out

Content.CopyContentToFolder Method

Copy the content (see page 468) 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 468) ID
`FID`	Folder (see page 383) ID
`Lang`	

Returns

Ektron.Cms.ContentData (see page 738)

Content.DeleteContentItem Method

Delete the content (see page 468) by the content (see page 468)'s ID.

See Also: The Example section.

C#

```
public Boolean DeleteContentItem(int Id);
```

Visual Basic

```
Public Function DeleteContentItem(ByVal Id As Integer) As Boolean
```

Parameters

Parameters	Description
`Id`	The content (see page 468)'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 (see page 534) can delete content (see page 468) 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 (see page 468). 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 (see page 468) API (see page 363).

```
Dim contentApi As New Ektron.Cms.API.Content.Content
```

3. Check to make sure the content (see page 468) 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 (see page 534) know whether the content (see page 468) 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 (see page 534) 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(int Id);
```

Visual Basic

```
Public Sub DeleteXmlConfiguration(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	XML configuration ID

Content.GetAddViewLanguage Method

Loads all the languages from the system for a given content (see page 468) ID and populates the Add/View drop down box with flag set on each item.

C#

```
public LanguageData[] GetAddViewLanguage(int Id);
```

Visual Basic

```
Public Function GetAddViewLanguage(ByVal Id As Integer) As LanguageData()
```

Parameters

Parameters	Description
Id	Content (see page 468) ID

Returns

Ektron.Cms.LanguageData()

Content.GetAllComments Method

Loads all of the comments.

C#

```
public CommentData[] GetAllComments(int KeyId, int CommentId, int RefId, String RefType, int
UserId, String OrderBy);
```

Visual Basic

```
Public Function GetAllComments(ByVal KeyId As Integer, ByVal CommentId As Integer, ByVal RefId
As Integer, ByVal RefType As String, ByVal UserId As Integer, ByVal OrderBy As String) As
CommentData()
```

Parameters

Parameters	Description
KeyId	key ID
CommentId	Comment ID
RefId	Reference ID, (content (see page 468) or forms)
RefType	Reference type, C = content (see page 468) or T = Task (see page 519)
UserId	The user (see page 534) 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 (see page 534)/groups for a given content (see page 468) ID.

C#

```
public ApprovalData[] GetAllUnassignedItemApprovals(int Id, String ItemType);
```

Visual Basic

```
Public Function GetAllUnassignedItemApprovals(ByVal Id As Integer, ByVal ItemType As String)
As ApprovalData()
```

Parameters

Parameters	Description
Id	The content (see page 468)'s ID
ItemType	The type of user (see page 534)

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()

Content.GetChildContent Method

Loads all of the contents for the given folder.

C#

```
public ContentData[] GetChildContent(int Id, Boolean Recursive, String OrderBy);
```

Visual Basic

```
Public Function GetChildContent(ByVal Id As Integer, Optional ByVal Recursive As Boolean =
False, Optional ByVal OrderBy As String = "name") As ContentData()
```

Parameters

Parameters	Description
Id	the folders ID
Recursive	True = recursive look up. False = none
OrderBy	An order by clause

Returns

Ektron.Cms.ContentData (see page 738)()

GetContent Method

Returns list of content. This function is overloaded.

Overload List

	Name	Description
	Content.GetContent (Ektron.Cms.Common.ContentRequest, System.Web.UI.Page) (see page 479)	Returns list of contents using content (see page 468) request object
	Content.GetContent (Integer, ContentResultType) (see page 479)	Using the content (see page 468)'s ID and the ContentType to get the content (see page 468) and return it as ContentData (see page 738). *See Also:* The Example section.

	Content.GetContent (Integer, Ektron.Cms.Content.EkContent.ContentResultType) (see page 480)	Loads the content (see page 468) details using the content (see page 468) ID

Content.GetContent Method (Ektron.Cms.Common.ContentRequest, System.Web.UI.Page)

Returns list of contents using content (see page 468) request object

C#

```
public Ektron.Cms.Common.ContentResult GetContent(Ektron.Cms.Common.ContentRequest req,
System.Web.UI.Page page);
```

Visual Basic

```
Public Function GetContent(ByVal req As Ektron.Cms.Common.ContentRequest, Optional ByVal page
As System.Web.UI.Page = Nothing) As Ektron.Cms.Common.ContentResult
```

Parameters

Parameters	Description
req	ContentRequest
page	Page reference

Returns

ContentResult

Content.GetContent Method (Integer, ContentResultType)

Using the content (see page 468)'s ID and the ContentType to get the content (see page 468) and return it as ContentData (see page 738).

See Also: The Example section.

C#

```
public ContentData GetContent(int Id, ContentResultType ContentType);
```

Visual Basic

```
Public Function GetContent(ByVal Id As Integer, ByVal ContentType As ContentResultType) As
ContentData
```

Parameters

Parameters	Description
Id	The ID of the content (see page 468) to be returned.
ContentType	Whether the content (see page 468) is published or staged. Enter either: Ektron.Cms.Content.EkContent.ContentResultType.Published or Ektron.Cms.Content.EkContent.ContentResultType.Staged

Returns

ContentData (see page 738)

Example

The following example shows how to create a Web page from which a user (see page 534) can view a content (see page 468) block and its title by entering the content (see page 468) 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 (see page 468). 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 (see page 468) API (see page 363).
```
Dim contentApi As New Ektron.Cms.API.Content.Content
```
3. Check to make sure the content (see page 468) 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 (see page 468) 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 (see page 468) or lets the user (see page 534) know if the content (see page 468) 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 (see page 534) know what happened.
```
Catch ex As Exception
    lblResult.Text = ex.Message
End Try

End Sub
```

Content.GetContent Method (Integer, Ektron.Cms.Content.EkContent.ContentResultType)

Loads the content (see page 468) details using the content (see page 468) ID

C#

```
public ContentData GetContent(int Id, Ektron.Cms.Content.EkContent.ContentResultType Type);
```

Visual Basic

```
Public Function GetContent(ByVal Id As Integer, Optional ByVal Type As
Ektron.Cms.Content.EkContent.ContentResultType = ContentResultType.Published) As ContentData
```

Parameters

Parameters	Description
Id	The content (see page 468) ID
Type	

Returns

Ektron.Cms.ContentData (see page 738)

Content.GetContentByHistory Method

Retrieves the corresponding content (see page 468) by the history ID.

C#

```
public ContentData GetContentByHistory(int Id);
```

Visual Basic

```
Public Function GetContentByHistory(ByVal Id As Integer) As ContentData
```

Parameters

Parameters	Description
Id	The history ID

Returns

Ektron.Cms.ContentData (see page 738)

Content.GetContentDisplayLink Method

Function takes in the required information about the string then will decide the best link to use to display the content (see page 468). It decides between Linkit, a generated quicklink or just the template file.

C#

```
public String GetContentDisplayLink(int ContentType, String ContentId, String ContentLanguage,
String FolderId, String host, String StagingDomain, String ProductionDomain);
```

Visual Basic

```
Public Function GetContentDisplayLink(ByVal ContentType As Integer, ByVal ContentId As String,
ByVal ContentLanguage As String, ByVal FolderId As String, ByVal host As String, ByVal
StagingDomain As String, ByVal ProductionDomain As String) As String
```

Parameters

Parameters	Description
ContentType	Integer describing the content (see page 468)'s type (word doc, multimedia, etc.)
ContentId	A string value of the content (see page 468)'s ID number
ContentLanguage	A string value of the content (see page 468)'s language number
FolderId	A string value of the content (see page 468)'s folder ID number
host	A string value of the host name to build the link out of
StagingDomain	A string value of the content (see page 468)'s staging domain
ProductionDomain	A string value of the content (see page 468)'s production domain

Returns

The page link as a String

Content.GetContentForEditing Method

Loads the content (see page 468) details with checkout mode.

C#

```
public ContentEditData GetContentForEditing(int Id);
```

Visual Basic

```
Public Function GetContentForEditing(ByVal Id As Integer) As ContentEditData
```

Parameters

Parameters	Description
Id	The content (see page 468) ID.

Returns

Ektron.Cms.ContentEditData

Content.GetContentReport Method

Loads the content (see page 468) 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 738)()

Content.GetContentState Method

Loads the content (see page 468)'s details.

C#

```
public ContentStateData GetContentState(int Id);
```

Visual Basic

```
Public Function GetContentState(ByVal Id As Integer) As ContentStateData
```

Parameters

Parameters	Description
Id	The content (see page 468)'s ID

Returns

Ektron.Cms.ContentStateData

Content.GetContentStatus Method

This will return a single letter string which is the status of the piece of content (see page 468). The letters returned are the same letters used in the Workarea view.

C#

```
public String GetContentStatus(int ContentId);
```

Visual Basic

```
Public Function GetContentStatus(ByVal ContentId As Integer) As String
```

Parameters

Parameters	Description
ContentId	Id of the content (see page 468) to return the status of

Returns

A string with the single letter status of the content (see page 468)

Content.GetDomain Method

Gets web domain for a piece of content (see page 468).

C#

```
public String GetDomain(int ContentId);
```

Visual Basic

```
Public Function GetDomain(ByVal ContentId As Integer) As String
```

Parameters

Parameters	Description
ContentId	The content (see page 468)'s ID

Returns

String

Content.GetExpireContent Method

Loads all of the expired content (see page 468).

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 738)()

Content.MoveContentToFolder Method

Moves content (see page 468) to a folder.

C#

```
public ContentData MoveContentToFolder(String cId, String cLanguages, Long FID);
```

Visual Basic

```
Public Function MoveContentToFolder(ByVal cId As String, ByVal cLanguages As String, ByVal FID
As Long) As ContentData
```

Parameters

Parameters	Description
cId	The content (see page 468)'s ID
cLanguages	The content (see page 468)'s language
FID	The folder's ID

Returns

Ektron.Cms.ContentData (see page 738)

Content.PublishContent Method

Function takes in information about the content (see page 468) to be checked in and published. This goes through the normal approval chain. The content (see page 468) must be in a checked out state to the current logged in user (see page 534) for this to succeed.

C#

```
public Boolean PublishContent(int ContentId, int FolderId, int ContentLanguage, String
dontCreateTask, int UserID, String TaskTitle);
```

Visual Basic

```
Public Function PublishContent(ByVal ContentId As Integer, ByVal FolderId As Integer, ByVal
ContentLanguage As Integer, ByVal dontCreateTask As String, ByVal UserID As Integer, ByVal
TaskTitle As String) As Boolean
```

Parameters

Parameters	Description
ContentId	Integer value of the content (see page 468)'s ID
FolderId	Integer value of the content (see page 468) folder's ID
ContentLanguage	Integer value of the content (see page 468)'s language ID
dontCreateTask	If this parameter is "" a task (see page 519) will be created for this submit, else it will not
UserID	The user (see page 534) ID of task (see page 519) assigned by (current user (see page 534)). This is needed only if "dontCreateTask" is "", it is ignored else
TaskTitle	Title for the task (see page 519). This is needed only if "dontCreateTask" is "", it is ignored else

Returns

A Boolean indicating success or not.

Content.RelicateXmlConfiguration Method

This method is obsolete. The recommended alternative is ReplicateXmlConfiguration (see page 485).

Replicate the XML configuration.

C#

```
[Obsolete("The recommended alternative is ReplicateXmlConfiguration(ByVal Id As Integer, ByVal
Title As String)")]
public int RelicateXmlConfiguration(int Id, String Title);
```

Visual Basic

```
<Obsolete("The recommended alternative is ReplicateXmlConfiguration(ByVal Id As Integer, ByVal
```

```
Title As String)")>
Public Function RelicateXmlConfiguration(ByVal Id As Integer, ByVal Title As String) As Integer
```

Parameters

Parameters	Description
Id	The inherited XML ID
Title	Title

Returns

Integer

Content.RenameContent Method

Renames the content (see page 468).

C#

```
public ContentData RenameContent(int cId, String title);
```

Visual Basic

```
Public Function RenameContent(ByVal cId As Integer, ByVal title As String) As ContentData
```

Parameters

Parameters	Description
cId	The content (see page 468)'s ID
title	The content (see page 468)'s new title

Returns

Boolean

Content.ReplicateXmlConfiguration Method

Replicate the XML configuration.

C#

```
public int ReplicateXmlConfiguration(int Id, String Title);
```

Visual Basic

```
Public Function ReplicateXmlConfiguration(ByVal Id As Integer, ByVal Title As String) As
Integer
```

Parameters

Parameters	Description
Id	The inherited XML ID
Title	Title

Returns

Integer

Content.RestoreHistoryContent Method

Restores the content (see page 468) to a particular version of content (see page 468).

C#

```
public RestoreHistoryContent(int Id);
```

Visual Basic

```
Public Sub RestoreHistoryContent(ByVal Id As Integer)
```

Parameters

Parameters	Description
`Id`	The content (see page 468)'s history ID

Content.SaveContent Method

Updates the checked out content (see page 468) 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 468) must be in the checked out state.

Content.SubmitForDelete Method

Delete the content (see page 468) by the content (see page 468)'s Id and folder Id.

C#

```
public Boolean SubmitForDelete(int cId, int fId);
```

Visual Basic

```
Public Function SubmitForDelete(ByVal cId As Integer, ByVal fId As Integer) As Boolean
```

Parameters

Parameters	Description
`cId`	The content (see page 468)'s Id
`fId`	The content (see page 468)'s Folder (see page 383) Id

Returns

Boolean

Content.UndoCheckout Method

This undoes the checkout for a piece of content (see page 468) which is checked out to the current user (see page 534).

C#

```
public Boolean UndoCheckout(int ContentId);
```

Visual Basic

```
Public Function UndoCheckout(ByVal ContentId As Integer) As Boolean
```

Parameters

Parameters	Description
ContentId	Id of the content (see page 468) for which to undo check out

Returns

A Boolean representing the success of the undo.

Content.UpdateContentMetaData Method

Updates a single piece of content (see page 468) metadata

C#

```
public Boolean UpdateContentMetaData(int ContentID, int MetaID, String MetaText);
```

Visual Basic

```
Public Function UpdateContentMetaData(ByVal ContentID As Integer, ByVal MetaID As Integer,
ByVal MetaText As String) As Boolean
```

Parameters

Parameters	Description
ContentID	The Content (see page 468) ID
MetaID	The Metadata (see page 412) 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 API.Content.ContentRating.

ContentRating Methods

The methods of the ContentRating class are listed here.

Public Methods

	Name	Description
	GetContentRating (see page 488)	Constructs a string that represents a content (see page 449) rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content (see page 449) rating server control wraps this functionality.
	GetContentRatingAddRating (see page 490)	Adds a content (see page 449) rating to the database. Internally a check is done to ensure the user (see page 534) has not already rated the content (see page 449), and passes down the rating and any comments the user (see page 534) supplied.
	GetContentRatingResults (see page 491)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 449) 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 491)	Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate. This method is overloaded.
	PurgeContentRatings (see page 492)	Purges all ratings of a particular content (see page 449) block over a defined date/time range.

Legend

	Method

ContentRating.GetContentRating Method

Constructs a string that represents a content (see page 449) rating element on a web site. This is composed of a rating dialog and a graph representing current ratings. Note that the content (see page 449) rating server control wraps this functionality.

C#

```
public String GetContentRating(int contentID, System.Web.UI.Page currentPage, String key,
ContentRatingLabels Labels, ContentAPI.RatingGraphDisplayType initialGraph,
```

```
ContentAPI.RatingGraphDisplayType ratingCompleteGraph, int padding, System.Drawing.Color
bgColor, System.Drawing.Color barColor, System.Drawing.Color TextColor);
```

Visual Basic

```
Public Function GetContentRating(ByVal contentID As Integer, ByVal currentPage As
System.Web.UI.Page, ByVal key As String, ByVal Labels As ContentRatingLabels, ByVal
initialGraph As ContentAPI.RatingGraphDisplayType, ByVal ratingCompleteGraph As
ContentAPI.RatingGraphDisplayType, ByVal padding As Integer, ByVal bgColor As
System.Drawing.Color, ByVal barColor As System.Drawing.Color, ByVal TextColor As
System.Drawing.Color) As String
```

Parameters

Parameters	Description
contentID	The numeric ID associated with the content (see page 449) block being rated.
currentPage	A reference to the System.Web.UI.Page object representing the active web page.
key	Specifies a unique ID of this element. No two content (see page 449) rating controls on the same page should specify the same key.
Labels	Specifies the labels of the control to be displayed. See Ektron.Cms.ContentRatingLabels
initialGraph	Specifies to show the initial (non-rated) graph to be horizontal to the control, below it, or not at all.
ratingCompleteGraph	Specifies to show the complete (rated) graph to be horizontal to the control, below it, or not at all.
padding	Specifies the number of pixels between the rating control and the graph.
bgColor	Specifies the background color of the graph.
barColor	Specifies the bar color of the graph.
TextColor	Specifies the text color of the graph.

Returns

Returns a string representing the final control.

Remarks

Note that the ContentRating (see page 488) server control nicely wraps this function.

Example

The following example shows how to create a Web page that retrieves a content (see page 449) rating element used to rate content (see page 449). 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 (see page 449). 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 (see page 449) 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 (see page 449) 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 (see page 534) know what happened.

```
            Catch ex As Exception
                lblResult.Text = ex.Message
            End Try

        End Sub
```

ContentRating.GetContentRatingAddRating Method

Adds a content (see page 449) rating to the database. Internally a check is done to ensure the user (see page 534) has not already rated the content (see page 449), and passes down the rating and any comments the user (see page 534) supplied.

C#

```
public GetContentRatingAddRating(int contentid, int userid, int userRating, String
userComments);
```

Visual Basic

```
Public Sub GetContentRatingAddRating(ByVal contentid As Integer, ByVal userid As Integer,
```

```
ByVal userRating As Integer, ByVal userComments As String)
```

Parameters

Parameters	Description
contentid	The numeric ID associated with the content (see page 449) block being rated.
userid	The unique global identifier stored as a cookie by the user (see page 534).
userRating	The rating the user (see page 534) supplies (1-9).
userComments	Optional comments that the user (see page 534) provides.

ContentRating.GetContentRatingResults Method

Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 449) from the range of StartDate to EndDate. In order to get the full range of ratings, pass DateTime.Min and DateTime.Max as arguments.

C#

```
public System.Data.DataSet GetContentRatingResults(int contentID, DateTime startDate, DateTime
endDate);
```

Visual Basic

```
Public Function GetContentRatingResults(ByVal contentID As Integer, ByVal startDate As
DateTime, ByVal endDate As DateTime) As System.Data.DataSet
```

Parameters

Parameters	Description
contentID	The numeric ID associated with the content (see page 449) block whose ratings are being queried.
startDate	The start date of a range of dates to query over.
endDate	The end date of a range of dates to query over.

Returns

(Username - String, RatingDate - DateTime, Rating - int, Comments - String)

GetContentRatingStatistics Method

Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate.

This method is overloaded.

Overload List

	Name	Description
	ContentRating.GetContentRatingStatistics (Integer, DateTime, DateTime) (see page 491)	Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate. This function is overloaded. See Also: GetContentRatingStatistics(Integer Integer String)
	ContentRating.GetContentRatingStatistics (Integer, Integer, String) (see page 492)	Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate. This function is overloaded. See Also: GetContentRatingStatistics(Integer DateTime DateTime)

ContentRating.GetContentRatingStatistics Method (Integer, DateTime, DateTime)

Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are

bounded by startDate and endDate.

This function is overloaded. See Also: GetContentRatingStatistics(Integer Integer String)

C#

```
public Collection GetContentRatingStatistics(int contentID, DateTime startDate, DateTime
endDate);
```

Visual Basic

```
Public Function GetContentRatingStatistics(ByVal contentID As Integer, ByVal startDate As
DateTime, ByVal endDate As DateTime) As Collection
```

Parameters

Parameters	Description
contentID	The numeric ID associated with the content (see page 449) block being rated.
startDate	Bounds the result set by only ratings that occurred after this date.
endDate	Bounds the result set by only ratings that occurred before this date.

Returns

Returns a collection of data with keys "total" => number of ratings, "sum" => sum of all ratings, "r1, "r2", "r3", "r4", "r5", "r6", "r7", "r8", "r9".

Remarks

Note that average rating = sum/total.

ContentRating.GetContentRatingStatistics Method (Integer, Integer, String)

Returns a collection with keys as ordinal numbers and frequency as the value. That is, {1=>3,2=>7,...,9=>12}. Rating results are bounded by startDate and endDate.

This function is overloaded. See Also: GetContentRatingStatistics(Integer DateTime DateTime)

C#

```
public Collection GetContentRatingStatistics(int contentid, int user_id, String visitor_id);
```

Visual Basic

```
Public Function GetContentRatingStatistics(ByVal contentid As Integer, ByVal user_id As
Integer, ByVal visitor_id As String) As Collection
```

Parameters

Parameters	Description
contentid	The numeric ID associated with the content (see page 449) block being rated.

Returns

Returns a collection of data with keys "total" => number of ratings, "sum" => sum of all ratings, "r1, "r2", "r3", "r4", "r5", "r6", "r7", "r8", "r9".

Remarks

Note that average rating = sum/total.

ContentRating.PurgeContentRatings Method

Purges all ratings of a particular content (see page 449) block over a defined date/time range.

C#

```
public PurgeContentRatings(int contentid, DateTime startDate, DateTime endDate);
```

Visual Basic

```
Public Sub PurgeContentRatings(ByVal contentid As Integer, ByVal startDate As DateTime, ByVal
endDate As DateTime)
```

Parameters

Parameters	Description
contentid	The ID of the content (see page 449) that is having its ratings purged.
startDate	The beginning of the date range of content (see page 449) to be purged.
endDate	The end of the date range of content (see page 449) to be purged.

Form Class

Class Hierarchy

C#

```
public class Form : Content;
```

Visual Basic

```
Public Class Form
Inherits Content
```

File

Form.vb

Description

This is class API.Content.Form.

Form Methods

The methods of the Form class are listed here.

Public Methods

	Name	Description
	AssignFormTask (see page 494)	Assigns a form (see page 493) submission task (see page 519), as specified in FormInfo (see page 351), of the SubmittedDataXml.
	EmailFormFieldData (see page 494)	Sends an email, as specified in FormInfo (see page 351), of the SubmittedDataXml.
	GetAllForms (see page 494)	Retrieves an array of all the forms contained with the specified folder. This method is overloaded.
	GetForm (see page 495)	Retrieves information about the specified form (see page 493).
	GetFormFieldData (see page 496)	Retrieves the submitted data for all fields from a specified form. This method is overloaded.
	GetFormFieldList (see page 497)	Retrieves the list of form (see page 493) fields defined in the specified form (see page 493).
	GetFormFieldQuestions (see page 497)	Retrieves the table of form (see page 493) fields and their questions in a specified form (see page 493). In a CMS form (see page 493), the form (see page 493) question is defined by the user (see page 534) in the descriptive name field in the form (see page 493) element dialog during design time.
	GetFormTitle (see page 497)	Same as GetFormById, but returns only the form (see page 493)'s title.
	SubmitFormFieldData (see page 498)	Adds form (see page 493) data as if it was submitted during a browser post to the web server.
	UpdateFormFieldData (see page 498)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form (see page 493).

Legend

	Method

Form.AssignFormTask Method

Assigns a form (see page 493) submission task (see page 519), as specified in FormInfo (see page 351), 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 (see page 493) associated with the data. This information may be retrieved by calling GetFormById. Value Type: Ektron.Cms.FormData (see page 781)
SubmittedDataXml	The form (see page 493) data submitted in XML format. Value Type: System.String

Returns

Returns True if the task (see page 519) was successfully assigned. Value Type: System.Boolean

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 498) if SendNotification is True and the form (see page 493) is configured to assign a task (see page 519) when data is submitted. The SubmittedDataXml is typically retrieved by calling SubmitFormFieldData (see page 498).

Form.EmailFormFieldData Method

Sends an email, as specified in FormInfo (see page 351), 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 (see page 493) associated with the data. This information may be retrieved by calling GetFormById. Value Type: Ektron.Cms.FormData (see page 781)
SubmittedDataXml	The form (see page 493) data submitted in XML format. Value Type: System.String

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 498) if SendNotification is True and the form (see page 493) is configured to send an email notification when data is submitted. The SubmittedDataXml is typically retrieved by calling SubmitFormFieldData (see page 498).

GetAllForms Method

Retrieves an array of all the forms contained with the specified folder. This method is overloaded.

Overload List

	Name	Description
	Form.GetAllForms () (see page 495)	Returns all of the forms as an array of FormData (see page 781).

	Form.GetAllForms (Integer, String, Integer, Cms.ContentAPI.FormResultType) (see page 495)	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 781).

C#

```
public FormData[] GetAllForms();
```

Visual Basic

```
Public Function GetAllForms() As FormData()
```

Returns

Ektron.Cms.FormData (see page 781)()

Form.GetAllForms Method (Integer, String, Integer, Cms.ContentAPI.FormResultType)

Retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type.

C#

```
public FormData[] GetAllForms(int Id, String OrderBy, int LanguageId,
Cms.ContentAPI.FormResultType Type);
```

Visual Basic

```
Public Function GetAllForms(ByVal Id As Integer, ByVal OrderBy As String, ByVal LanguageId As
Integer, ByVal Type As Cms.ContentAPI.FormResultType) As FormData()
```

Parameters

Parameters	Description
Id	Specifies the folder.
OrderBy	Specifies how to sort the forms
LanguageId	Specifies a language or EkConstants.g_AllContentLanguages (-1) for all languages.
Type	Specifies whether only active forms or archived forms should be returned.

Form.GetForm Method

Retrieves information about the specified form (see page 493).

C#

```
public FormData GetForm(int Id);
```

Visual Basic

```
Public Function GetForm(ByVal Id As Integer) As FormData
```

Parameters

Parameters	Description
Id	Specifies the form (see page 493).

Returns

Returns information about the form (see page 493).

GetFormFieldData Method

Retrieves the submitted data for all fields from a specified form. This method is overloaded.

Overload List

	Name	Description
	Form.GetFormFieldData (Integer, Integer) (see page 496)	Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form (see page 493).
	Form.GetFormFieldData (Integer, String, String) (see page 496)	Retrieves the submitted data for all fields from a specified form (see page 493). All the data may be retrieved or just data submitted within a given date range.

Form.GetFormFieldData Method (Integer, Integer)

Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form (see page 493).

C#

```
public FormSubmittedData GetFormFieldData(int FormId, int FormDataId);
```

Visual Basic

```
Public Function GetFormFieldData(ByVal FormId As Integer, ByVal FormDataId As Integer) As
FormSubmittedData
```

Parameters

Parameters	Description
FormId	Specifies the form (see page 493). 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 799)

Form.GetFormFieldData Method (Integer, String, String)

Retrieves the submitted data for all fields from a specified form (see page 493). All the data may be retrieved or just data submitted within a given date range.

C#

```
public FormSubmittedData[] GetFormFieldData(int FormId, String StartDate, String EndDate);
```

Visual Basic

```
Public Function GetFormFieldData(ByVal FormId As Integer, Optional ByVal StartDate As String =
"", Optional ByVal EndDate As String = "") As FormSubmittedData()
```

Parameters

Parameters	Description
FormId	Specifies the form (see page 493). 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 799)

Form.GetFormFieldList Method

Retrieves the list of form (see page 493) fields defined in the specified form (see page 493).

C#

```
public FormFieldList GetFormFieldList(int FormId);
```

Visual Basic

```
Public Function GetFormFieldList(ByVal FormId As Integer) As FormFieldList
```

Parameters

Parameters	Description
FormId	Specifies the form (see page 493). Value Type: System.Int32

Returns

The list for defined fields in the given form (see page 493). Ektron.Cms.FormFieldList (see page 794)

Form.GetFormFieldQuestions Method

Retrieves the table of form (see page 493) fields and their questions in a specified form (see page 493). In a CMS form (see page 493), the form (see page 493) question is defined by the user (see page 534) in the descriptive name field in the form (see page 493) element dialog during design time.

C#

```
public Hashtable GetFormFieldQuestions(int FormId);
```

Visual Basic

```
Public Function GetFormFieldQuestions(ByVal FormId As Integer) As Hashtable
```

Parameters

Parameters	Description
FormId	Specifies the form (see page 493). Value Type: System.Int32

Returns

A table of defined fields with the corresponding questions in a given form (see page 493).

Remarks

These questions are mainly used for reporting.

Form.GetFormTitle Method

Same as GetFormById, but returns only the form (see page 493)'s title.

C#

```
public String GetFormTitle(int Id);
```

Visual Basic

```
Public Function GetFormTitle(ByVal Id As Integer) As String
```

Parameters

Parameters	Description
Id	Specifies the form (see page 493).

Returns

The title of the specified form (see page 493).

Form.SubmitFormFieldData Method

Adds form (see page 493) data as if it was submitted during a browser post to the web server.

C#

```
public SubmitFormFieldData(int FormId, FormSubmittedData Data, Boolean SendNotification,
[Out()] String SubmittedDataXml);
```

Visual Basic

```
Public Sub SubmitFormFieldData(ByVal FormId As Integer, ByVal Data As FormSubmittedData,
Optional ByVal SendNotification As Boolean = True, <Out()> Optional ByRef SubmittedDataXml As
String = "")
```

Parameters

Parameters	Description
FormId	Specifies the form (see page 493) that corresponds to the data. Value Type: System.Int32
Data	The data to submit. Value Type: Ektron.Cms. FormSubmittedData (see page 799)
SendNotification	(Optional) Specifies whether to send notifications that a form (see page 493) submission has occurred or not. For example, whether to send an email or assign a form (see page 493) submission task (see page 519) as configured for the form (see page 493). 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 (see page 493).

C#

```
public UpdateFormFieldData(int FormId, int FormDataId, FormSubmittedData SubmittedData);
```

Visual Basic

```
Public Sub UpdateFormFieldData(ByVal FormId As Integer, ByVal FormDataId As Integer, ByVal
SubmittedData As FormSubmittedData)
```

Parameters

Parameters	Description
FormId	Specifies the form (see page 493). 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 799)

Taxonomy Class

Class Hierarchy

C#

```
public class Taxonomy : Content;
```

Visual Basic

```
Public Class Taxonomy
Inherits Content
```

File

Taxonomy.vb

Description

This is class API.Content.Taxonomy.

Taxonomy Methods

The methods of the Taxonomy class are listed here.

Public Methods

	Name	Description
	AddTaxonomyItem (see page 499)	
	AddTaxonomySyncFolder (see page 500)	Adds a folder entry into taxonomy (see page 498).
	CreateTaxonomy (see page 500)	Creates new taxonomy (see page 498).
	DeleteTaxonomy (see page 500)	Deletes the existing taxonomy (see page 498).
	GetTaxonomyIdByPath (see page 501)	Returns the taxonomy (see page 498) ID by a given path.
	ImportTaxonomy (see page 501)	Creates a taxonomy (see page 498) using Xml file.
	LoadTaxonomy (see page 501)	Returns the TaxonomyData with all of its subcategory and its items.
	ReadAllAssignedCategory (see page 502)	Returns all the assigned categories for the content (see page 449) ID.
	ReadTaxonomy (see page 502)	Read the taxonomy (see page 498) data.
	RemoveTaxonomyFolder (see page 502)	Removes the sync up folder from taxonomy (see page 498).
	RemoveTaxonomyItem (see page 502)	Removes an item from the taxonomy (see page 498).
	ReOrderTaxonomyItems (see page 503)	Reorder the display of the taxonomy (see page 498) items.
	UpdateTaxonomy (see page 503)	Updates the existing taxonomy (see page 498).

Legend

	Method

AddTaxonomyItem Method

Overload List

	Name	Description
	Taxonomy.AddTaxonomyItem (TaxonomyContentRequest) (see page 499)	Inserts an item into taxonomy (see page 498).
	Taxonomy.AddTaxonomyItem (TaxonomyRequest) (see page 500)	Inserts an item into taxonomy (see page 498).

Taxonomy.AddTaxonomyItem Method (TaxonomyContentRequest)

Inserts an item into taxonomy (see page 498).

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 taxonomy (see page 498).

C#

```
public AddTaxonomyItem(TaxonomyRequest request);
```

Visual Basic

```
Public Sub AddTaxonomyItem(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
`request`	taxonomyrequest

Taxonomy.AddTaxonomySyncFolder Method

Adds a folder entry into taxonomy (see page 498).

C#

```
public AddTaxonomySyncFolder(TaxonomySyncRequest request);
```

Visual Basic

```
Public Sub AddTaxonomySyncFolder(ByVal request As TaxonomySyncRequest)
```

Parameters

Parameters	Description
`request`	taxonomyrequest

Taxonomy.CreateTaxonomy Method

Creates new taxonomy (see page 498).

C#

```
public int CreateTaxonomy(TaxonomyData request);
```

Visual Basic

```
Public Function CreateTaxonomy(ByVal request As TaxonomyData) As Integer
```

Parameters

Parameters	Description
`request`	Taxonomy (see page 498) Data

Returns

New Taxonomy (see page 498) ID

Taxonomy.DeleteTaxonomy Method

Deletes the existing taxonomy (see page 498).

C#

```
public DeleteTaxonomy(TaxonomyRequest request);
```

Visual Basic

```
Public Sub DeleteTaxonomy(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
request	taxonomy (see page 498) request

Taxonomy.GetTaxonomyIdByPath Method

Returns the taxonomy (see page 498) ID by a given path.

C#

```
public int GetTaxonomyIdByPath(String path);
```

Visual Basic

```
Public Function GetTaxonomyIdByPath(ByVal path As String) As Integer
```

Parameters

Parameters	Description
path	path

Returns

Taxonomy (see page 498) ID

Taxonomy.ImportTaxonomy Method

Creates a taxonomy (see page 498) using Xml file.

C#

```
public int ImportTaxonomy(String TaxonomyXml, String Title);
```

Visual Basic

```
Public Function ImportTaxonomy(ByVal TaxonomyXml As String, ByVal Title As String) As Integer
```

Parameters

Parameters	Description
TaxonomyXml	Taxonomy (see page 498) XML
Title	title for the taxonomy (see page 498)

Returns

A Created Taxonomy (see page 498) ID

Taxonomy.LoadTaxonomy Method

Returns the TaxonomyData with all of its subcategory and its items.

C#

```
public TaxonomyData LoadTaxonomy(TaxonomyRequest request);
```

Visual Basic

```
Public Function LoadTaxonomy(ByRef request As TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
request	TaxonomyRequest

Returns

TaxonomyData

Taxonomy.ReadAllAssignedCategory Method

Returns all the assigned categories for the content (see page 449) ID.

C#

```
public TaxonomyBaseData[] ReadAllAssignedCategory(int contentid);
```

Visual Basic

```
Public Function ReadAllAssignedCategory(ByVal contentid As Integer) As TaxonomyBaseData()
```

Parameters

Parameters	Description
contentid	contentid

Returns

array of taxonomybasedata

Taxonomy.ReadTaxonomy Method

Read the taxonomy (see page 498) data.

C#

```
public TaxonomyData ReadTaxonomy(TaxonomyRequest request);
```

Visual Basic

```
Public Function ReadTaxonomy(ByRef request As TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
request	taxonomyrequest

Returns

taxonomydata

Taxonomy.RemoveTaxonomyFolder Method

Removes the sync up folder from taxonomy (see page 498).

C#

```
public RemoveTaxonomyFolder(TaxonomySyncRequest request);
```

Visual Basic

```
Public Sub RemoveTaxonomyFolder(ByVal request As TaxonomySyncRequest)
```

Parameters

Parameters	Description
request	taxonomyrequest

Taxonomy.RemoveTaxonomyItem Method

Removes an item from the taxonomy (see page 498).

C#

```
public RemoveTaxonomyItem(TaxonomyRequest request);
```

Visual Basic

```
Public Sub RemoveTaxonomyItem(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
request	taxonomyrequest

Taxonomy.ReOrderTaxonomyItems Method

Reorder the display of the taxonomy (see page 498) items.

C#

```
public ReOrderTaxonomyItems(TaxonomyRequest request);
```

Visual Basic

```
Public Sub ReOrderTaxonomyItems(ByVal request As TaxonomyRequest)
```

Parameters

Parameters	Description
request	

Taxonomy.UpdateTaxonomy Method

Updates the existing taxonomy (see page 498).

C#

```
public UpdateTaxonomy(TaxonomyData request);
```

Visual Basic

```
Public Sub UpdateTaxonomy(ByVal request As TaxonomyData)
```

Parameters

Parameters	Description
request	taxonomydata

ThreadedDiscussion Class

Class Hierarchy

C#

```
public class ThreadedDiscussion : Content;
```

Visual Basic

```
Public Class ThreadedDiscussion
Inherits Content
```

File

ThreadedDiscussion.vb

Description

This is class API.Content.ThreadedDiscussion.

ThreadedDiscussion Methods

The methods of the ThreadedDiscussion class are listed here.

Public Methods

	Name	Description
	AddBoard (see page 504)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.
	AddForum (see page 505)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum. *See Also:* The Example section.
	AddTopic (see page 507)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.
	DeleteBoard (see page 508)	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 508)	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 508)	Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value.
	GetBoardCategories (see page 509)	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 509)	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 509)	Method to get a discussion board by ID or path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.
	GetDiscussionBoardString (see page 510)	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 511)	Method to obtain a particular forum by ID. Returns a discussion board with a 1 length Discussion Forum (see page 167) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned. *See Also:* The Example section.
	GetRepliesForTopics (see page 512)	Method returns a list of replies for a given forum ID. • ArrayList Structure ArrayListItem(j)(0) • content_id ArrayListItem(j)(1) • comment_total ArrayListItem(j)(2) • comment_sum ArrayListItem(j)(3) • last_posted ArrayListItem(j)(4) • content_language
	GetTopic (see page 513)	Method to obtain a particular Topic by ID. Returns a discussion board with a 1 length Discussion Forum (see page 167) 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 513)	Method to obtain an array of topic replies matching search (see page 514) 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.

Legend

	Method

ThreadedDiscussion.AddBoard Method

Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.

C#

```
public int AddBoard(int ParentID, String Name, String Title, Boolean RequiresAuthentication,
String StyleSheet, String Categories, ));
```

Visual Basic

```
Public Function AddBoard(ByVal ParentID As Integer, ByVal Name As String, ByVal Title As String, ByVal RequiresAuthentication As Boolean, ByVal StyleSheet As String, ByVal Categories As String,  As )) As Integer
```

Parameters

Parameters	Description
ParentID	Integer ID of folder to add discussion board to.
Name	Name of board.
Title	Title of board.
RequiresAuthentication	Does the Board require authentication?
StyleSheet	The stylesheet to apply to the board.
Categories	String array of categories for the board.

Returns

Integer ID of new discussion board.

ThreadedDiscussion.AddForum Method

Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum.

See Also: The Example section.

C#

```
public int AddForum(int BoardID, String Name, String Description, Boolean ModeratePosts, Boolean LockForum, int SortOrder, int CategoryID);
```

Visual Basic

```
Public Function AddForum(ByVal BoardID As Integer, ByVal Name As String, ByVal Description As String, ByVal ModeratePosts As Boolean, ByVal LockForum As Boolean, ByVal SortOrder As Integer, ByVal CategoryID As Integer) As Integer
```

Parameters

Parameters	Description
BoardID	The integer 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? **True** - prevent posting to the forum. Only members of the Administrators user (see page 534) group can add, edit or delete posts. **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	Integer of the Category ID to which the forum belongs.

Returns

Integer ID of new discussion forum.

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 Integer
```

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 (see page 534) know what happened.

```
        Catch ex As Exception
            lblResult.Text = ex.Message
        End Try

    End Sub
```

AddTopic Method

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.

Overload List

	Name	Description
	ThreadedDiscussion.AddTopic (Integer, String, String) (see page 507)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer.
	ThreadedDiscussion.AddTopic (Integer, String, String, Integer) (see page 508)	Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as integer.

ThreadedDiscussion.AddTopic Method (Integer, String, String)

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer.

C#

```
public int AddTopic(int ForumID, String Title, String Message);
```

Visual Basic

```
Public Function AddTopic(ByVal ForumID As Integer, ByVal Title As String, ByVal Message As
String) As Integer
```

Parameters

Parameters	Description
ForumID	Integer value of forum to which the topic is added.
Title	String value of the topic title.
Message	String value of the topic message.

Returns

Integer value for the topic ID.

ThreadedDiscussion.AddTopic Method (Integer, String, String, Integer)

Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as integer.

C#

```
public int AddTopic(int ForumID, String Title, String Message, int LanguageId);
```

Visual Basic

```
Public Function AddTopic(ByVal ForumID As Integer, ByVal Title As String, ByVal Message As
String, ByVal LanguageId As Integer) As Integer
```

Parameters

Parameters	Description
ForumID	Integer value of forum to which the topic is added.
Title	String value of the topic title.
Message	String value of the topic message.
LanguageId	Integer id of the Language ID

Returns

Integer value for the topic ID.

ThreadedDiscussion.DeleteBoard Method

Deletes a board with a given ID. This deletes the board as well as all forums, topics and posts under it. BoardID must be greater than 0.

C#

```
public DeleteBoard(int BoardID);
```

Visual Basic

```
Public Sub DeleteBoard(ByVal BoardID As Integer)
```

Parameters

Parameters	Description
BoardID	The integer ID of the board. This corresponds to the folder ID.

ThreadedDiscussion.DeleteForum Method

Deletes a forum with a given ID. This deletes the forum as well as all topics and posts under it. forumID must be greater than 0 (zero).

C#

```
public DeleteForum(int forumID);
```

Visual Basic

```
Public Sub DeleteForum(ByVal forumID As Integer)
```

Parameters

Parameters	Description
forumID	The integer ID of the forum. This corresponds to the folder ID.

ThreadedDiscussion.DeleteTopic Method

Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value.

C#

```
public Boolean DeleteTopic(int topicid);
```

Visual Basic

```
Public Function DeleteTopic(ByVal topicid As Integer) As Boolean
```

Parameters

Parameters	Description
topicid	Integer value of the topic to delete.

Returns

Boolean

ThreadedDiscussion.GetBoardCategories Method

Method to get the Categories for a board by ID Returns either zero length or populated array. See DiscussionCategory class for more information.

C#

```
public DiscussionCategory[] GetBoardCategories(int boardID);
```

Visual Basic

```
Public Function GetBoardCategories(ByVal boardID As Integer) As DiscussionCategory()
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns an array of type DiscussionCategory.

ThreadedDiscussion.GetBoardForums Method

Method to get the Forums for a board by ID Returns either zero length or populated array. See DiscussionForum class for more information.

C#

```
public DiscussionForum[] GetBoardForums(int boardID);
```

Visual Basic

```
Public Function GetBoardForums(ByVal boardID As Integer) As DiscussionForum()
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns an array of type DiscussionForum.

GetDiscussionBoard Method

Method to get a discussion board by ID or path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

Overload List

	Name	Description
	ThreadedDiscussion.GetDiscussionBoard (Integer) (see page 510)	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 510)	Mthod 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 (Integer)

Method to get a board by ID. Returns a DiscussionBoard class for a given ID value. See DiscussionBoard class for more information on the properties of the returned object.

C#

```
public DiscussionBoard GetDiscussionBoard(int boardID);
```

Visual Basic

```
Public Function GetDiscussionBoard(ByVal boardID As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns a class of type DiscussionBoard.

ThreadedDiscussion.GetDiscussionBoard Method (String)

Mthod 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(int boardID, String URLpath);
```

Visual Basic

```
Public Function GetDiscussionBoardString(ByVal boardID As Integer, ByVal URLpath As String) As String
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.
URLpath	String value of the URL path used. This is generally the Server Variable "URL". Ex: "/forum.aspx".

Returns

Returns a string of the rendered Discussion Board, suitable for the server control or a Web service.

ThreadedDiscussion.GetForum Method

Method to obtain a particular forum by ID. Returns a discussion board with a 1 length Discussion Forum (see page 167) 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(int forumid);
```

Visual Basic

```
Public Function GetForum(ByVal forumid As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
forumid	Integer value of the forum ID.

Returns

An instance of type DiscussionBoard.

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 167) 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 (see page 534) 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(int ForumID);
```

Visual Basic

```
Public Function GetRepliesForTopics(ByVal ForumID As Integer) As ArrayList
```

Parameters

Parameters	Description
ForumID	Integer value of the forum ID.

Returns

An instance of type ArrayList.

Remarks

This is an older method and is deprecated. The information returned herein was merged into the higher performance GetForum (see page 511)(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 167) array containing a 1 length DiscussionTopic array. If the topic does not exist, then a DiscussionBoard with a 1 length Forums property containing a zero length Topics property will be returned.

C#

```
public DiscussionBoard GetTopic(int topicid);
```

Visual Basic

```
Public Function GetTopic(ByVal topicid As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
topicid	Integer value of the Topic ID.

Returns

An instance of type DiscussionBoard.

ThreadedDiscussion.SearchReplies Method

Method to obtain an array of topic replies matching search (see page 514) 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(int ForumIDs, ), String SearchTerms, ), EkEnumeration.SearchTypes
stType);
```

Visual Basic

```
Public Function SearchReplies(ByVal ForumIDs As Integer,  As ), ByVal SearchTerms As String,
As ), ByVal stType As EkEnumeration.SearchTypes) As EkTasks
```

Parameters

Parameters	Description
ForumIDs	Integer array of the Forum (see page 167) IDs you want to search (see page 514). Use zero length array, or 0, to search (see page 514) all.
SearchTerms	String array of the search (see page 514) terms (case insensitive).
stType	EkEnumeration.SearchTypes enumeration value for the search (see page 514) type.

Returns

An instance of type ekTasks (an array of Tasks).

API.Search Namespace

The Search Developer API (see page 363) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 424), if you chose to install it, has excellent examples of programmatically using the SearchManager (see page 515). 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 (see page 449) that contains the word *cms*.
- Retrieve content (see page 449) that contains *research* and has a searchable metadata *wellness* set to Cancer.
- Retrieve images that contain *RotatingGraphic1* in their title.
- Retrieve geo-content (see page 449) from CMS400.Net.

Classes

	Name	Description
	SearchManager (see page 515)	The Search (see page 517) Developer API (see page 363) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 424), 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 (see page 449) that contains the word *cms*. • Retrieve content (see page 449) that contains *research* and has a searchable metadata *wellness* set to Cancer. • Retrieve images that contain *RotatingGraphic1* in their title. • Retrieve geo-content (see page 449) from CMS400.Net.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	SearchManager (see page 515)	The Search (see page 517) Developer API (see page 363) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 424), 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 (see page 449) that contains the word *cms*. • Retrieve content (see page 449) that contains *research* and has a searchable metadata *wellness* set to Cancer. • Retrieve images that contain *RotatingGraphic1* in their title. • Retrieve geo-content (see page 449) from CMS400.Net.

Legend

	Class

SearchManager Class

The Search (see page 517) Developer API (see page 363) enables searching a CMS400.NET website programmatically based on custom criteria. The Demo Developers Site (see page 424), 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 (see page 449) that contains the word *cms*.
- Retrieve content (see page 449) that contains *research* and has a searchable metadata *wellness* set to Cancer.
- Retrieve images that contain *RotatingGraphic1* in their title.
- Retrieve geo-content (see page 449) 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 515)	Returns the array of ContentData (see page 738). This method is overloaded.
	Search (see page 517)	
	SearchForCategories (see page 519)	Search (see page 517) based on taxonomy category.
	SearchTable (see page 519)	This API (see page 363) is used internally by the Ektron Windows Service to update catalogs when a taxonomy category is updated This ensures that subsequent searches function correctly.

Legend

	Method
	virtual

Execute Method

Returns the array of ContentData (see page 738). This method is overloaded.

Overload List

	Name	Description
	SearchManager.Execute (Ektron.Cms.ContentSearchCondition) (see page 516)	Returns the array of ContentData (see page 738) based on the criteria ContentSearchCondition.

	SearchManager.Execute (Ektron.Cms.UserSearchCondition) (see page 516)	Returns the array of UserData (see page 801) based on the criteria UserSearchCondition.

SearchManager.Execute Method (Ektron.Cms.ContentSearchCondition)

Returns the array of ContentData (see page 738) 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 738)

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 801) 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 801)

Example

```
Dim isMemberShip As UserSearchCondition = New UserSearchCondition
isMemberShip.setType = SearchType.EQUAL
isMemberShip.setValue = 1 'The value can be integer,string,date and boolean should be match
with db type
isMemberShip.setVariable = users.membership_user

Dim isInEktron As UserSearchCondition = New UserSearchCondition
isInEktron.setType = SearchType.EQUAL
isInEktron.setValue = "03031"  'The value can be integer,string,date and boolean should be
match with dynamic_data_tbl labels type
isInEktron.setVariable = "customproperties.zip code"

Dim condition As UserSearchCondition = New UserSearchCondition
condition.setType = SearchType.AND
condition.AddCondition(isInEktron)
condition.AddCondition(isMemberShip)

Dim search As New SearchManager
Dim result As userdata() = search.Execute(condition)
```

Search Method

Overload List

	Name	Description
	SearchManager.Search (MapRequestData, HttpContext, Integer, Integer, Integer) (see page 517)	Map (see page 232) search for retrieving geographical content (see page 449) from CMS400.Net site. This function has two additional overloads.
	SearchManager.Search (SearchRequestData, HttpContext, Integer) (see page 518)	Search HTML, content (see page 449), images, multimedia and documents based on content (see page 449) or metadata. This function has two additional overloads.
	SearchManager.Search (String, HttpContext, Integer) (see page 518)	Search the site for occurrence(s) of search text. This function has two additional overloads.

SearchManager.Search Method (MapRequestData, HttpContext, Integer, Integer, Integer)

Map (see page 232) search for retrieving geographical content (see page 449) from CMS400.Net site.

This function has two additional overloads.

C#

```
public MapResponseData[] Search(MapRequestData mapReqData, HttpContext context, int index, int
count, 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 (see page 449) satisfying location criteria

SearchManager.Search Method (SearchRequestData, HttpContext, Integer)

Search HTML, content (see page 449), images, multimedia and documents based on content (see page 449) or metadata.

This function has two additional overloads.

C#

```
public SearchResponseData[] Search(SearchRequestData searchReqData, HttpContext context, 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 (see page 449) satisfying search criteria

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, 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 (see page 449) satisfying search criteria

SearchManager.SearchForCategories Method

Search (see page 517) based on taxonomy category.

C#

```
public String[] SearchForCategories(SearchRequestData searchReqData, HttpContext context, int index, int count, 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 517) 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 363) 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, 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 517) criteria bundled in a SearchRequestData object.
searchParams	Database columns to be included in search (see page 517).
context	The current Http request.
resultCount	Number of records returned.

Returns

DataSet with a single DataTable which contains content (see page 449) whose taxonomy was updated.

API.Task Namespace

This is namespace API.Task.

Classes

	Name	Description
	Task (see page 520)	This is class API.Task.Task.
	TaskCategory (see page 530)	This is class API.Task.TaskCategory.
	TaskCategoryType (see page 532)	This is class API.Task.TaskCategoryType.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Task (see page 520)	This is class API.Task.Task.
	TaskCategory (see page 530)	This is class API.Task.TaskCategory.
	TaskCategoryType (see page 532)	This is class API.Task.TaskCategoryType.

Legend

	Class

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 API.Task.Task.

Task Methods

The methods of the Task class are listed here.

Public Methods

	Name	Description
	AddTask (see page 521)	Adds a task (see page 520) in the system.
	AssignTaskToUser (see page 521)	
	AssignTaskToUsergroup (see page 522)	

	DeleteTask (see page 523)	
	DeleteTasks (see page 524)	
	GetTask (see page 525)	Get the task (see page 520) information.
	GetTasks (see page 525)	
	GetTasksAssignedByUser (see page 526)	
	GetTasksAssignedToUser (see page 527)	
	GetTasksAssignedToUsergroup (see page 528)	
	GetTaskState (see page 529)	Get the task (see page 520) state.
	SetStateForTask (see page 529)	Set the task (see page 520) state.
	UpdateTask (see page 529)	Updates the task (see page 520)

Legend

	Method

Task.AddTask Method

Adds a task (see page 520) in the system.

C#

```
public Boolean AddTask(TaskData data);
```

Visual Basic

```
Public Function AddTask(ByRef data As TaskData) As Boolean
```

Parameters

Parameters	Description
data	TaskData

Returns

False=if successfully added otherwise True

AssignTaskToUser Method

Overload List

	Name	Description
	Task.AssignTaskToUser (Integer, Integer) (see page 521)	Assign a task (see page 520) to a user (see page 534).
	Task.AssignTaskToUser (Integer, Integer, Integer) (see page 522)	Assign a task (see page 520) to a user (see page 534) by a specified user (see page 534).

Task.AssignTaskToUser Method (Integer, Integer)

Assign a task (see page 520) to a user (see page 534).

C#

```
public Boolean AssignTaskToUser(int taskId, int assignToUserId);
```

Visual Basic

```
Public Function AssignTaskToUser(ByVal taskId As Integer, ByVal assignToUserId As Integer) As
Boolean
```

Parameters

Parameters	Description
taskId	Task (see page 520) ID
assignToUserId	User (see page 534)'s ID

Returns

False = if successfully assigned to a user (see page 534). Otherwise, returns True.

Task.AssignTaskToUser Method (Integer, Integer, Integer)

Assign a task (see page 520) to a user (see page 534) by a specified user (see page 534).

C#

```
public Boolean AssignTaskToUser(int taskId, int assignToUserId, int assignByUserId);
```

Visual Basic

```
Public Function AssignTaskToUser(ByVal taskId As Integer, ByVal assignToUserId As Integer,
ByVal assignByUserId As Integer) As Boolean
```

Parameters

Parameters	Description
taskId	Task (see page 520) ID
assignToUserId	Assign to User (see page 534)'s ID
assignByUserId	Assign by User (see page 534)'s ID

Returns

False = if successfully assigned to a user (see page 534). Otherwise, returns True

AssignTaskToUsergroup Method

Overload List

	Name	Description
	Task.AssignTaskToUsergroup (Integer, Integer) (see page 522)	Assign a task (see page 520) to a usergroup.
	Task.AssignTaskToUsergroup (Integer, Integer, Integer) (see page 523)	Assign a task (see page 520) to a usergroup by a specified user (see page 534)..

Task.AssignTaskToUsergroup Method (Integer, Integer)

Assign a task (see page 520) to a usergroup.

C#

```
public Boolean AssignTaskToUsergroup(int taskId, int assignToUsergroup);
```

Visual Basic

```
Public Function AssignTaskToUsergroup(ByVal taskId As Integer, ByVal assignToUsergroup As
Integer) As Boolean
```

Parameters

Parameters	Description
taskId	Task (see page 520) ID
assignToUsergroup	Assign to Usergroup's ID

Returns

False = if successfully assigned to a usergroup. Otherwise, returns True.

Task.AssignTaskToUsergroup Method (Integer, Integer, Integer)

Assign a task (see page 520) to a usergroup by a specified user (see page 534)..

C#

```
public Boolean AssignTaskToUsergroup(int taskId, int assignToUsergroup, int assignByUserId);
```

Visual Basic

```
Public Function AssignTaskToUsergroup(ByVal taskId As Integer, ByVal assignToUsergroup As
Integer, ByVal assignByUserId As Integer) As Boolean
```

Parameters

Parameters	Description
taskId	Task (see page 520) ID
assignToUsergroup	Assign to Usergroup's ID
assignByUserId	Assign by User (see page 534)'s ID

Returns

False = if successfully assigned to a usergroup. Otherwise, returns True.

DeleteTask Method

Overload List

	Name	Description
	Task.DeleteTask (Integer) (see page 523)	Mark the task (see page 520) deleted.
	Task.DeleteTask (Integer, Boolean) (see page 524)	Mark the task (see page 520) deleted or delete the task (see page 520) permanently.

Task.DeleteTask Method (Integer)

Mark the task (see page 520) deleted.

C#

```
public Boolean DeleteTask(int taskId);
```

Visual Basic

```
Public Function DeleteTask(ByVal taskId As Integer) As Boolean
```

Parameters

Parameters	Description
taskId	Task (see page 520) ID

Returns

False=if successfully deleted otherwise True

Task.DeleteTask Method (Integer, Boolean)

Mark the task (see page 520) deleted or delete the task (see page 520) permanently.

C#

```
public Boolean DeleteTask(int taskId, Boolean Permanently);
```

Visual Basic

```
Public Function DeleteTask(ByVal taskId As Integer, ByVal Permanently As Boolean) As Boolean
```

Parameters

Parameters	Description
taskId	Task (see page 520) ID
Permanently	True=Delete the task (see page 520)

Returns

False=if successfully deleted otherwise True

DeleteTasks Method

Overload List

	Name	Description
	Task.DeleteTasks (Integer) (see page 524)	Mark tasks deleted.
	Task.DeleteTasks (Integer, Boolean) (see page 525)	Mark tasks deleted or delete tasks permanently.

Task.DeleteTasks Method (Integer)

Mark tasks deleted.

C#

```
public Boolean DeleteTasks(int taskId);
```

Visual Basic

```
Public Function DeleteTasks(ByVal taskId() As Integer) As Boolean
```

Parameters

Parameters	Description
taskId	Array of Task (see page 520) ID

Returns

False=if successfully deleted otherwise True

Remarks

When an error occurs the method will quit without deleting rest of the tasks.

Task.DeleteTasks Method (Integer, Boolean)

Mark tasks deleted or delete tasks permanently.

C#

```
public Boolean DeleteTasks(int taskId, Boolean Permanently);
```

Visual Basic

```
Public Function DeleteTasks(ByVal taskId() As Integer, ByVal Permanently As Boolean) As Boolean
```

Parameters

Parameters	Description
taskId	Array of Task (see page 520) ID
Permanently	True=Delete tasks

Returns

False=if successfully deleted otherwise True

Remarks

When an error occurs the method will quit without deleting rest of the tasks.

Task.GetTask Method

Get the task (see page 520) information.

C#

```
public TaskData GetTask(int taskId);
```

Visual Basic

```
Public Function GetTask(ByVal taskId As Integer) As TaskData
```

Parameters

Parameters	Description
taskId	Task (see page 520) integer ID

Returns

Task (see page 520) Data

GetTasks Method

Overload List

	Name	Description
	Task.GetTasks () (see page 525)	Get All tasks from the system.
	Task.GetTasks (Integer) (see page 526)	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 (see page 520) data.

Task.GetTasks Method (Integer)

Get All tasks from the system.

C#

```
public TaskData[] GetTasks(int folderId);
```

Visual Basic

```
Public Function GetTasks(ByVal folderId() As Integer) As TaskData()
```

Parameters

Parameters	Description
folderId	Array of FolderIDs

Returns

Array of task (see page 520) data.

GetTasksAssignedByUser Method

Overload List

	Name	Description
	Task.GetTasksAssignedByUser (Integer) (see page 526)	Get all tasks assigned by the user (see page 534).
	Task.GetTasksAssignedByUser (Integer, EkEnumeration.TaskState) (see page 526)	Get all task (see page 520) assign by the user (see page 534) by a state of tasks specified.

Task.GetTasksAssignedByUser Method (Integer)

Get all tasks assigned by the user (see page 534).

C#

```
public TaskData[] GetTasksAssignedByUser(int userId);
```

Visual Basic

```
Public Function GetTasksAssignedByUser(ByVal userId As Integer) As TaskData()
```

Parameters

Parameters	Description
userId	User (see page 534)'s ID

Returns

Array of tasks.

Task.GetTasksAssignedByUser Method (Integer, EkEnumeration.TaskState)

Get all task (see page 520) assign by the user (see page 534) by a state of tasks specified.

C#

```
public TaskData[] GetTasksAssignedByUser(int userId, EkEnumeration.TaskState taskState);
```

Visual Basic

```
Public Function GetTasksAssignedByUser(ByVal userId As Integer, ByVal taskState As
EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
`userId`	User (see page 534)'s ID
`taskState`	Task (see page 520) State

Returns

Array of tasks.

GetTasksAssignedToUser Method

Overload List

	Name	Description
	Task.GetTasksAssignedToUser (Integer) (see page 527)	Get all tasks assigned to the user (see page 534).
	Task.GetTasksAssignedToUser (Integer, EkEnumeration.TaskState) (see page 527)	Get all tasks assigned to the user (see page 534) by a state of the task (see page 520) specified.

Task.GetTasksAssignedToUser Method (Integer)

Get all tasks assigned to the user (see page 534).

C#

```
public TaskData[] GetTasksAssignedToUser(int userId);
```

Visual Basic

```
Public Function GetTasksAssignedToUser(ByVal userId As Integer) As TaskData()
```

Parameters

Parameters	Description
`userId`	User (see page 534)'s ID

Returns

Array of tasks.

Task.GetTasksAssignedToUser Method (Integer, EkEnumeration.TaskState)

Get all tasks assigned to the user (see page 534) by a state of the task (see page 520) specified.

C#

```
public TaskData[] GetTasksAssignedToUser(int userId, EkEnumeration.TaskState taskState);
```

Visual Basic

```
Public Function GetTasksAssignedToUser(ByVal userId As Integer, ByVal taskState As
EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
`userId`	User (see page 534)'s ID
`taskState`	Task (see page 520) State

Returns

Array of tasks.

GetTasksAssignedToUsergroup Method

Overload List

	Name	Description
	Task.GetTasksAssignedToUsergroup (Integer) (see page 528)	Get all tasks assigned to the usergroup.
	Task.GetTasksAssignedToUsergroup (Integer, EkEnumeration.TaskState) (see page 528)	Get all tasks assigned to the usergroup by a state of tasks specified.

Task.GetTasksAssignedToUsergroup Method (Integer)

Get all tasks assigned to the usergroup.

C#

```
public TaskData[] GetTasksAssignedToUsergroup(int groupId);
```

Visual Basic

```
Public Function GetTasksAssignedToUsergroup(ByVal groupId As Integer) As TaskData()
```

Parameters

Parameters	Description
`groupId`	Usergroup's ID

Returns

Array of tasks.

Task.GetTasksAssignedToUsergroup Method (Integer, EkEnumeration.TaskState)

Get all tasks assigned to the usergroup by a state of tasks specified.

C#

```
public TaskData[] GetTasksAssignedToUsergroup(int groupId, EkEnumeration.TaskState taskState);
```

Visual Basic

```
Public Function GetTasksAssignedToUsergroup(ByVal groupId As Integer, ByVal taskState As
EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
`groupId`	Usergroup's ID
`taskState`	Task (see page 520) State

Returns

Array of tasks.

Task.GetTaskState Method

Get the task (see page 520) state.

C#

```
public EkEnumeration.TaskState GetTaskState(int taskId);
```

Visual Basic

```
Public Function GetTaskState(ByVal taskId As Integer) As EkEnumeration.TaskState
```

Parameters

Parameters	Description
taskId	Task (see page 520) ID

Returns

If successful returns task (see page 520) state otherwise it will throw an error.

Task.SetStateForTask Method

Set the task (see page 520) state.

C#

```
public Boolean SetStateForTask(int taskId, EkEnumeration.TaskState newTaskState);
```

Visual Basic

```
Public Function SetStateForTask(ByVal taskId As Integer, ByVal newTaskState As
EkEnumeration.TaskState) As Boolean
```

Parameters

Parameters	Description
taskId	Task (see page 520) ID
newTaskState	New Task (see page 520) State

Returns

False = if successfully state of the changed. Otherwise, returns True.

Task.UpdateTask Method

Updates the task (see page 520)

C#

```
public Boolean UpdateTask(TaskData data);
```

Visual Basic

```
Public Function UpdateTask(ByRef data As TaskData) As Boolean
```

Parameters

Parameters	Description
data	Task (see page 520) Data

Returns

False=if successfully updated otherwise 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 API.Task.TaskCategory.

TaskCategory Methods

The methods of the TaskCategory class are listed here.

Public Methods

	Name	Description
	AddTaskCategory (see page 530)	Adds a task (see page 519) category.
	DeleteTaskCategory (see page 531)	Deletes the task (see page 519) category.
	GetAllCategories (see page 531)	Gets All Task (see page 520) categories and task (see page 519) types belongs to those categories.
	GetTaskCategory (see page 531)	Gets the specified task (see page 519) category and task (see page 519) types belongs to the category.
	UpdateTaskCategory (see page 531)	Updates the category title

Legend

	Method

TaskCategory.AddTaskCategory Method

Adds a task (see page 519) category.

C#

```
public int AddTaskCategory(String title);
```

Visual Basic

```
Public Function AddTaskCategory(ByVal title As String) As Integer
```

Parameters

Parameters	Description
`title`	Category Title

Returns

Category ID as an integer.

TaskCategory.DeleteTaskCategory Method

Deletes the task (see page 519) category.

C#

```
public Boolean DeleteTaskCategory(int categoryId);
```

Visual Basic

```
Public Function DeleteTaskCategory(ByVal categoryId As Integer) As Boolean
```

Parameters

Parameters	Description
categoryId	Category ID

Returns

False = if successfully deleted. Otherwise, returns True.

TaskCategory.GetAllCategories Method

Gets All Task (see page 520) categories and task (see page 519) types belongs to those categories.

C#

```
public TaskCategoryData[] GetAllCategories();
```

Visual Basic

```
Public Function GetAllCategories() As TaskCategoryData()
```

Returns

Array of TaskCategoryData

TaskCategory.GetTaskCategory Method

Gets the specified task (see page 519) category and task (see page 519) types belongs to the category.

C#

```
public TaskCategoryData GetTaskCategory(int categoryID);
```

Visual Basic

```
Public Function GetTaskCategory(ByVal categoryID As Integer) As TaskCategoryData
```

Parameters

Parameters	Description
categoryID	Category ID

Returns

TaskCategoryData

TaskCategory.UpdateTaskCategory Method

Updates the category title

C#

```
public Boolean UpdateTaskCategory(int categoryId, String title);
```

Visual Basic

```
Public Function UpdateTaskCategory(ByVal categoryId As Integer, ByVal title As String) As
Boolean
```

Parameters

Parameters	Description
`categoryId`	Category Id
`title`	New Title

Returns

False = if successfully updated. Otherwise, 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 API.Task.TaskCategoryType.

TaskCategoryType Methods

The methods of the TaskCategoryType class are listed here.

Public Methods

	Name	Description
	AddTaskType (see page 532)	Adds a task (see page 519) type to a category.
	DeleteTaskType (see page 533)	Deletes the task (see page 519) type.
	GetTaskType (see page 533)	Gets the task (see page 519) type information.
	UpdateTaskType (see page 533)	Updates the task (see page 519) type.

Legend

	Method

TaskCategoryType.AddTaskType Method

Adds a task (see page 519) type to a category.

C#

```
public int AddTaskType(String title, String description, int categoryId, Boolean active);
```

Visual Basic

```
Public Function AddTaskType(ByVal title As String, ByVal description As String, ByVal
categoryId As Integer, ByVal active As Boolean) As Integer
```

Parameters

Parameters	Description
title	Title
description	Description
categoryId	Task (see page 520) Category ID
active	Set as active

Returns

Task (see page 520) Type ID

TaskCategoryType.DeleteTaskType Method

Deletes the task (see page 519) type.

C#

```
public Boolean DeleteTaskType(int taskTypeId);
```

Visual Basic

```
Public Function DeleteTaskType(ByVal taskTypeId As Integer) As Boolean
```

Parameters

Parameters	Description
taskTypeId	task (see page 519) type id to be deleted.

Returns

False = if successfully deleted. Otherwise, returns True.

TaskCategoryType.GetTaskType Method

Gets the task (see page 519) type information.

C#

```
public TaskTypeData GetTaskType(int taskTypeId);
```

Visual Basic

```
Public Function GetTaskType(ByVal taskTypeId As Integer) As TaskTypeData
```

Parameters

Parameters	Description
taskTypeId	TaskTypeId

Returns

TaskTypeData

TaskCategoryType.UpdateTaskType Method

Updates the task (see page 519) type.

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`	Task (see page 520) Type ID
`title`	Title
`description`	Description
`active`	Set as active

Returns

False = if successfully updated. Otherwise, returns True.

API.User Namespace

This is namespace API.User.

Classes

	Name	Description
	ADUser (see page 534)	This is class API.User.ADUser.
	User (see page 542)	This is class API.User.User.

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	ADUser (see page 534)	This is class API.User.ADUser.
	User (see page 542)	This is class API.User.User.

Legend

	Class

ADUser Class

Class Hierarchy

C#

```
public class ADUser : User;
```

Visual Basic

```
Public Class ADUser
Inherits User
```

File

ADUser.vb

Description

This is class API.User.ADUser.

ADUser Methods

The methods of the ADUser class are listed here.

Public Methods

	Name	Description
	AddADGroupToCMS (see page 535)	This function allows you add Active Directory (see page 118) groups to the CMS.
	AddADUsersToCMSByUsername (see page 536)	Adds Active Directory (see page 118) users to the CMS by their user (see page 534) name and domain.
	DeSynchUserGroups (see page 536)	Updates the group with a new group name.
	DeSynchUsers (see page 536)	Update the user (see page 534) with new user (see page 534) details.
	GetADMapping (see page 536)	Loads the CMS and Active Directory (see page 118) name.
	GetADStatus (see page 537)	Used to load required Synch, Desynch and relationship flags of the configured Active Directory (see page 118).
	GetAvailableADGroups (see page 537)	Loads all of the Active Directory (see page 118) groups by a given domain and filter.
	GetAvailableADUsers (see page 537)	Loads all available Active Directory (see page 118) users.
	GetCMSGroupsToSync (see page 538)	Loads all the CMS groups that need to be synchronized.
	GetCMSRelationshipsToSync (see page 538)	Loads all the CMS relationships that need to be synchronized with Active Directory (see page 118).
	GetCMSUsersToSync (see page 538)	Loads all the CMS users that need to be synchronized with Active Directory (see page 118).
	GetDomains (see page 539)	Loads a list of domains from Active Directory (see page 118).
	MapCMSUserGroupToAD (see page 539)	Updates the given CMS group corresponding to the Active Directory (see page 118) group.
	MapCMSUserToAD (see page 539)	Maps the given user (see page 534) ID to the corresponding Active Directory (see page 118) entry.
	SynchCMSGroupsToAD (see page 540)	Synchronizes the CMS group data to the Active Directory (see page 118) group data. This updates the CMS data with the corresponding to Active Directory (see page 118) changes.
	SynchCMSRelationshipsToAD (see page 540)	Synchronizes the CMS relations, such as groups, to the Active Directory (see page 118) for the given users.
	SynchCMSUsersToAD (see page 540)	Synchronize the CMS user (see page 534) data to the Active Directory (see page 118) data.
	UpdateADMapping (see page 541)	Updates the Active directory information into the CMS based on the user (see page 534) ID and user (see page 534) data.
	UpdateADUsersGroups (see page 541)	Updates Active Directory (see page 118) users in a CMS group.
	UpdateUsersInfoFromAD (see page 541)	Updates the user (see page 534) details from the Active Directory (see page 118) details.

Legend

	Method

ADUser.AddADGroupToCMS Method

This function allows you add Active Directory (see page 118) 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 118) group paths.

ADUser.AddADUsersToCMSByUsername Method

Adds Active Directory (see page 118) users to the CMS by their user (see page 534) 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 (see page 534) 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 true, update the user (see page 534) group with the new group information, otherwise none.

Returns

Ektron.Cms.AdDeSyncGroupData()

ADUser.DeSynchUsers Method

Update the user (see page 534) with new user (see page 534) details.

C#

```
public AdDeSyncUserData[] DeSynchUsers(Boolean Flag);
```

Visual Basic

```
Public Function DeSynchUsers(ByVal Flag As Boolean) As AdDeSyncUserData()
```

Parameters

Parameters	Description
Flag	If true, update the user (see page 534) with new user (see page 534) information, otherwise none.

Returns

Ektron.Cms.AdDeSyncUserData()

ADUser.GetADMapping Method

Loads the CMS and Active Directory (see page 118) name.

C#

```
public Overloads AdMappingData[] GetADMapping(int CallerId, String MapType, int Map, int
Exclude, int Editable);
```

Visual Basic

```
Public Overloads Function GetADMapping(ByVal CallerId As Integer, ByVal MapType As String,
ByVal Map As Integer, ByVal Exclude As Integer, ByVal Editable As Integer) As AdMappingData()
```

Parameters

Parameters	Description
CallerId	Who is calling this function.
MapType	Possible value is “userprop” (user (see page 534) properties).
Map	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
Exclude	Flag (0 – false, 1- true).
Editable	Flag(0 - false, 1 - true).

Returns

Ektron.Cms.AdMappingData()

ADUser.GetADStatus Method

Used to load required Synch, Desynch and relationship flags of the configured Active Directory (see page 118).

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 118) groups by a given domain and filter.

C#

```
public GroupData[] GetAvailableADGroups(String Filter, String Domain);
```

Visual Basic

```
Public Function GetAvailableADGroups(ByVal Filter As String, ByVal Domain As String) As
GroupData()
```

Parameters

Parameters	Description
Filter	Filter by complete or a specific group name.
Domain	The domain name. (Required)

Returns

Ektron.Cms.GroupData()

ADUser.GetAvailableADUsers Method

Loads all available Active Directory (see page 118) 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 801)()

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 118).

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 118).

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 801)()

ADUser.GetDomains Method

Loads a list of domains from Active Directory (see page 118).

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 118) (1 = check otherwise resumed).

Returns

Ektron.Cms.DomainData()

ADUser.MapCMSUserGroupToAD Method

Updates the given CMS group corresponding to the Active Directory (see page 118) group.

C#

```
public MapCMSUserGroupToAD(int GroupId, String GroupName, String Domain);
```

Visual Basic

```
Public Sub MapCMSUserGroupToAD(ByVal GroupId As Integer, ByVal GroupName As String, ByVal
Domain As String)
```

Parameters

Parameters	Description
GroupId	CMS group ID.
GroupName	CMS group name.
Domain	The Active Directory (see page 118) domain.

ADUser.MapCMSUserToAD Method

Maps the given user (see page 534) ID to the corresponding Active Directory (see page 118) entry.

C#

```
public MapCMSUserToAD(int UserId, String UserName, String Domain, int PullInfo);
```

Visual Basic

```
Public Sub MapCMSUserToAD(ByVal UserId As Integer, ByVal UserName As String, ByVal Domain As
String, ByVal PullInfo As Integer)
```

Parameters

Parameters	Description
UserId	The user (see page 534)'s ID.
UserName	The user (see page 534)'s name.
Domain	The user (see page 534)'s domain.
PullInfo	Flag to load Active Directory (see page 118) information. (1 = Load Active Directory (see page 118) information, otherwise none.)

ADUser.SynchCMSGroupsToAD Method

Synchronizes the CMS group data to the Active Directory (see page 118) group data. This updates the CMS data with the corresponding to Active Directory (see page 118) 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 118) 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 (see page 534) IDs.
GroupIds	A collection of group IDs.
Actions	A collection of actions (possible action is “delete”).

ADUser.SynchCMSUsersToAD Method

Synchronize the CMS user (see page 534) data to the Active Directory (see page 118) 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 (see page 534) 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 118) information when the value is 1, otherwise none.

ADUser.UpdateADMapping Method

Updates the Active directory information into the CMS based on the user (see page 534) ID and user (see page 534) data.

C#

```
public UpdateADMapping(int UserId, Object pagedata);
```

Visual Basic

```
Public Sub UpdateADMapping(ByVal UserId As Integer, ByVal pagedata As Object)
```

Parameters

Parameters	Description
UserId	The user (see page 534)’s ID.
pagedata	Key, value pair for given user (see page 534) ID.

ADUser.UpdateADUsersGroups Method

Updates Active Directory (see page 118) 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 (see page 534)’s username.
Domain	The user (see page 534)’s Domain.

ADUser.UpdateUsersInfoFromAD Method

Updates the user (see page 534) details from the Active Directory (see page 118) 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 (see page 534)’s username.
Domain	The user (see page 534)’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 API.User.User.

User Methods

The methods of the User class are listed here.

Public Methods

	Name	Description
	ActivateUserAccount (see page 544)	Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters. This method is overloaded
	AddMembershipUser (see page 545)	Adds a new membership user (see page 542) to CMS.
	AddNewUser (see page 547)	**This method is obsolete. Please use the AddMembershipUser (see page 545) method.** Adds a new membership user (see page 542) to CMS.
	AddRolePermission (see page 547)	Adds permission for the given role.
	AddUser (see page 548)	
	AddUserGroup (see page 548)	Creates new user (see page 542) group.
	AddUserToGroup (see page 549)	Adds an existing user (see page 542) to an existing group.
	AutoLogInUser (see page 549)	Login (see page 229) the given user (see page 542) credentials.
	CreateBlankUserObject (see page 549)	This method is overloaded.
	DeleteUserByID (see page 550)	Removes the user (see page 542) from the system. This will not purge the data, but it disables the user (see page 542) from further accessing the system.
	DeleteUserByIds (see page 550)	Removes the user (see page 542) from the system. This will not purge the data, but it disables the user (see page 542) from further accessing the system.
	DeleteUserFromGroup (see page 551)	Removes a user (see page 542) from the designated group.
	DeleteUserGroup (see page 551)	Removes the group from the system permanently.
	EditUserCustomProperties (see page 551)	This method creates the editable fields(UI) for user (see page 542)'s custom properties.
	GetActiveUser (see page 552)	Returns all of the user (see page 542) properties with or without preference.
	GetADMapping (see page 552)	Loads the CMS and Active Directory (see page 118) name.
	GetAllCmsUsers (see page 552)	Loads all the CMS users.

	GetAllCustomProperties (see page 553)	Returns array of all the custom properties defined in the system.
	GetAllUserByCustomProperty (see page 553)	Returns all the users based on the given properties.
	GetAllUserGroups (see page 553)	Loads all of the user groups from the system. This method is overloaded.
	GetAllUsers (see page 554)	This method is overloaded.
	GetGroupsUserIsIn (see page 555)	Loads all the groups by user (see page 542).
	GetMembershipUser (see page 556)	Returns user (see page 542) data based on the user (see page 542)'s login name (username).
	GetUser (see page 556)	Loads the user (see page 542) details for the given user (see page 542).
	GetUserByUsername (see page 556)	Loads the user (see page 542) details for the given user (see page 542).
	GetUserCustomProperties (see page 558)	Returns an array of custom properties for a given user (see page 542)'s ID.
	GetUserEmailInfo (see page 558)	Returns user (see page 542) data for a given user (see page 542)'s ID.
	GetUserForFolderAdmin (see page 559)	
	GetUserGroup (see page 559)	Loads a given group's details.
	GetUserGroupByName (see page 559)	Loads a given group's details.
	GetUserGroupForFolderAdmin (see page 559)	Returns all the group's data for the administrator.
	GetUserPermissions (see page 560)	Loads the user (see page 542) permissions as a array of UserPermissionData.
	GetUserPreference (see page 560)	Loads the preferences for the given user (see page 542).
	GetUsers (see page 561)	An Overloaded function that allows you to get the users from a specific group or load and array of user group data. This method is overloaded.
	GetUsersForTask (see page 562)	Loads the users that have permissions to the given content (see page 449).
	GetUsersNotInGroup (see page 562)	Loads all the users that are not in the given group.
	IsAdmin (see page 563)	Determines the if the current user (see page 542) is an Administrator.
	IsARoleMember (see page 563)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method is overloaded.
	IsARoleMemberForContent (see page 566)	Determines whether a Role Member is allowed to work with a given piece of content (see page 449) as an administrator or not. Administrations privileges are for the given piece of content (see page 449) 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 (see page 542).
	IsARoleMemberForContent_FolderUserAdmin (see page 566)	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 (see page 542).
	IsARoleMemberForFolder (see page 566)	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 (see page 542).
	IsARoleMemberForFolder_FolderUserAdmin (see page 567)	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 (see page 542).
	IsARoleMemberForFolder_MetadataAdmin (see page 567)	Determines whether a Role Member is allowed to work with a given folder as an meta data administrator or not.
	LogInUser (see page 568)	Used to log in the user (see page 542).
	ReadCustomProperties (see page 570)	Returns the custom properties as a Visual Basic collection. Importing the Microsoft.VisualBasic name space might be required.
	ResetMembershipUserPassword (see page 570)	
	SetAuthenticationCookie (see page 571)	This method sets the authentication cookie.
	UnsubscribeUser (see page 571)	Unsubscribes a user (see page 542) from the system.

	UpdateCustomPropertiesItemOrder (see page 571)	Updates the order of available custom properties.
	UpdateUser (see page 572)	
	UpDateUserGroup (see page 572)	
	UpdateUserPreferences (see page 573)	Updates the given user (see page 542)'s preference data.
	UserObject (see page 573)	Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Legend

	Method

ActivateUserAccount Method

Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters.

This method is overloaded

Overload List

	Name	Description
	User.ActivateUserAccount (Integer) (see page 544)	Activates a deleted (deactivated) user (see page 542) into the current/active state using the UserId. This function is overloaded. See Also: ActivateUserAccount (String String)
	User.ActivateUserAccount (String, String) (see page 544)	Activates the deleted (deactivated) user (see page 542) into the current/active state using the Username and GUID (AccountID) parameter. This function is overloaded. See Also: ActivateUserAccount (Integer)

User.ActivateUserAccount Method (Integer)

Activates a deleted (deactivated) user (see page 542) into the current/active state using the UserId.

This function is overloaded. See Also: ActivateUserAccount (String String)

C#

```
public Boolean ActivateUserAccount(int UserId);
```

Visual Basic

```
Public Function ActivateUserAccount(ByVal UserId As Integer) As Boolean
```

Parameters

Parameters	Description
`UserId`	A unique ID for the required user (see page 542).

Returns

Boolean

User.ActivateUserAccount Method (String, String)

Activates the deleted (deactivated) user (see page 542) into the current/active state using the Username and GUID (AccountID) parameter.

This function is overloaded. See Also: ActivateUserAccount (Integer)

C#

```
public Boolean ActivateUserAccount(String Username, String AccountID);
```

Visual Basic

```
Public Function ActivateUserAccount(ByVal Username As String, ByVal AccountID As String) As Boolean
```

Parameters

Parameters	Description
Username	The login name for the user (see page 542).
AccountID	The GUID for the user (see page 542).

Returns

Boolean

User.AddMembershipUser Method

Adds a new membership user (see page 542) 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 (see page 542) elements.

Returns

Ektron.Cms.UserData (see page 801)

Example

The following example shows how to create a Web page from which a new membership user (see page 542) 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 240) User (see page 542). 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 (see page 542) data items.

```
Dim memDetails As New Ektron.Cms.UserData
```

3. Create an object for the new membership user (see page 542) information.

```
Dim newMemUser As New Ektron.Cms.UserData
```

4. Create an object for the User (see page 542) API (see page 363).

```
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 (see page 542) 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 545) method.

Adds a new membership user (see page 542) 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 (see page 542) elements.

Returns

Ektron.Cms.UserData (see page 801)

User.AddRolePermission Method

Adds permission for the given role.

C#

```
public Boolean AddRolePermission(String RoleName);
```

Visual Basic

```
Public Function AddRolePermission(ByVal RoleName As String) As Boolean
```

Parameters

Parameters	Description
RoleName	Required role name.

Returns

True if successfully added. Otherwise, returns false

AddUser Method

Overload List

	Name	Description
	User.AddUser (Collection) (see page 548)	Adds a new user (see page 542) to CMS.
	User.AddUser (UserData) (see page 548)	Adds a new user (see page 542) to CMS.

User.AddUser Method (Collection)

Adds a new user (see page 542) 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 542) details as a value, key pair.

Returns

Ektron.Cms.UserData (see page 801)

User.AddUser Method (UserData)

Adds a new user (see page 542) 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 542) details as a value strongly typed UserData (see page 801) object.

Returns

Ektron.Cms.UserData (see page 801)

User.AddUserGroup Method

Creates new user (see page 542) group.

C#

```
public Collection AddUserGroup(String GroupName, String GroupPath, String GroupDomain);
```

Visual Basic

```
Public Function AddUserGroup(ByVal GroupName As String, Optional ByVal GroupPath As String =
```

```
"", Optional ByVal GroupDomain As String = "") As Collection
```

Parameters

Parameters	Description
GroupName	The name of the group.
GroupPath	Group path, usually required for Active Directory (see page 118), otherwise CMS group assumed.
GroupDomain	Group domain, usually required for Active Directory (see page 118).

User.AddUserToGroup Method

Adds an existing user (see page 542) to an existing group.

C#

```
public AddUserToGroup(int UserId, int GroupId);
```

Visual Basic

```
Public Sub AddUserToGroup(ByVal UserId As Integer, ByVal GroupId As Integer)
```

Parameters

Parameters	Description
UserId	The user (see page 542)'s ID.
GroupId	The required group's ID.

User.AutoLogInUser Method

Login (see page 229) the given user (see page 542) 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 (see page 542) name
domain	domain
ServerName	server name

Returns

True if the user (see page 542) successfully logged in otherwise false.

CreateBlankUserObject Method

This method is overloaded.

Overload List

	Name	Description
	User.CreateBlankUserObject () (see page 550)	Returns the instance of the user (see page 542) data. This function is overloaded. See Also: CreateBlankUserObject (UserData (see page 801))
	User.CreateBlankUserObject (UserData) (see page 550)	Sets the given user (see page 542) data values into the UserData (see page 801) class. This function is overloaded. See Also: CreateBlankUserObject (see page 550)

User.CreateBlankUserObject Method ()

Returns the instance of the user (see page 542) data.

This function is overloaded. See Also: CreateBlankUserObject (UserData (see page 801))

C#

```
public UserData CreateBlankUserObject();
```

Visual Basic

```
Public Function CreateBlankUserObject() As UserData
```

Returns

Ektron.Cms.UserData (see page 801)

User.CreateBlankUserObject Method (UserData)

Sets the given user (see page 542) data values into the UserData (see page 801) class.

This function is overloaded. See Also: CreateBlankUserObject (see page 550)

C#

```
public CreateBlankUserObject(UserData udUserData);
```

Visual Basic

```
Public Sub CreateBlankUserObject(ByRef udUserData As UserData)
```

Parameters

Parameters	Description
udUserData	An Object of user (see page 542) properties.

Remarks

User.DeleteUserByID Method

Removes the user (see page 542) from the system. This will not purge the data, but it disables the user (see page 542) from further accessing the system.

C#

```
public DeleteUserByID(int Id);
```

Visual Basic

```
Public Sub DeleteUserByID(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	The user (see page 542)'s ID.

User.DeleteUserByIds Method

Removes the user (see page 542) from the system. This will not purge the data, but it disables the user (see page 542) 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 (see page 542) IDs (comma separated values).

User.DeleteUserFromGroup Method

Removes a user (see page 542) from the designated group.

C#

```
public DeleteUserFromGroup(int UserId, int GroupId, Boolean ReturnDeleted);
```

Visual Basic

```
Public Sub DeleteUserFromGroup(ByVal UserId As Integer, ByVal GroupId As Integer, Optional
ByVal ReturnDeleted As Boolean = False)
```

Parameters

Parameters	Description
UserId	The user (see page 542)'s ID.
GroupId	The group's ID.

User.DeleteUserGroup Method

Removes the group from the system permanently.

C#

```
public DeleteUserGroup(int Id);
```

Visual Basic

```
Public Sub DeleteUserGroup(ByVal Id As Integer)
```

Parameters

Parameters	Description
Id	The group's ID.

User.EditUserCustomProperties Method

This method creates the editable fields(UI) for user (see page 542)'s custom properties.

C#

```
public String EditUserCustomProperties(int UserId, Boolean bReadOnly);
```

Visual Basic

```
Public Function EditUserCustomProperties(Optional ByVal UserId As Integer = 0, Optional ByVal
bReadOnly As Boolean = False) As String
```

Parameters

Parameters	Description
UserId	User (see page 542)'s ID. If the user (see page 542) ID is not specified then it will return empty fields.
bReadOnly	

Returns

String

User.GetActiveUser Method

Returns all of the user (see page 542) properties with or without preference.

C#

```
public UserData GetActiveUser(int Id, Boolean Preference);
```

Visual Basic

```
Public Function GetActiveUser(ByVal Id As Integer, Optional ByVal Preference As Boolean =
False) As UserData
```

Parameters

Parameters	Description
Id	The user (see page 542)'s ID.
Preference	Preference is returned only if the value true, otherwise nothing is returned for preference.

Returns

Ektron.Cms.UserData (see page 801)

User.GetADMapping Method

Loads the CMS and Active Directory (see page 118) name.

C#

```
public AdMappingData[] GetADMapping(int CallerId, String MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
Public Function GetADMapping(ByVal CallerId As Integer, ByVal MapType As String, ByVal Map As
Integer, ByVal Exclude As Integer, ByVal Editable As Integer) As AdMappingData()
```

Parameters

Parameters	Description
CallerId	Who is calling this function.
MapType	Possible value is “userprop” (user (see page 542) properties).
Map	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
Exclude	Flag (0 – false, 1-true).
Editable	Flag (0-false, 1-true).

Returns

Ektron.Cms.AdMappingData()

User.GetAllCmsUsers Method

Loads all the CMS users.

C#

```
public UserData[] GetAllCmsUsers(String OrderBy);
```

Visual Basic

```
Public Function GetAllCmsUsers(ByVal OrderBy As String) As UserData()
```

Parameters

Parameters	Description
`OrderBy`	Used to arrange data in a specific order. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData (see page 801)()

Description

User.GetAllCustomProperties Method

Returns array of all the custom properties defined in the system.

C#

```
public CustomAttribute[] GetAllCustomProperties();
```

Visual Basic

```
Public Function GetAllCustomProperties() As CustomAttribute()
```

Returns

Ektron.Cms.CustomAttribute()

User.GetAllUserByCustomProperty Method

Returns all the users based on the given properties.

C#

```
public UserData[]
GetAllUserByCustomProperty(Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes PropType,
String PropValues);
```

Visual Basic

```
Public Function GetAllUserByCustomProperty(ByVal PropType As
Ektron.Cms.Common.EkEnumeration.ObjectPropertyValueTypes, ByVal PropValues As String) As
UserData()
```

Parameters

Parameters	Description
`PropType`	
`PropValues`	Required property

GetAllUserGroups Method

Loads all of the user groups from the system. This method is overloaded.

Overload List

	Name	Description
	User.GetAllUserGroups (Ektron.Cms.Common.EkEnumeration.UserTypes, String) (see page 554)	Loads all of the user (see page 542) groups from the system. This function is overloaded. See Also: GetAllUserGroups (String)
	User.GetAllUserGroups (String) (see page 554)	Loads all of the user (see page 542) groups from the system. This function is overloaded. See Also: GetAllUserGroups (EkEnumeration.UserTypes String)

User.GetAllUserGroups Method (Ektron.Cms.Common.EkEnumeration.UserTypes, String)

Loads all of the user (see page 542) 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 240); 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 (see page 542) groups from the system.

This function is overloaded. See Also: GetAllUserGroups (EkEnumeration.UserTypes String)

C#

```
public UserGroupData[] GetAllUserGroups(String OrderBy);
```

Visual Basic

```
Public Function GetAllUserGroups(ByVal OrderBy As String) As UserGroupData()
```

Parameters

Parameters	Description
OrderBy	Arranges data in a specific order. Possible values are **GroupName** or **GroupID**.

Returns

Ektron.Cms.UserGroupData()

GetAllUsers Method

This method is overloaded.

Overload List

	Name	Description
	User.GetAllUsers (Integer, Integer) (see page 555)	Gets the User (see page 542) Group Data of all the allowed users in the folder specified.
	User.GetAllUsers (UserRequestData) (see page 555)	Returns the list of users from the database based on the UserRequestData class.

User.GetAllUsers Method (Integer, Integer)

Gets the User (see page 542) Group Data of all the allowed users in the folder specified.

C#

```
public UserGroupData[] GetAllUsers(int CallerID, int FolderId);
```

Visual Basic

```
Public Function GetAllUsers(ByVal CallerID As Integer, ByVal FolderId As Integer) As
UserGroupData()
```

Parameters

Parameters	Description
CallerID	The user (see page 542) ID that is logged on to the system.
FolderId	The ID of the folder.

Returns

Ektron.Cms.UserGroupData()

User.GetAllUsers Method (UserRequestData)

Returns the list of users from the database based on the UserRequestData class.

C#

```
public UserData[] GetAllUsers(UserRequestData userRequest);
```

Visual Basic

```
Public Function GetAllUsers(ByRef userRequest As UserRequestData) As UserData()
```

Parameters

Parameters	Description
userRequest	list of constraints to load users

Returns

An array of users as UserData (see page 801).

User.GetGroupsUserIsIn Method

Loads all the groups by user (see page 542).

C#

```
public GroupData[] GetGroupsUserIsIn(int UserId, String OrderBy);
```

Visual Basic

```
Public Function GetGroupsUserIsIn(ByVal UserId As Integer, ByVal OrderBy As String) As
```

```
GroupData()
```

Parameters

Parameters	Description
UserId	The user (see page 542)'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 (see page 542) data based on the user (see page 542)'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 542)'s login name (username). This is required

Returns

Ektron.Cms.UserData (see page 801)

User.GetUser Method

Loads the user (see page 542) details for the given user (see page 542).

C#

```
public UserData GetUser(int Id, Boolean Preference, Boolean ReturnDeleted);
```

Visual Basic

```
Public Function GetUser(ByVal Id As Integer, Optional ByVal Preference As Boolean = False,
Optional ByVal ReturnDeleted As Boolean = False) As UserData
```

Parameters

Parameters	Description
Id	The user (see page 542) ID.
Preference	Loads preference only if required (true = load preferences as a part of user (see page 542) data).

Returns

Ektron.Cms.UserData (see page 801)

User.GetUserByUsername Method

Loads the user (see page 542) details for the given user (see page 542).

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 801)

Example

The following example shows how to create a Web page that retrieves user (see page 542) 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 (see page 542) 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 (see page 542) API (see page 363).
```
            Dim userAPI As New Ektron.Cms.API.User.User
```
3. Create an object for the user (see page 542) 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 (see page 542) 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 (see page 542) 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 (see page 542)'s ID.

C#

```
public CustomAttribute[] GetUserCustomProperties(int UserId);
```

Visual Basic

```
Public Function GetUserCustomProperties(ByVal UserId As Integer) As CustomAttribute()
```

Parameters

Parameters	Description
UserId	User (see page 542)'s ID

Returns

Ektron.Cms.CustomAttribute()

User.GetUserEmailInfo Method

Returns user (see page 542) data for a given user (see page 542)'s ID.

C#

```
public UserData GetUserEmailInfo(int Id);
```

Visual Basic

```
Public Function GetUserEmailInfo(ByVal Id As Integer) As UserData
```

Parameters

Parameters	Description
Id	The user (see page 542)'s ID.

Returns

Object of UserData (see page 801)

User.GetUserForFolderAdmin Method

C#

```
public UserData GetUserForFolderAdmin(int FolderId, int Id, Boolean Preference, Boolean
ReturnDeleted);
```

Visual Basic

```
Public Function GetUserForFolderAdmin(ByVal FolderId As Integer, ByVal Id As Integer, Optional
ByVal Preference As Boolean = False, Optional ByVal ReturnDeleted As Boolean = False) As
UserData
```

Parameters

Parameters	Description
FolderId	
Id	
Preference	

User.GetUserGroup Method

Loads a given group's details.

C#

```
public UserGroupData GetUserGroup(int Id);
```

Visual Basic

```
Public Function GetUserGroup(ByVal Id As Integer) As UserGroupData
```

Parameters

Parameters	Description
Id	The group's ID.

Returns

Ektron.Cms.UserGroupData

User.GetUserGroupByName Method

Loads a given group's details.

C#

```
public UserGroupData GetUserGroupByName(String name);
```

Visual Basic

```
Public Function GetUserGroupByName(ByVal name As String) As UserGroupData
```

Parameters

Parameters	Description
name	The group's name.

Returns

Ektron.Cms.UserGroupData

User.GetUserGroupForFolderAdmin Method

Returns all the group's data for the administrator.

C#

```
public UserGroupData GetUserGroupForFolderAdmin(int FolderId, int Id);
```

Visual Basic

```
Public Function GetUserGroupForFolderAdmin(ByVal FolderId As Integer, ByVal Id As Integer) As
UserGroupData
```

Parameters

Parameters	Description
FolderId	Folder (see page 383) ID
Id	User (see page 542) ID

Returns

UserGroupData

User.GetUserPermissions Method

Loads the user (see page 542) permissions as a array of UserPermissionData.

C#

```
public UserPermissionData[] GetUserPermissions(int Id, String ItemType, int UserId, String
UserGroupList, PermissionUserType PermissionType, PermissionRequestType PermissionRequest);
```

Visual Basic

```
Public Function GetUserPermissions(ByVal Id As Integer, ByVal ItemType As String, ByVal UserId
As Integer, ByVal UserGroupList As String, Optional ByVal PermissionType As PermissionUserType
= PermissionUserType.All, Optional ByVal PermissionRequest As PermissionRequestType =
PermissionRequestType.All) As UserPermissionData()
```

Parameters

Parameters	Description
Id	The item's ID (content (see page 449) or folder id whose permissions you are seeking).
ItemType	The item's type (valid values - content (see page 449) or folder).
UserId	The user (see page 542)'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 (see page 542)).
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 (see page 542).

C#

```
public UserPreferenceData GetUserPreference(int Id);
```

Visual Basic

```
Public Function GetUserPreference(ByVal Id As Integer) As UserPreferenceData
```

Parameters

Parameters	Description
Id	The user (see page 542)'s ID.

Returns

Ektron.Cms.UserPreferenceData

GetUsers Method

An Overloaded function that allows you to get the users from a specific group or load and array of user group data. This method is overloaded.

Overload List

	Name	Description
	User.GetUsers (Integer, String) (see page 561)	Loads the all the users from the given group.
	User.GetUsers (String, String) (see page 561)	Returns an array of user (see page 542) group data based on the userName and userNameType.

User.GetUsers Method (Integer, String)

Loads the all the users from the given group.

C#

```
public UserData[] GetUsers(int GroupId, String OrderBy);
```

Visual Basic

```
Public Function GetUsers(ByVal GroupId As Integer, ByVal OrderBy As String) As UserData()
```

Parameters

Parameters	Description
`GroupId`	The group's ID
`OrderBy`	The order of the users. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

- Ektron.Cms.UserData (see page 801)()

Description

User.GetUsers Method (String, String)

Returns an array of user (see page 542) group data based on the userName and userNameType.

C#

```
public UserGroupData[] GetUsers(String userNames, String userNameTypes);
```

Visual Basic

```
Public Function GetUsers(ByVal userNames As String, Optional ByVal userNameTypes As String =
"") As UserGroupData()
```

Parameters

Parameters	Description
`userNames`	The user (see page 542)'s name. (Required)
`userNameTypes`	The user (see page 542) type possible values are **user (see page 542)** or **membership user (see page 542)**.

Returns

Ektron.Cms.UserGroupData()

User.GetUsersForTask Method

Loads the users that have permissions to the given content (see page 449).

C#

```
public UserGroupData[] GetUsersForTask(int CallerID, int ContentID);
```

Visual Basic

```
Public Function GetUsersForTask(ByVal CallerID As Integer, ByVal ContentID As Integer) As
UserGroupData()
```

Parameters

Parameters	Description
`CallerID`	The user (see page 542) ID that is logged on to the system.
`ContentID`	The ID of the content (see page 449). (Required)

Returns

Ektron.Cms.UserGroupData()

User.GetUsersNotInGroup Method

Loads all the users that are not in the given group.

C#

```
public UserData[] GetUsersNotInGroup(int Id, String OrderBy, int GroupType);
```

Visual Basic

```
Public Function GetUsersNotInGroup(ByVal Id As Integer, ByVal OrderBy As String, Optional
ByVal GroupType As Integer = 0) As UserData()
```

Parameters

Parameters	Description
`Id`	The group ID.
`OrderBy`	The order of the users. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData (see page 801)()

Description

User.IsAdmin Method

Determines the if the current user (see page 542) is an Administrator.

C#

```
public Boolean IsAdmin();
```

Visual Basic

```
Public Function IsAdmin() As Boolean
```

Returns

True if the logged in user (see page 542) is administrator. Otherwise, returns false.

IsARoleMember Method

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method is overloaded.

Overload List

	Name	Description
	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) (see page 564)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 542). This function is overloaded. See Also: • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Integer Boolean) • IsARoleMember (Integer Integer Boolean) • IsARoleMember (Integer)
	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean) (see page 564)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 542). This function is overloaded. See Also: • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) • IsARoleMember (Integer Integer Boolean) • IsARoleMember (Integer)
	User.IsARoleMember (Integer) (see page 565)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 542). This function is overloaded. See Also: • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Integer Boolean) • IsARoleMember (Integer Integer Boolean)

	User.IsARoleMember (Integer, Integer, Boolean) (see page 565)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 542). This function is overloaded. See Also: • IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Integer Boolean) • IsARoleMember (Integer) • 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 (see page 542).

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Integer Boolean)
- IsARoleMember (Integer Integer Boolean)
- IsARoleMember (Integer)

C#

```
public Boolean IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds) As
Boolean
```

Parameters

Parameters	Description
RoleId	_nt_

Remarks

User.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, Integer, Boolean)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 542).

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- IsARoleMember (Integer Integer Boolean)
- IsARoleMember (Integer)

C#

```
public Boolean IsARoleMember(Ektron.Cms.Common.EkEnumeration.CmsRoleIds RoleId, int nUserId,
Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Ektron.Cms.Common.EkEnumeration.CmsRoleIds,
ByVal nUserId As Integer, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
RoleId	_nt_
nUserId	_nt_
GroupFlag	_nt_

Remarks

User.IsARoleMember Method (Integer)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 542).

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)
- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Integer Boolean)
- IsARoleMember (Integer Integer Boolean)

C#

```
public Boolean IsARoleMember(int RoleId);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Integer) As Boolean
```

Parameters

Parameters	Description
RoleId	_nt_

Remarks

User.IsARoleMember Method (Integer, Integer, Boolean)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 542).

This function is overloaded. See Also:

- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds Integer Boolean)
- IsARoleMember (Integer)
- IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds)

C#

```
public Boolean IsARoleMember(int RoleId, int nUserId, Boolean GroupFlag);
```

Visual Basic

```
Public Function IsARoleMember(ByVal RoleId As Integer, ByVal nUserId As Integer, Optional
ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
RoleId	_nt_
nUserId	_nt_

GroupFlag	_nt_

Remarks

User.IsARoleMemberForContent Method

Determines whether a Role Member is allowed to work with a given piece of content (see page 449) as an administrator or not. Administrations privileges are for the given piece of content (see page 449) 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 (see page 542).

C#

```
public Boolean IsARoleMemberForContent(int RoleId, int ContentId, int nUserId, Boolean
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForContent(ByVal RoleId As Integer, ByVal ContentId As Integer,
Optional ByVal nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
RoleId	
ContentId	The numeric content (see page 449) 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 (see page 542).

C#

```
public Boolean IsARoleMemberForContent_FolderUserAdmin(int ContentId, int nUserId, Boolean
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForContent_FolderUserAdmin(ByVal ContentId As Integer, Optional
ByVal nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
ContentId	required content (see page 449) id
nUserId	user (see page 542) 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 (see page 542).

C#

```
public Boolean IsARoleMemberForFolder(int RoleId, int FolderId, int nUserId, Boolean
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForFolder(ByVal RoleId As Integer, ByVal FolderId As Integer,
Optional ByVal nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
RoleId	
FolderId	The numeric folder ID for which you are giving the Role Member privileges.
nUserId	
GroupFlag	

Returns

True if granted otherwise false

User.IsARoleMemberForFolder_FolderUserAdmin Method

Determines whether a Role Member is allowed to work with a given folder as an administrator or not. Administrations privileges are for the given folder only. Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 542).

C#

```
public Boolean IsARoleMemberForFolder_FolderUserAdmin(int FolderId, int nUserId, Boolean
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForFolder_FolderUserAdmin(ByVal FolderId As Integer, Optional
ByVal nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
FolderId	required folder id
nUserId	user (see page 542) id
GroupFlag	group id

Returns

True if granted otherwise false

User.IsARoleMemberForFolder_MetadataAdmin Method

Determines whether a Role Member is allowed to work with a given folder as an meta data administrator or not.

C#

```
public Boolean IsARoleMemberForFolder_MetadataAdmin(int FolderId, int nUserId, Boolean
GroupFlag);
```

Visual Basic

```
Public Function IsARoleMemberForFolder_MetadataAdmin(ByVal FolderId As Integer, Optional ByVal
nUserId As Integer = 0, Optional ByVal GroupFlag As Boolean = False) As Boolean
```

Parameters

Parameters	Description
FolderId	required folder id

nUserId	user (see page 542) id
GroupFlag	group id

Returns

True if granted otherwise false

User.LogInUser Method

Used to log in the user (see page 542).

C#

```
public UserData LogInUser(String username, String Password, String ServerName, String Domain,
String Protocol);
```

Visual Basic

```
Public Function LogInUser(ByVal username As String, ByVal Password As String, ByVal ServerName
As String, Optional ByVal Domain As String = "", Optional ByVal Protocol As String = "") As
UserData
```

Parameters

Parameters	Description
username	The system username.
Password	The system password.
ServerName	The system server name.
Domain	The user (see page 542)'s domain, required only when users login using Active Directory (see page 118).
Protocol	Required only when using a secure login or when you are using any valid Active Directory (see page 118) protocols.

Returns

Ektron.Cms.UserData (see page 801)

Example

The following example shows how to create a Web page that programmatically logs a user (see page 542) into CMS400.NET. Please note that the code-behind in this example does not login a new user (see page 542) 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 (see page 542) 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 (see page 542) 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 (see page 542) API (see page 363).

```
        Dim userAPI As New Ektron.Cms.API.User.User
```

4. Create an object for the common API (see page 363).

```
        Dim api As New Ektron.Cms.CommonApi
```

5. Create an object to store the user (see page 542) 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 (see page 542) is already logged in.

```
        Try
            If (api.RequestInformationRef.CallerId > 0) Then
                lblResult.Text = api.RequestInformationRef.LoggedInUsername & " is already
logged in."
```

8. If the user (see page 542) 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 (see page 542) 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 (see page 542) properties.

Returns

Microsoft.VisualBasic.Collection

ResetMembershipUserPassword Method

Overload List

	Name	Description
	User.ResetMembershipUserPassword (String) (see page 570)	Reset the password for the given membership user (see page 542).
	User.ResetMembershipUserPassword (String, String) (see page 570)	Reset the password for the given membership user (see page 542).

User.ResetMembershipUserPassword Method (String)

Reset the password for the given membership user (see page 542).

C#

```
public String ResetMembershipUserPassword(String Username);
```

Visual Basic

```
Public Function ResetMembershipUserPassword(ByVal Username As String) As String
```

Parameters

Parameters	Description
Username	User (see page 542)'s username to be reset.

Returns

String

User.ResetMembershipUserPassword Method (String, String)

Reset the password for the given membership user (see page 542).

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 542)'s username to be reset.
`AccountId`	If verification is turned on, specify verification id(email to a user (see page 542)).

Returns

String

User.SetAuthenticationCookie Method

This method sets the authentication cookie.

C#

```
public SetAuthenticationCookie(UserData userInfo);
```

Visual Basic

```
Public Sub SetAuthenticationCookie(ByVal userInfo As UserData)
```

Parameters

Parameters	Description
`userInfo`	Returned value of the logInUser (see page 568) method.

Remarks

First, you must call the logInUser (see page 568) method to login the user (see page 542).

User.UnsubscribeUser Method

Unsubscribes a user (see page 542) from the system.

C#

```
public Boolean UnsubscribeUser(String Username, String Password);
```

Visual Basic

```
Public Function UnsubscribeUser(ByVal Username As String, Optional ByVal Password As String =
"") As Boolean
```

Parameters

Parameters	Description
`Username`	The user (see page 542)'s username.
`Password`	The user (see page 542)'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 (see page 542).

Returns

Boolean

UpdateUser Method

Overload List

	Name	Description
	User.UpdateUser (Collection) (see page 572)	Updates the user (see page 542) details.
	User.UpdateUser (UserData) (see page 572)	Updates the user (see page 542) details.

User.UpdateUser Method (Collection)

Updates the user (see page 542) details.

C#

```
public UpdateUser(Collection data);
```

Visual Basic

```
Public Sub UpdateUser(ByVal data As Collection)
```

Parameters

Parameters	Description
`data`	A collection of user (see page 542) data.

User.UpdateUser Method (UserData)

Updates the user (see page 542) details.

C#

```
public UpdateUser(UserData data);
```

Visual Basic

```
Public Sub UpdateUser(ByVal data As UserData)
```

Parameters

Parameters	Description
`data`	A UserData (see page 801) object of user (see page 542) data.

UpDateUserGroup Method

Overload List

	Name	Description
	User.UpDateUserGroup (Collection) (see page 573)	Updates the user (see page 542) group details.

	User.UpDateUserGroup (UserData) (see page 573)	Updates the user (see page 542) group details.

User.UpDateUserGroup Method (Collection)

Updates the user (see page 542) 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 (see page 542) 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 801))

User.UpdateUserPreferences Method

Updates the given user (see page 542)'s preference data.

C#

```
public UpdateUserPreferences(int Id, Collection data);
```

Visual Basic

```
Public Sub UpdateUserPreferences(ByVal Id As Integer, ByVal data As Collection)
```

Parameters

Parameters	Description
Id	The user (see page 542)'s ID.
data	A collection of user (see page 542) preferences.

UserObject Method

Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Overload List

	Name	Description
	User.UserObject (Integer) (see page 574)	Returns user (see page 542) data based on the user (see page 542)'s ID. This function is overloaded. See Also: UserObject (String)
	User.UserObject (String) (see page 574)	Returns user (see page 542) data based on the user (see page 542)'s login or username. This function is overloaded. See Also: UserObject (Integer)

User.UserObject Method (Integer)

Returns user (see page 542) data based on the user (see page 542)'s ID.

This function is overloaded. See Also: UserObject (String)

C#

```
public UserData UserObject(int Id);
```

Visual Basic

```
Public Function UserObject(ByVal Id As Integer) As UserData
```

Parameters

Parameters	Description
Id	The user (see page 542)'s ID.

Returns

Ektron.Cms.UserData (see page 801)

User.UserObject Method (String)

Returns user (see page 542) data based on the user (see page 542)'s login or username.

This function is overloaded. See Also: UserObject (Integer)

C#

```
public UserData UserObject(String sUserName);
```

Visual Basic

```
Public Function UserObject(ByVal sUserName As String) As UserData
```

Parameters

Parameters	Description
sUserName	User (see page 542)'s login or username.

Returns

Ektron.Cms.UserData (see page 801)

User Properties

The properties of the User class are listed here.

Public Properties

	Name	Description
R	UserIdAuthenticated (see page 575)	Returns the id of the currently logged in user (see page 542).

Legend

	Property
	read only

User.UserIdAuthenticated Property

Returns the id of the currently logged in user (see page 542).

C#

```
public ReadOnly int UserIdAuthenticated;
```

Visual Basic

```
Public ReadOnly Property UserIdAuthenticated() As Integer
```

Returns

Boolean

Web Services API

The Web Services API (see page 363) 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 575)	This is namespace Ektron.Services.

Ektron.Services Namespace

This is namespace Ektron.Services.

Namespaces

Name	Description
Task (see page 576)	This is namespace Ektron.Services.Task.

Classes

	Name	Description
	ActiveDirectoryUser (see page 593)	Summary description for ActiveDirectoryUser
	Asset (see page 601)	Summary description for Asset
	Blog (see page 607)	Summary description for Blog
	BusinessRules (see page 619)	Summary description for BusinessRules
	Calendar (see page 621)	Summary description for Calendar
	CalendarEvent (see page 621)	Summary description for CalendarEvent
	CalendarEventType (see page 621)	Summary description for CalendarEventType
	Content (see page 622)	Summary description for ContentService
	ContentRating (see page 634)	Summary description for ContentRating

	CustomFields (see page 636)	Summary description for CustomFields
	Folder (see page 644)	Summary description for Folder
	Font (see page 648)	Summary description for Font
	Form (see page 651)	Summary description for Form
	Library (see page 658)	Summary description for Library
	Metadata (see page 667)	Summary description for Metadata
	Permissions (see page 671)	Summary description for Permissions
	SearchManager (see page 676)	Summary description for SearchManager
	Site (see page 681)	Summary description for Site
	SiteMap (see page 688)	Summary description for SiteMap
	Taxonomy (see page 691)	The summary description for Taxonomy.
	ThreadedDiscussion (see page 697)	Summary description for ThreadedDiscussion
	User (see page 706)	Summary description for User

Legend

	Class

Ektron.Services.Task Namespace

This is namespace Ektron.Services.Task.

Classes

	Name	Description
	Task (see page 576)	Summary description for Task
	TaskCategory (see page 587)	Summary description for TaskCategory
	TaskCategoryType (see page 590)	Summary description for TaskCategoryType

Legend

	Class

Classes

The following table lists classes in this documentation.

Classes

	Name	Description
	Task (see page 576)	Summary description for Task
	TaskCategory (see page 587)	Summary description for TaskCategory
	TaskCategoryType (see page 590)	Summary description for TaskCategoryType

Legend

	Class

Task Class

Summary description for Task

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Task : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Task
Inherits EktronBaseService
```

File

Task.cs

Task Methods

The methods of the Task class are listed here.

Public Methods

	Name	Description
	AddTask (see page 577)	Adds a task (see page 519) in the system.
	AssignTaskToUser (see page 578)	
	AssignTaskToUsergroup (see page 579)	
	DeleteTask (see page 580)	
	DeleteTasks (see page 581)	
	GetTask (see page 582)	Get the task (see page 519) information.
	GetTasks (see page 582)	Get All tasks from the system.
	GetTasksAssignedByUser (see page 583)	
	GetTasksAssignedToUser (see page 584)	
	GetTasksAssignedToUsergroup (see page 585)	
	GetTaskState (see page 586)	Get the task (see page 519) state.
	SetStateForTask (see page 586)	Set the task (see page 519) state.
	UpdateTask (see page 587)	Updates the task (see page 519)

Legend

	Method

Task.AddTask Method

Adds a task (see page 519) 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(data As ref TaskData) As bool
```

Parameters

Parameters	Description
data	TaskData

Returns

False=if successfully added otherwise True

AssignTaskToUser Method

Overload List

	Name	Description
	Task.AssignTaskToUser (int, int) (see page 578)	Assign a task (see page 519) to a user (see page 534).
	Task.AssignTaskToUser (int, int, int) (see page 578)	Assign a task (see page 519) to a user (see page 534) by a specified user (see page 534).

Task.AssignTaskToUser Method (int, int)

Assign a task (see page 519) to a user (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUser", Description = "Assign a task to a user.")]
public bool AssignTaskToUser(int taskId, int assignToUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUser", Description = "Assign a task to a user.")>
Public Function AssignTaskToUser(taskId As Integer, assignToUserId As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task (see page 576) ID
assignToUserId	User (see page 706)'s ID

Returns

False=if successfully assigned to a user (see page 534) otherwise True

Task.AssignTaskToUser Method (int, int, int)

Assign a task (see page 519) to a user (see page 534) by a specified user (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUserByUser", Description = "Assign a task to a user.")]
public bool AssignTaskToUser(int taskId, int assignToUserId, int assignByUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
```

```
<WebMethod(MessageName = "AssignTaskToUserByUser", Description = "Assign a task to a user.")>
Public Function AssignTaskToUser(taskId As Integer, assignToUserId As Integer, assignByUserId
As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task (see page 576) ID
assignToUserId	Assign to User (see page 706)'s ID
assignByUserId	Assign by User (see page 706)'s ID

Returns

False=if successfully assigned to a user (see page 534) otherwise True

AssignTaskToUsergroup Method

Overload List

	Name	Description
	Task.AssignTaskToUsergroup (int, int) (see page 579)	Assign a task (see page 519) to a usergroup.
	Task.AssignTaskToUsergroup (int, int, int) (see page 579)	Assign a task (see page 519) to a usergroup by a specified user (see page 534).

Task.AssignTaskToUsergroup Method (int, int)

Assign a task (see page 519) to a usergroup.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUsergroup", Description = "Assign a task to a
usergroup.")]
public bool AssignTaskToUsergroup(int taskId, int assignToUsergroup);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUsergroup", Description = "Assign a task to a
usergroup.")>
Public Function AssignTaskToUsergroup(taskId As Integer, assignToUsergroup As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task (see page 576) ID
assignToUsergroup	Assign to Usergroup's ID

Returns

False=if successfully assigned to a usergroup otherwise True

Task.AssignTaskToUsergroup Method (int, int, int)

Assign a task (see page 519) to a usergroup by a specified user (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AssignTaskToUsergroupByUser", Description = "Assign a task to a
```

```
usergroup by a specified user.")]
public bool AssignTaskToUsergroup(int taskId, int assignToUsergroup, int assignByUserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AssignTaskToUsergroupByUser", Description = "Assign a task to a
usergroup by a specified user.")>
Public Function AssignTaskToUsergroup(taskId As Integer, assignToUsergroup As Integer,
assignByUserId As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task (see page 576) ID
assignToUsergroup	Assign to Usergroup's ID
assignByUserId	Assign by User (see page 706)'s ID

Returns

False=if successfully assigned to a usergroup otherwise True

DeleteTask Method

Overload List

	Name	Description
	Task.DeleteTask (int) (see page 580)	Mark the task (see page 519) deleted.
	Task.DeleteTask (int, bool) (see page 580)	Mark the task (see page 519) deleted or delete the task (see page 519) permanently.

Task.DeleteTask Method (int)

Mark the task (see page 519) deleted.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTask", Description = "Mark the task deleted.")]
public bool DeleteTask(int taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTask", Description = "Mark the task deleted.")>
Public Function DeleteTask(taskId As Integer) As bool
```

Parameters

Parameters	Description
taskId	Task (see page 576) ID

Returns

False=if successfully deleted otherwise True

Task.DeleteTask Method (int, bool)

Mark the task (see page 519) deleted or delete the task (see page 519) permanently.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaskPermanently", Description = "Mark the task deleted or
delete the task permanently.")]
public bool DeleteTask(int 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 Integer, permanently As bool) As bool
```

Parameters

Parameters	Description
taskId	Task (see page 576) ID
Permanently	True=Delete the task (see page 519)

Returns

False=if successfully deleted otherwise True

DeleteTasks Method

Overload List

	Name	Description
	Task.DeleteTasks (int[]) (see page 581)	Mark tasks deleted.
	Task.DeleteTasks (int[], bool) (see page 582)	Mark tasks deleted or delete tasks permanently.

Task.DeleteTasks Method (int[])

Mark tasks deleted.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTasks", Description = "Mark tasks deleted.")]
public bool DeleteTasks(int[] taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTasks", Description = "Mark tasks deleted.")>
Public Function DeleteTasks(taskId As int[]) As bool
```

Parameters

Parameters	Description
taskId	Array of Task (see page 576) ID

Returns

False=if successfully deleted otherwise True

Remarks

When an error occurs the method will quite without deleting rest of the tasks.

Task.DeleteTasks Method (int[], bool)

Mark tasks deleted or delete tasks permanently.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTasksPermanently", Description = "Mark tasks deleted or delete
tasks permanently.")]
public bool DeleteTasks(int[] 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 int[], permanently As bool) As bool
```

Parameters

Parameters	Description
taskId	Array of Task (see page 576) 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 (see page 519) information.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTask", Description = "Get the task information.")]
public TaskData GetTask(int taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTask", Description = "Get the task information.")>
Public Function GetTask(taskId As Integer) As TaskData
```

Parameters

Parameters	Description
taskId	Task (see page 576) Id

Returns

Task (see page 576) 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 (see page 519) data.

GetTasksAssignedByUser Method

Overload List

	Name	Description
	Task.GetTasksAssignedByUser (int) (see page 583)	Get all tasks assigned by the user (see page 534).
	Task.GetTasksAssignedByUser (int, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 583)	Get all task (see page 519) assign by the user (see page 534) by a state of tasks specified.

Task.GetTasksAssignedByUser Method (int)

Get all tasks assigned by the user (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedByUser", Description = "Get all tasks assigned by
the user.")]
public TaskData[] GetTasksAssignedByUser(int userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedByUser", Description = "Get all tasks assigned by
the user.")>
Public Function GetTasksAssignedByUser(userId As Integer) As TaskData()
```

Parameters

Parameters	Description
`userId`	User (see page 706)'s ID

Returns

Array of tasks.

Task.GetTasksAssignedByUser Method (int, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all task (see page 519) assign by the user (see page 534) by a state of tasks specified.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedByUserByState", Description = "")]
```

```
public TaskData[] GetTasksAssignedByUser(int userId, Ektron.Cms.Common.EkEnumeration.TaskState
taskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedByUserByState", Description = "")>
Public Function GetTasksAssignedByUser(userId As Integer, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
`userId`	User (see page 706)'s ID
`taskState`	Task (see page 576) State

Returns

Array of tasks.

GetTasksAssignedToUser Method

Overload List

	Name	Description
	Task.GetTasksAssignedToUser (int) (see page 584)	Get all tasks assigned to the user (see page 534).
	Task.GetTasksAssignedToUser (int, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 584)	Get all tasks assigned to the user (see page 534) by a state of the task (see page 519) specified.

Task.GetTasksAssignedToUser Method (int)

Get all tasks assigned to the user (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUser", Description = "Get all tasks assigned to
the user.")]
public TaskData[] GetTasksAssignedToUser(int userId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUser", Description = "Get all tasks assigned to
the user.")>
Public Function GetTasksAssignedToUser(userId As Integer) As TaskData()
```

Parameters

Parameters	Description
`userId`	User (see page 706)'s ID

Returns

Array of tasks.

Task.GetTasksAssignedToUser Method (int, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all tasks assigned to the user (see page 534) by a state of the task (see page 519) 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(int 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 Integer, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
`userId`	User (see page 706)'s ID
`taskState`	Task (see page 576) State

Returns

Array of tasks.

GetTasksAssignedToUsergroup Method

Overload List

	Name	Description
	Task.GetTasksAssignedToUsergroup (int) (see page 585)	Get all tasks assigned to the usergroup.
	Task.GetTasksAssignedToUsergroup (int, Ektron.Cms.Common.EkEnumeration.TaskState) (see page 586)	Get all tasks assigned to the usergroup by a state of tasks specified.

Task.GetTasksAssignedToUsergroup Method (int)

Get all tasks assigned to the usergroup.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUsergroup", Description = "Get all tasks assigned
to the usergroup.")]
public TaskData[] GetTasksAssignedToUsergroup(int groupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUsergroup", Description = "Get all tasks assigned
to the usergroup.")>
Public Function GetTasksAssignedToUsergroup(groupId As Integer) As TaskData()
```

Parameters

Parameters	Description
`groupId`	Usergroup's ID

Returns

Array of tasks.

Task.GetTasksAssignedToUsergroup Method (int, Ektron.Cms.Common.EkEnumeration.TaskState)

Get all tasks assigned to the usergroup by a state of tasks specified.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTasksAssignedToUsergroupByState", Description = "Get all tasks
assigned to the usergroup by a state of tasks specified.")]
public TaskData[] GetTasksAssignedToUsergroup(int groupId,
Ektron.Cms.Common.EkEnumeration.TaskState taskState);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTasksAssignedToUsergroupByState", Description = "Get all tasks
assigned to the usergroup by a state of tasks specified.")>
Public Function GetTasksAssignedToUsergroup(groupId As Integer, taskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As TaskData()
```

Parameters

Parameters	Description
groupId	Usergroup's ID
taskState	Task (see page 576) State

Returns

Array of tasks.

Task.GetTaskState Method

Get the task (see page 519) state.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskState", Description = "Get the task state.")]
public Ektron.Cms.Common.EkEnumeration.TaskState GetTaskState(int taskId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskState", Description = "Get the task state.")>
Public Function GetTaskState(taskId As Integer) As Ektron.Cms.Common.EkEnumeration.TaskState
```

Parameters

Parameters	Description
taskId	Task (see page 576) ID

Returns

If successful returns task (see page 519) state otherwise it will throw an error.

Task.SetStateForTask Method

Set the task (see page 519) state.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetStateForTask", Description = "Set the task state.")]
public bool SetStateForTask(int 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 Integer, newTaskState As
Ektron.Cms.Common.EkEnumeration.TaskState) As bool
```

Parameters

Parameters	Description
taskId	Task (see page 576) ID
newTaskState	New Task (see page 576) State

Returns

False=if successfully state of the changed otherwise True

Task.UpdateTask Method

Updates the task (see page 519)

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(data As ref TaskData) As bool
```

Parameters

Parameters	Description
data	Task (see page 576) 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 588)	Adds a task (see page 519) category.
	DeleteTaskCategory (see page 588)	Deletes the task (see page 519) category.
	GetAllCategories (see page 589)	Gets All Task (see page 576) categories and task (see page 519) types belongs to those categories.
	GetTaskCategory (see page 589)	Gets the specified task (see page 519) category and task (see page 519) types belongs to the category.
	UpdateTaskCategory (see page 590)	Updates the category title.

Legend

	Method

TaskCategory.AddTaskCategory Method

Adds a task (see page 519) category.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaskCategory", Description = "Adds a task category.")]
public int AddTaskCategory(string title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaskCategory", Description = "Adds a task category.")>
Public Function AddTaskCategory(title As string) As Integer
```

Parameters

Parameters	Description
`title`	Category Title

Returns

Category Id

TaskCategory.DeleteTaskCategory Method

Deletes the task (see page 519) category.

C#

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaskCategory", Description = "Deletes the task category.")]
public bool DeleteTaskCategory(int categoryId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaskCategory", Description = "Deletes the task category.")>
Public Function DeleteTaskCategory(categoryId As Integer) As bool
```

Parameters

Parameters	Description
categoryId	category id

Returns

False=if successfully deleted otherwise True

TaskCategory.GetAllCategories Method

Gets All Task (see page 576) categories and task (see page 519) 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 (see page 519) category and task (see page 519) 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(int 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 Integer) 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(int categoryId, string title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateTaskCategory", Description = "Updates the category title.")>
Public Function UpdateTaskCategory(categoryId As Integer, title As string) As bool
```

Parameters

Parameters	Description
categoryId	Category Id
title	New Title

Returns

False=if successfully updated otherwise True

TaskCategoryType Class

Summary description for TaskCategoryType

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class TaskCategoryType : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class TaskCategoryType
Inherits EktronBaseService
```

File

TaskCategoryType.cs

TaskCategoryType Methods

The methods of the TaskCategoryType class are listed here.

Public Methods

	Name	Description
	AddTaskType (see page 591)	Adds a task (see page 519) type to a category.

	DeleteTaskType (see page 591)	Deletes the task (see page 519) type.
	GetTaskType (see page 592)	Gets the task (see page 519) type information.
	UpdateTaskType (see page 592)	Updates the task (see page 519) type.

Legend

	Method

TaskCategoryType.AddTaskType Method

Adds a task (see page 519) type to a category.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTaskType", Description = "Adds a task type to a category.")]
public int AddTaskType(string title, string description, int categoryId, bool active);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTaskType", Description = "Adds a task type to a category.")>
Public Function AddTaskType(title As string, description As string, categoryId As Integer,
active As bool) As Integer
```

Parameters

Parameters	Description
title	Title
description	Description
categoryId	Task (see page 576) Category ID
active	Set as active

Returns

Task (see page 576) Type ID

TaskCategoryType.DeleteTaskType Method

Deletes the task (see page 519) type.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTaskType", Description = "Deletes the task type.")]
public bool DeleteTaskType(int taskTypeId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTaskType", Description = "Deletes the task type.")>
Public Function DeleteTaskType(taskTypeId As Integer) As bool
```

Parameters

Parameters	Description
taskTypeId	task (see page 519) type id to be deleted.

Returns

False=if successfully deleted otherwise True

TaskCategoryType.GetTaskType Method

Gets the task (see page 519) type information.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskType", Description = "Gets a task type")]
public TaskTypeData GetTaskType(int taskTypeId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskType", Description = "Gets a task type")>
Public Function GetTaskType(taskTypeId As Integer) As TaskTypeData
```

Parameters

Parameters	Description
taskTypeId	TaskTypeId

Returns

TaskTypeData

TaskCategoryType.UpdateTaskType Method

Updates the task (see page 519) type.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateTaskType", Description = "Updates the task type.")]
public bool UpdateTaskType(int 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 Integer, title As string, description As string,
active As bool) As bool
```

Parameters

Parameters	Description
taskTypeId	Task (see page 576) 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 593)	Summary description for ActiveDirectoryUser
	Asset (see page 601)	Summary description for Asset
	Blog (see page 607)	Summary description for Blog
	BusinessRules (see page 619)	Summary description for BusinessRules
	Calendar (see page 621)	Summary description for Calendar
	CalendarEvent (see page 621)	Summary description for CalendarEvent
	CalendarEventType (see page 621)	Summary description for CalendarEventType
	Content (see page 622)	Summary description for ContentService
	ContentRating (see page 634)	Summary description for ContentRating
	CustomFields (see page 636)	Summary description for CustomFields
	Folder (see page 644)	Summary description for Folder
	Font (see page 648)	Summary description for Font
	Form (see page 651)	Summary description for Form
	Library (see page 658)	Summary description for Library
	Metadata (see page 667)	Summary description for Metadata
	Permissions (see page 671)	Summary description for Permissions
	SearchManager (see page 676)	Summary description for SearchManager
	Site (see page 681)	Summary description for Site
	SiteMap (see page 688)	Summary description for SiteMap
	Taxonomy (see page 691)	The summary description for Taxonomy.
	ThreadedDiscussion (see page 697)	Summary description for ThreadedDiscussion
	User (see page 706)	Summary description for User

Legend

	Class

ActiveDirectoryUser Class

Summary description for ActiveDirectoryUser

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class ActiveDirectoryUser : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class ActiveDirectoryUser
Inherits EktronBaseService
```

File

ActiveDirectoryUser.cs

ActiveDirectoryUser Methods

The methods of the ActiveDirectoryUser class are listed here.

Public Methods

	Name	Description
	AddADGroupToCMS (see page 594)	This function allows you add Active Directory (see page 118) groups to the CMS.
	DeSynchUserGroups (see page 595)	Updates the group with a new group name.
	DeSynchUsers (see page 595)	Update the user (see page 534) with new user (see page 534) details.
	GetADMapping (see page 596)	Loads the CMS and Active Directory (see page 118) name.
	GetADStatus (see page 596)	Used to load required Synch, Desynch and relationship flags of the configured Active Directory (see page 118).
	GetAvailableADGroups (see page 596)	Loads all of the Active Directory (see page 118) groups by a given domain and filter.
	GetAvailableADUsers (see page 597)	Loads all available Active Directory (see page 118) users.
	GetCMSGroupsToSync (see page 597)	Loads all the CMS groups that need to be synchronized.
	GetCMSRelationshipsToSync (see page 598)	Loads all the CMS relationships that need to be synchronized with Active Directory (see page 118).
	GetCMSUsersToSync (see page 598)	Loads all the CMS users that need to be synchronized with Active Directory (see page 118).
	GetDomains (see page 599)	Loads a list of domains from Active Directory (see page 118).
	MapCMSUserGroupToAD (see page 599)	Updates the given CMS group corresponding to the Active Directory (see page 118) group.
	MapCMSUserToAD (see page 600)	Maps the given user (see page 534) ID to the corresponding Active Directory (see page 118) entry.
	UpdateADMapping (see page 600)	Updates the Active directory information into the CMS based on the user (see page 534) ID and user (see page 534) data.
	UpdateADUsersGroups (see page 601)	Updates Active Directory (see page 118) users in a CMS group.
	UpdateUsersInfoFromAD (see page 601)	Updates the user (see page 534) details from the Active Directory (see page 118) details.

Legend

	Method

ActiveDirectoryUser.AddADGroupToCMS Method

This function allows you add Active Directory (see page 118) 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 118) group paths.

Remarks

A user (see page 534) 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 (see page 534) group with the new group information, otherwise none.

Returns

Ektron.Cms.AdDeSyncGroupData()

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.DeSynchUsers Method

Update the user (see page 534) with new user (see page 534) 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 (see page 534) with new user (see page 534) information, otherwise none.

Returns

Ektron.Cms.AdDeSyncUserData()

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetADMapping Method

Loads the CMS and Active Directory (see page 118) name.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetADMapping", Description = "")]
public AdMappingData[] GetADMapping(int CallerId, string MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetADMapping", Description = "")>
Public Function GetADMapping(CallerId As Integer, MapType As string, Map As Integer, Exclude
As Integer, Editable As Integer) As AdMappingData()
```

Parameters

Parameters	Description
`CallerId`	Who is calling this function.
`MapType`	Possible value is “userprop” (user (see page 534) properties).
`Map`	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
`Exclude`	Flag (0 – false, 1-true).
`Editable`	Flag(0-false,1-true).

Returns

Ektron.Cms.AdMappingData()

Remarks

A user (see page 534) 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 118).

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 (see page 534) with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetAvailableADGroups Method

Loads all of the Active Directory (see page 118) 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 (see page 534) with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetAvailableADUsers Method

Loads all available Active Directory (see page 118) 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 801)()

Remarks

A user (see page 534) 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 (see page 534) 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 118).

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 (see page 534) 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 118).

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 801)()

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.GetDomains Method

Loads a list of domains from Active Directory (see page 118).

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 118) (1 = check otherwise resumed).

Returns

Ektron.Cms.DomainData()

ActiveDirectoryUser.MapCMSUserGroupToAD Method

Updates the given CMS group corresponding to the Active Directory (see page 118) group.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MapCMSUserGroupToAD", Description = "")]
public void MapCMSUserGroupToAD(int GroupId, string GroupName, string Domain);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MapCMSUserGroupToAD", Description = "")>
Public Function MapCMSUserGroupToAD(GroupId As Integer, GroupName As string, Domain As string)
As void
```

Parameters

Parameters	Description
GroupId	CMS group ID.
GroupName	CMS group name.
Domain	The Active Directory (see page 118) domain.

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.MapCMSUserToAD Method

Maps the given user (see page 534) ID to the corresponding Active Directory (see page 118) entry.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "MapCMSUserToAD", Description = "")]
public void MapCMSUserToAD(int UserId, string UserName, string Domain, int PullInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "MapCMSUserToAD", Description = "")>
Public Function MapCMSUserToAD(UserId As Integer, UserName As string, Domain As string,
PullInfo As Integer) As void
```

Parameters

Parameters	Description
UserId	The user (see page 534)'s ID.
UserName	The user (see page 534)'s name.
Domain	The user (see page 534)'s domain.
PullInfo	Flag to load Active Directory (see page 118) information. (1 = Load Active Directory (see page 118) information, otherwise none.)

Remarks

A user (see page 534) 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 (see page 534) ID and user (see page 534) data.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateADMapping", Description = "")]
public void UpdateADMapping(int UserId, object pagedata);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateADMapping", Description = "")>
Public Function UpdateADMapping(UserId As Integer, pagedata As object) As void
```

Parameters

Parameters	Description
UserId	The user (see page 534)'s ID.
pagedata	Key, value pair for given user (see page 534) ID.

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.UpdateADUsersGroups Method

Updates Active Directory (see page 118) 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 (see page 534)'s username.
Domain	The user (see page 534)'s Domain.

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

ActiveDirectoryUser.UpdateUsersInfoFromAD Method

Updates the user (see page 534) details from the Active Directory (see page 118) 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 (see page 534)'s username.
Domain	The user (see page 534)'s Domain.

Remarks

A user (see page 534) 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 602)	Loads all of the registered plug-in configuration details.
	GetAssetMgtInfo (see page 603)	Loads all of the registered plug-in type details.
	GetAssetSupertypes (see page 603)	Loads all of the asset supertypes supported by documents add-in.
	GetCreateDragDropSnippet (see page 603)	Loads the drag and drop client snippet used with the Document Management functionality.
	GetCreateEktronExplorerSnippet (see page 604)	Loads the explorer snippet used with the Document Management functionality.
	GetCreateSnippet (see page 604)	Loads the create document snippet used with the Document Management functionality.
	GetDragDropWOSnippet (see page 604)	Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.
	GetEditSnippet (see page 605)	Loads the edit snippet for the given document. Used with the Document Management functionality.
	GetMyAssetMgtID (see page 605)	Retrieves the Document Management functionality client ID.
	MakeCopyDMSAsset (see page 606)	Create a copy of DMS Asset (see page 601).
	SetAssetMgtConfigInfo (see page 606)	Updates the Document Management configuration information.

Legend

	Method

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.GetCreateDragDropSnippet Method

Loads the drag and drop client snippet used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCreateDragDropSnippet", Description = "")]
public string GetCreateDragDropSnippet();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCreateDragDropSnippet", Description = "")>
Public Function GetCreateDragDropSnippet() As string
```

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetCreateEktronExplorerSnippet Method

Loads the explorer snippet used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCreateEktronExplorerSnippet", Description = "")]
public string GetCreateEktronExplorerSnippet();
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCreateEktronExplorerSnippet", Description = "")>
Public Function GetCreateEktronExplorerSnippet() As string
```

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetCreateSnippet Method

Loads the create document snippet used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetCreateSnippet", Description = "")]
public string GetCreateSnippet(int ContentType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetCreateSnippet", Description = "")>
Public Function GetCreateSnippet(ContentType As Integer) As string
```

Parameters

Parameters	Description
`ContentType`	The type of the content (see page 449), such as office documents or managed files.

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetDragDropWOSnippet Method

Loads the edit snippet for the given document when working offline. Used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDragDropWOSnippet", Description = "")]
public string GetDragDropWOSnippet(string AssetId, int ContentType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDragDropWOSnippet", Description = "")>
Public Function GetDragDropWOSnippet(AssetId As string, ContentType As Integer) As string
```

Parameters

Parameters	Description
AssetId	The document's ID (Required)
ContentType	The type of the content (see page 449), such as office documents or managed files or * (all).

Returns

String

Remarks

The Document Management functionality with a valid license is required to perform this operation.

Asset.GetEditSnippet Method

Loads the edit snippet for the given document. Used with the Document Management functionality.

C#

```
[SoapHeader("AuthHeader")]
[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 (see page 449), 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 601).

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 706) login with create content (see page 449) 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 706) 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 608)	Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.
	AddCommentForPost (see page 608)	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 609)	Adds a post to a given blog. Accepts a ContentData (see page 738) class as the post content (see page 449). Return new post ID.
	DeleteBlog (see page 609)	Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0.
	DeletePost (see page 610)	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 610)	Returns a blank BlogPostData class. Used Primarily in the Workarea.
	GetBlog (see page 610)	Returns a BlogData class for a given blog id, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This method is overloaded.
	GetBlogData (see page 612)	Overloaded Method to obtain a BlogData Class.
	GetBlogDataForPost (see page 615)	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 616)	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 616)	Returns a populated blog roll class. This blog roll is the list of links for the blog. See the BlogRoll class for the properties and usage of the blog roll. This is an overloaded method.
	GetPost (see page 618)	Gets a blog post for a given id. Will also return the post information separately (ByRef) in a BlogPostData class. Returns ContentData (see page 738) Class.

	GetPostbyID (see page 618)	Gets a blog post for a given id. Includes trackback, pinback, comments, and other information. Returns BlogPostData Class.

Legend

	Method

Blog.AddBlog Method

Adds a blog to the ParentFolderID. Accepts standard blog properties. Returns the ID of the newly created blog.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddBlog", Description = "")]
public int AddBlog(int ParentFolderID, string BlogName, string BlogTitle, string
BlogDescription, Ektron.Cms.Common.EkEnumeration.BlogVisibility Visibility, bool
EnableComments, bool ModerateComments, bool RequireAuthentication, string[] Categories,
BlogRoll RollLinks);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddBlog", Description = "")>
Public Function AddBlog(ParentFolderID As Integer, BlogName As string, BlogTitle As string,
BlogDescription As string, Visibility As Ektron.Cms.Common.EkEnumeration.BlogVisibility,
EnableComments As bool, ModerateComments As bool, RequireAuthentication As bool, Categories As
string[], RollLinks As BlogRoll) As Integer
```

Parameters

Parameters	Description
ParentFolderID	
BlogName	Name of the Blog (see page 607).
BlogTitle	Title of the Blog (see page 607).
BlogDescription	Description of the Blog (see page 607).
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 607)'s roll links.

Returns

Integer value of the blog ID that is greater than 0.

Blog.AddCommentForPost Method

Adds a post comment to a blog post. Accepts a BlogComment class as the input. Will return the result of that add action. False indicates the action was successful.

C#

```
[SoapHeader("AuthHeader")]
[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 738) class as the post content (see page 449). Return new post ID.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddPost", Description = "")]
public int AddPost(int 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 Integer, PostContent As ContentData, Categories As string[],
Pingback As bool, Tags As string, TrackbackURL As string) As Integer
```

Parameters

Parameters	Description
blogID	Integer ID of the blog to add the post to.
PostContent	ContentData (see page 738) class representing the HTML, title, etc.
Categories	String Array representing the categories this post applies to.
Pingback	Pingback URL(s) in this post?
Tags	String list of tags that that applies to this post.
TrackbackURL	Trackback URL(s) to notify.

Returns

Integer ID of the post created.

Blog.DeleteBlog Method

Deletes a blog with a given ID. This will delete the blog as well as all posts and comments for the blog. BlogID must be greater than 0.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteBlog", Description = "")]
public void DeleteBlog(int BlogID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteBlog", Description = "")>
Public Function DeleteBlog(BlogID As Integer) As void
```

Parameters

Parameters	Description
BlogID	The integer ID of the blog. This corresponds to the folder id of the blog.

Blog.DeletePost Method

Deletes a post by the post's ID This method will delete all comments associated with the post as well. Returns boolean value.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeletePost", Description = "")]
public bool DeletePost(int post_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeletePost", Description = "")>
Public Function DeletePost(post_id As Integer) As bool
```

Parameters

Parameters	Description
post_id	Integer value of the post ID you wish to delete.

Returns

Boolean

Blog.GetBlankBlogPostData Method

Returns a blank BlogPostData class. Used Primarily in the Workarea.

C#

```
[SoapHeader("AuthHeader")]
[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

Returns a BlogData class for a given blog id, with parameters. This class contains the content for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This method is overloaded.

Overload List

	Name	Description
	Blog.GetBlog (int, string, int, int) (see page 611)	Returns a BlogData class for a given blog id, with parameters. This class contains the content (see page 449) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This function is overloaded. See Also: Blog.GetBlog Method (string, string, int, int) (see page 612)
	Blog.GetBlog (string, string, int, int) (see page 612)	Returns a BlogData class for a given path, with parameters. This class contains the content (see page 449) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information. This function is overloaded. See Also: Ektron.Services.Blog.GetBlog@int@string@int@int (see page 611)

Blog.GetBlog Method (int, string, int, int)

Returns a BlogData class for a given blog id, with parameters. This class contains the content (see page 449) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

This function is overloaded. See Also:

Blog.GetBlog Method (string, string, int, int) (see page 612)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlog by ID, StartDate, MaxEntries and LanguageID", Description =
"")]
public BlogData GetBlog(int ID, string StartDate, int MaxEntries, int LanguageID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlog by ID, StartDate, MaxEntries and LanguageID", Description =
"")>
Public Function GetBlog(ID As Integer, StartDate As string, MaxEntries As Integer, LanguageID
As Integer) As BlogData
```

Parameters

Parameters	Description
`ID`	The id of the blog. This is a integer value that corresponds to the folder id of the blog.
`StartDate`	The date to get entries from. Used if you want to get historical information. Gets information only for the date specified.
`MaxEntries`	The total number of posts to return. This is an integer value. If set to 0, the `postsvisible` property of the blog is used. If set to -1, the content (see page 449) is not returned, only the post info for the content (see page 449).
`LanguageID`	The language ID for the content (see page 449) to get. An integer value.

Returns

For a given blog id, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

Remarks

Developers are advised to check for null prior to using.

Blog.GetBlog Method (string, string, int, int)

Returns a BlogData class for a given path, with parameters. This class contains the content (see page 449) for a blog, the post info, the roll, and the categories information. See the BlogData class for more information.

This function is overloaded. See Also:

Ektron.Services.Blog.GetBlog@int@string@int@int (see page 611)

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. This is an integer value. If set to 0, the `postsvisible` property of the blog is used. If set to -1, the content (see page 449) is not returned, only the post info for the content (see page 449).
`LanguageID`	The language ID for the content (see page 449) to get. An integer value.

Returns

For a given path, this returns either null or a BlogData class with the associated posts. This also includes the roll, categories, and post information.

Remarks

Developers are advised to check for null prior to using.

GetBlogData Method

Overloaded Method to obtain a BlogData Class.

Overload List

	Name	Description
	Blog.GetBlogData (FolderData) (see page 613)	Returns the corresponding blog (if applicable) for a complete folderData object. If there is code that already leverages our FolderData (see page 755) class, this can piggyback atop it. **Returns only if the folder is a blog.** This function is overloaded. See Also: Ektron.Services.Blog.GetBlogData@int (see page 614) Ektron.Services.Blog.GetBlogData@int@DateTime (see page 614) Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 615)

	Blog.GetBlogData (int) (see page 614)	Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog. Returns either a populated BlogData class or null (nothing). This function is overloaded. See Also: Ektron.Services.Blog.GetBlogData@FolderData (see page 613) Ektron.Services.Blog.GetBlogData@int@DateTime (see page 614) Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 615)
	Blog.GetBlogData (int, DateTime) (see page 614)	Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing). This function is overloaded. See Also: Ektron.Services.Blog.GetBlogData@FolderData (see page 613) Ektron.Services.Blog.GetBlogData@int (see page 614) Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 615)
	Blog.GetBlogData (int, DateTime, int, int) (see page 615)	Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing). This function is overloaded. See Also: Ektron.Services.Blog.GetBlogData@FolderData (see page 613) Ektron.Services.Blog.GetBlogData@int (see page 614) Ektron.Services.Blog.GetBlogData@int@DateTime (see page 614)

Blog.GetBlogData Method (FolderData)

Returns the corresponding blog (if applicable) for a complete folderData object. If there is code that already leverages our FolderData (see page 755) class, this can piggyback atop it. **Returns only if the folder is a blog.**

This function is overloaded. See Also:

Ektron.Services.Blog.GetBlogData@int (see page 614)

Ektron.Services.Blog.GetBlogData@int@DateTime (see page 614)

Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 615)

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 (see page 755) class. (see FolderData (see page 755))

Returns

A populated BlogData class or nothing

Remarks

Blog.GetBlogData Method (int)

Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

Ektron.Services.Blog.GetBlogData@FolderData (see page 613)

Ektron.Services.Blog.GetBlogData@int@DateTime (see page 614)

Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 615)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData By ID", Description = "")]
public BlogData GetBlogData(int BlogID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData By ID", Description = "")>
Public Function GetBlogData(BlogID As Integer) As BlogData
```

Parameters

Parameters	Description
`BlogID`	Integer ID of the blog. This corresponds to the id of the blog folder.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogData Method (int, DateTime)

Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog, as well as the StartDate (as DateTime) for the posts. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

Ektron.Services.Blog.GetBlogData@FolderData (see page 613)

Ektron.Services.Blog.GetBlogData@int (see page 614)

Ektron.Services.Blog.GetBlogData@int@DateTime@int@int (see page 615)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData By ID and StartDate", Description = "")]
public BlogData GetBlogData(int BlogID, DateTime StartDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData By ID and StartDate", Description = "")>
Public Function GetBlogData(BlogID As Integer, StartDate As DateTime) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog. This corresponds to the id of the blog folder.
StartDate	The DateTime value to get posts from. You can obtain historical posts with this property, or leave it blank to begin now.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogData Method (int, DateTime, int, int)

Overloaded Method to obtain a BlogData Class. Accepts the integer ID of the blog, the StartDate (as DateTime) for the posts, the number of posts visible and the language ID. Returns either a populated BlogData class or null (nothing).

This function is overloaded. See Also:

Ektron.Services.Blog.GetBlogData@FolderData (see page 613)

Ektron.Services.Blog.GetBlogData@int (see page 614)

Ektron.Services.Blog.GetBlogData@int@DateTime (see page 614)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogData by ID, StartDate, PostsVisible and LanguageID",
Description = "")]
public BlogData GetBlogData(int BlogID, DateTime StartDate, int Postsvisible, int LanguageID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogData by ID, StartDate, PostsVisible and LanguageID",
Description = "")>
Public Function GetBlogData(BlogID As Integer, StartDate As DateTime, Postsvisible As Integer,
LanguageID As Integer) As BlogData
```

Parameters

Parameters	Description
BlogID	Integer ID of the blog. This corresponds to the id of the blog folder.
StartDate	The DateTime value to get posts from. You can obtain historical posts with this property, or leave it blank to begin now.
Postsvisible	Integer number of posts to return. Leave as 0 to use the blog setting, or -1 to return none.
LanguageID	Integer ID of the language.

Returns

Either a populated BlogData class or null (nothing).

Remarks

Blog.GetBlogDataForPost Method

Returns the blog information for a blog post. This contains information about categories, whether comments are enabled, etc. Accepts the post ID.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogDataForPost", Description = "")]
public BlogData GetBlogDataForPost(int post_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogDataForPost", Description = "")>
Public Function GetBlogDataForPost(post_id As Integer) As BlogData
```

Parameters

Parameters	Description
post_id	Integer value of the post ID

Returns

BlogData class

Blog.GetBlogPostData Method

Gets the Post data for a specific blog post. Accepts the ID of the post. Will return tags, trackback, pingback, and categories for the post.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogPostData", Description = "")]
public BlogPostData GetBlogPostData(int id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogPostData", Description = "")>
Public Function GetBlogPostData(id As Integer) As BlogPostData
```

Parameters

Parameters	Description
id	Integer value of the post ID

Returns

BlogPostData Class.

GetBlogRoll Method

Returns a populated blog roll class. This blog roll is the list of links for the blog. See the BlogRoll class for the properties and usage of the blog roll.

This is an overloaded method.

Overload List

	Name	Description
	Blog.GetBlogRoll (int) (see page 617)	Returns a populated blog roll class. Accepts a blog ID integer value. See the blogroll class for more information. This function is overloaded. See Also: Ektron.Services.Blog.GetBlogRoll@string (see page 617)

	Blog.GetBlogRoll (string) (see page 617)	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 46) class for the properties and usage of the blog roll. This is an overloaded method. See Also: Ektron.Services.Blog.GetBlogRoll@int (see page 617).

Blog.GetBlogRoll Method (int)

Returns a populated blog roll class. Accepts a blog ID integer value. See the blogroll class for more information.

This function is overloaded. See Also:

Ektron.Services.Blog.GetBlogRoll@string (see page 617)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBlogRoll", Description = "")]
public BlogRoll GetBlogRoll(int blog_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBlogRoll", Description = "")>
Public Function GetBlogRoll(blog_id As Integer) As BlogRoll
```

Parameters

Parameters	Description
`blog_id`	Integer ID of the blog. This corresponds to the folder ID of the blog.

Returns

A populated BlogRoll (see page 46) 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 46) class for the properties and usage of the blog roll.

This is an overloaded method. See Also:

Ektron.Services.Blog.GetBlogRoll@int (see page 617).

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 46) 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 738) Class.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPost", Description = "")]
public ContentData GetPost(int post_id, BlogPostData PostInfo);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPost", Description = "")>
Public Function GetPost(post_id As Integer, PostInfo As BlogPostData) As ContentData
```

Parameters

Parameters	Description
post_id	Integer ID for the post.
PostInfo	BlogPostData class representing the blog specific post info. This is also in the metadata.

Returns

ContentData (see page 738) 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(int post_id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPostbyID", Description = "")>
Public Function GetPostbyID(post_id As Integer) As BlogPostData
```

Parameters

Parameters	Description
post_id	Integer ID for the post.

Returns

BlogPostData class

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 619)	Method to get the results from a business rule. This method is overloaded.

Legend

	Method

GetBusinessRules Method

Method to get the results from a business rule. This method is overloaded.

Overload List

	Name	Description
	BusinessRules.GetBusinessRules (string, string, string, string, string, string, string, int) (see page 619)	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 620)	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
`formValues`	Form (see page 651) Value Collection
`queryValues`	Querystring Value Collection
`serverValues`	Server Variables Collection
`paramValues`	Custom Value Collection
`RuleID`	Integer representing the ID of the Ruleset

Returns

A string representing the output of the evaluated ruleset.

BusinessRules.GetBusinessRules Method (string, string, string, string, string, string, string, string)

Method to get the results from a business rule by identifier Accepts environmental parameters such as HTTPHost, URLString, Cookies, forms, etc Returns a string representing the output of the evaluated ruleset. Identifier is a combination of the path/page/uniqueID of the business rules control.

C#

```
[SoapHeader("AuthHeader")]
[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
`formValues`	Form (see page 651) Value Collection
`queryValues`	Querystring Value Collection
`serverValues`	Server Variables Collection

`paramValues`	Custom Value Collection
`RuleID`	String representing the identifier of the Ruleset. The identifier is a combination of the path/page/uniqueID of the business rules control.

Returns

A string representing the output of the evaluated ruleset.

Calendar Class

Summary description for Calendar

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Calendar : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class Calendar
Inherits EktronBaseService
```

File

Calendar.cs

CalendarEvent Class

Summary description for CalendarEvent

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class CalendarEvent : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class CalendarEvent
Inherits EktronBaseService
```

File

CalendarEvent.cs

CalendarEventType Class

Summary description for CalendarEventType

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class CalendarEventType : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class CalendarEventType
Inherits EktronBaseService
```

File

CalendarEventType.cs

Content Class

Summary description for ContentService

Class Hierarchy

C#

```
public class Content : EktronBaseService;
```

Visual Basic

```
Public Class Content
Inherits EktronBaseService
```

File

Content.cs

Description

[WebService(Namespace = "http://ektron.com/webservices/")] [WebServiceBinding(ConformsTo = WsiProfiles.None)]

Content Methods

The methods of the Content class are listed here.

Public Methods

	Name	Description
	AddContent (see page 623)	Adds Content to the CMS. This method is overloaded.
	CheckOutContent (see page 625)	This will check the content (see page 449) out to the current user (see page 534) if the content (see page 449) is currently checked in and the user (see page 534) has privileges to check it out.
	CopyContentToFolder (see page 626)	Copy the content (see page 449) into required folder.
	DeleteContentItem (see page 626)	Delete the content (see page 449) by the content (see page 449)'s ID.
	GetAddViewLanguage (see page 627)	Loads all the languages from the system for a given content (see page 449) ID and populates the Add/View drop down box with flag set on each item.
	GetAllComments (see page 627)	Loads all of the comments.
	GetAllTemplates (see page 628)	Loads all of the templates.
	GetAllUnassignedItemApprovals (see page 628)	Returns all of the unassigned user (see page 534)/groups for a given content (see page 449) ID.
	GetChildContent (see page 628)	Loads all of the contents for the given folder.

	GetContent (see page 629)	Get Content (see page 622) By ID
	GetContentByHistory (see page 629)	Retrieves the corresponding content (see page 449) by the history ID.
	GetContentForEditing (see page 629)	Loads the content (see page 449) details with checkout mode.
	GetContentState (see page 630)	Loads the content (see page 449)'s details.
	GetContentStatus (see page 630)	This will return a single letter string which is the status of the piece of content (see page 449). The letters returned are the same letters used in the Workarea view.
	GetDomain (see page 631)	Gets web domain for a piece of content (see page 449).
	MoveContentToFolder (see page 631)	Moves content (see page 449) to a folder.
	PublishContent (see page 631)	Function takes in information about the content (see page 449) to be checked in and published. This goes through the normal approval chain. The content (see page 449) must be in a checked out state to the current logged in user (see page 534) for this to succeed.
	RenameContent (see page 632)	Renames the content (see page 449).
	SaveContent (see page 632)	Updates the checked out content (see page 449) 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 633)	Delete the content (see page 449) by the content (see page 449)'s Id and folder Id.
	UndoCheckout (see page 633)	This undoes the checkout for a piece of content (see page 449) which is checked out to the current user (see page 534).
	UpdateContentMetaData (see page 633)	Updates a specific metadata item attached to a specified piece of content (see page 449)

Legend

	Method

AddContent Method

Adds Content to the CMS. This method is overloaded.

Overload List

	Name	Description
	Content.AddContent (string, string, string, string, string, string, int, object, object, string) (see page 623)	Adds Content (see page 622) to the CMS
	Content.AddContent (string, string, string, string, string, string, int, object, object, string, int) (see page 624)	Adds Content (see page 622) to the CMS
	Content.AddContent (string, string, string, string, string, string, int, object, object, string, int, int) (see page 625)	Adds Content (see page 622) to the CMS

Content.AddContent Method (string, string, string, string, string, string, int, object, object, string)

Adds Content (see page 622) to the CMS

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContent", Description = "Adds a piece of content.")]
public int AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, int FolderId, object GoLive, object
EndDate, string MetaInfoXml);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
Integer, GoLive As object, EndDate As object, MetaInfoXml As string) As Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 449) to add
ContentComment	Content (see page 622) comments
ContentHtml	Content (see page 622) HTML to display
SearchText	Search (see page 514) Text
SummaryHtml	Summary HTML
ContentLanguage	Content (see page 622) Language
FolderId	Folder (see page 644) to place the content (see page 449) in
GoLive	Go Live Date
EndDate	Content (see page 622) End Date
MetaInfoXml	Meta Info XML

Returns

Integer - ID of newly created content (see page 449)

Content.AddContent Method (string, string, string, string, string, string, int, object, object, string, int)

Adds Content (see page 622) to the CMS

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContent With ContentID", Description = "Adds a piece of
content.")]
public int AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, int FolderId, object GoLive, object
EndDate, string MetaInfoXml, int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
Integer, GoLive As object, EndDate As object, MetaInfoXml As string, ContentID As Integer) As
Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 449) to add
ContentComment	Content (see page 622) comments

ContentHtml	Content (see page 622) HTML to display
SearchText	Search (see page 514) Text
SummaryHtml	Summary HTML
ContentLanguage	Content (see page 622) Language
FolderId	Folder (see page 644) to place the content (see page 449) in
GoLive	Go Live Date
EndDate	Content (see page 622) End Date
MetaInfoXml	Meta Info XML
ContentID	Content (see page 622) ID

Returns

Integer - ID of newly created content (see page 449)

Content.AddContent Method (string, string, string, string, string, string, int, object, object, string, int, int)

Adds Content (see page 622) to the CMS

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContent with XmlID and TemplateID", Description = "Adds a piece
of content.")]
public int AddContent(string ContentTitle, string ContentComment, string ContentHtml, string
SearchText, string SummaryHtml, string ContentLanguage, int FolderId, object GoLive, object
EndDate, string MetaInfoXml, int XmlId, int TemplateID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<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
Integer, GoLive As object, EndDate As object, MetaInfoXml As string, XmlId As Integer,
TemplateID As Integer) As Integer
```

Parameters

Parameters	Description
ContentTitle	Title of content (see page 449) to add
ContentComment	Content (see page 622) comments
ContentHtml	Content (see page 622) HTML to display
SearchText	Search (see page 514) Text
SummaryHtml	Summary HTML
ContentLanguage	Content (see page 622) Language
FolderId	Folder (see page 644) to place the content (see page 449) in
GoLive	Go Live Date
EndDate	Content (see page 622) End Date
XmlId	XmlId
TemplateID	TemplateID

Returns

Integer - ID of newly created content (see page 449)

Content.CheckOutContent Method

This will check the content (see page 449) out to the current user (see page 534) if the content (see page 449) is currently checked in and the user (see page 534) 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(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CheckOutContent", Description = "Checks the content out to the
current user")>
Public Function CheckOutContent(ContentID As Integer) As bool
```

Parameters

Parameters	Description
ContentId	Id of the content (see page 449) to check out

Returns

A Boolean representing the success of the check out

Content.CopyContentToFolder Method

Copy the content (see page 449) 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 622) ID
FID	Folder (see page 644) ID
Lang	Language ID

Returns

Ektron.Cms.ContentData (see page 738)

Content.DeleteContentItem Method

Delete the content (see page 449) by the content (see page 449)'s ID.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteContentItem", Description = "Delete the content by the
content's ID")]
```

```
public bool DeleteContentItem(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteContentItem", Description = "Delete the content by the
content's ID")>
Public Function DeleteContentItem(ContentID As Integer) As bool
```

Parameters

Parameters	Description
ContentID	The content (see page 449)'s ID

Content.GetAddViewLanguage Method

Loads all the languages from the system for a given content (see page 449) 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(int 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 Integer) As LanguageData()
```

Parameters

Parameters	Description
ContentID	The Content (see page 622)'s ID

Content.GetAllComments Method

Loads all of the comments.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllComments", Description = "Gets all comments")]
public CommentData[] GetAllComments(int KeyID, int CommentID, int RefID, string RefType, int
UserID, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllComments", Description = "Gets all comments")>
Public Function GetAllComments(KeyID As Integer, CommentID As Integer, RefID As Integer,
RefType As string, UserID As Integer, OrderBy As string) As CommentData()
```

Parameters

Parameters	Description
KeyID	key ID
CommentID	Comment ID
RefID	Reference ID, (content (see page 449) or forms)
RefType	Reference type, C = content (see page 449) or T = Task (see page 576)

UserID	The user (see page 534) 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 (see page 534)/groups for a given content (see page 449) 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(int 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 Integer, ItemType As string) As
ApprovalData()
```

Parameters

Parameters	Description
ContentID	The content (see page 449)'s ID
ItemType	The type of user (see page 534)

Content.GetChildContent Method

Loads all of the contents 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(int 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 Integer, Recursive As bool, OrderBy As string) As
ContentData()
```

Parameters

Parameters	Description
FolderID	the folders ID
Recursive	True = recursive look up. False = none
OrderBy	An order by clause

Content.GetContent Method

Get Content (see page 622) By ID

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContent", Description = "Gets the content details by content id")]
public ContentData GetContent(int ContentID, Ektron.Cms.Content.EkContent.ContentResultType
ResultType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContent", Description = "Gets the content details by content id")>
Public Function GetContent(ContentID As Integer, ResultType As
Ektron.Cms.Content.EkContent.ContentResultType) As ContentData
```

Parameters

Parameters	Description
ContentID	Content (see page 622) ID
Preview	Preview
Published	Is Published

Content.GetContentByHistory Method

Retrieves the corresponding content (see page 449) by the history ID.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentByHistory", Description = "Retrieves the corresponding
content by the history ID")]
public ContentData GetContentByHistory(int HistoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentByHistory", Description = "Retrieves the corresponding
content by the history ID")>
Public Function GetContentByHistory(HistoryID As Integer) As ContentData
```

Parameters

Parameters	Description
HistoryID	The history ID

Content.GetContentForEditing Method

Loads the content (see page 449) details with checkout mode.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentForEditing", Description = "Loads the content details with
checkout mode")]
public ContentEditData GetContentForEditing(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentForEditing", Description = "Loads the content details with
checkout mode")>
Public Function GetContentForEditing(ContentID As Integer) As ContentEditData
```

Parameters

Parameters	Description
ContentID	The content (see page 449)'s ID

Content.GetContentState Method

Loads the content (see page 449)'s details.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentState", Description = "Gets the content's details")]
public ContentStateData GetContentState(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentState", Description = "Gets the content's details")>
Public Function GetContentState(ContentID As Integer) As ContentStateData
```

Parameters

Parameters	Description
ContentID	The content (see page 449)'s ID

Content.GetContentStatus Method

This will return a single letter string which is the status of the piece of content (see page 449). 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(int 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 Integer) As string
```

Parameters

Parameters	Description
ContentID	Id of the content (see page 449) to return the status of

Content.GetDomain Method

Gets web domain for a piece of content (see page 449).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDomain", Description = "Gets web domain for a piece of content")]
public string GetDomain(int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDomain", Description = "Gets web domain for a piece of content")>
Public Function GetDomain(ContentID As Integer) As string
```

Parameters

Parameters	Description
ContentID	The content (see page 449)'s ID

Content.MoveContentToFolder Method

Moves content (see page 449) 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 (see page 449)'s ID
ContentLanguage	The content (see page 449)'s language
FolderID	The folder's ID

Content.PublishContent Method

Function takes in information about the content (see page 449) to be checked in and published. This goes through the normal approval chain. The content (see page 449) must be in a checked out state to the current logged in user (see page 534) 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(int ContentID, int FolderID, int ContentLanguage, string
DontCreateTask, int 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 Integer, FolderID As Integer, ContentLanguage As
Integer, DontCreateTask As string, UserID As Integer, TaskTitle As string) As bool
```

Parameters

Parameters	Description
ContentID	Integer value of the content (see page 449)'s ID
FolderID	Integer value of the content (see page 449) folder's ID
ContentLanguage	Integer value of the content (see page 449)'s language ID
DontCreateTask	If this parameter is "" a task (see page 519) will be created for this submit, else it will not
UserID	The user (see page 534) ID of task (see page 519) assigned by (current user (see page 534)). This is needed only if "dontCreateTask" is "", it is ignored else
TaskTitle	Title for the task (see page 519). This is needed only if "dontCreateTask" is "", it is ignored else

Content.RenameContent Method

Renames the content (see page 449).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenameContent", Description = "Renames the content by ID")]
public ContentData RenameContent(int ContentID, string Title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RenameContent", Description = "Renames the content by ID")>
Public Function RenameContent(ContentID As Integer, Title As string) As ContentData
```

Parameters

Parameters	Description
ContentID	The content (see page 449)'s ID
Title	The content (see page 449)'s new title

Content.SaveContent Method

Updates the checked out content (see page 449) 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 622) must be in the checkedout state.

Content.SubmitForDelete Method

Delete the content (see page 449) by the content (see page 449)'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(int ContentID, int 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 Integer, FolderID As Integer) As bool
```

Parameters

Parameters	Description
ContentID	The content (see page 449)'s Id
FolderID	The content (see page 449)'s Folder (see page 644) Id

Content.UndoCheckout Method

This undoes the checkout for a piece of content (see page 449) which is checked out to the current user (see page 534).

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(int 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 Integer) As bool
```

Parameters

Parameters	Description
ContentID	Id of the content (see page 449) for which to undo check out

Content.UpdateContentMetaData Method

Updates a specific metadata item attached to a specified piece of content (see page 449)

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "UpdateContentMetaData", Description = "Updates Content Metadata")]
public bool UpdateContentMetaData(int ContentID, int MetaDataTypeID, string MetaDataText);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateContentMetaData", Description = "Updates Content Metadata")>
Public Function UpdateContentMetaData(ContentID As Integer, MetaDataTypeID As Integer,
MetaDataText As string) As bool
```

ContentRating Class

Summary description for ContentRating

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class ContentRating : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)>
Public Class ContentRating
Inherits EktronBaseService
```

File

ContentRating.cs

ContentRating Methods

The methods of the ContentRating class are listed here.

Public Methods

	Name	Description
	AddContentRating (see page 634)	Adds a content (see page 449) rating to the database. Internally a check is done to ensure the user (see page 534) has not already rated the content (see page 449), and passes down the rating and any comments the user (see page 534) supplied.
	GetContentRatingResults (see page 635)	Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 449) 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 635)	Purges all ratings of a particular content (see page 449) block over a defined date/time range.

Legend

	Method

ContentRating.AddContentRating Method

Adds a content (see page 449) rating to the database. Internally a check is done to ensure the user (see page 534) has not already rated the content (see page 449), and passes down the rating and any comments the user (see page 534) supplied.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddContentRating", Description = "")]
public void AddContentRating(int contentid, int userid, int userRating, string userComments);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddContentRating", Description = "")>
Public Function AddContentRating(contentid As Integer, userid As Integer, userRating As
Integer, userComments As string) As void
```

Parameters

Parameters	Description
contentid	The numeric ID associated with the content (see page 449) block being rated.
userid	The unique global identifier stored as a cookie by the user (see page 534).
userRating	The rating the user (see page 534) supplies (1-9)
userComments	Optional comments that the user (see page 534) provides.

ContentRating.GetContentRatingResults Method

Returns a weakly typed dataset (Username - String, RatingDate - DateTime, Rating - int, Comments - String) representing ratings on particular content (see page 449) 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(int contentID, DateTime startDate, DateTime
endDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentRatingResults", Description = "")>
Public Function GetContentRatingResults(contentID As Integer, startDate As DateTime, endDate
As DateTime) As System.Data.DataSet
```

Parameters

Parameters	Description
contentID	The numeric ID associated with the content (see page 449) block whose ratings are being queried.
startDate	The start date of a range of dates to query over.
endDate	The end date of a range of dates to query over.

Returns

(Username - String, RatingDate - DateTime, Rating - int, Comments - String)

ContentRating.PurgeContentRatings Method

Purges all ratings of a particular content (see page 449) block over a defined date/time range.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "PurgeContentRatings", Description = "")]
public void PurgeContentRatings(int contentid, DateTime startDate, DateTime endDate);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "PurgeContentRatings", Description = "")>
Public Function PurgeContentRatings(contentid As Integer, startDate As DateTime, endDate As
DateTime) As void
```

Parameters

Parameters	Description
contentid	The ID of the content (see page 449) that is having its ratings purged.
startDate	The beginning of the date range of content (see page 449) to be purged.
endDate	The end of the date range of content (see page 449) 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 637)	Returns the HTML string for the add/edit custom fields.
	BoolToYesNo (see page 638)	Returns labels for yes or no based on locale.

	GetEditableCustomFieldAssignments (see page 638)	Returns the editable custom fields. Break Inheritance checkbox: • Visible if not at root. • Checked if any fields assigned without inheritance. • If unchecked & form saved, removes all assignments. • If checked & form saved and no assignments chosen, creates NULL assignment (CustomFieldID -1). • When checked, enables all assignment checkboxes (iterate through all of this tables elements). • When unchecked, disables all assignment checkboxes (iterate through all of this tables elements). • Assignment Checkboxes: • When clicked: if 'Assignment' button -> going unchecked, then ensure required us unchecked. • When clicked: if 'Required' button, don't accept checks if assigned is unchecked. • When form submitted, copy checked... more (see page 638)
	GetEnhancedMetadataArea (see page 639)	Returns the HTML needed to render the EnhancedMetadata selection area.
	GetEnhancedMetadataScript (see page 639)	Returns support javascript code for enhanced metadata.
	GetFieldsByFolder (see page 640)	Returns all of the custom attributes by folder.
	GetSearchProperties (see page 640)	Returns an HTML string for the search (see page 514) properties display.
	ProcessCustomFields (see page 640)	Process Folders' Custom-Field (and metadata) assignments: When form submitted, copy checked items to hidden field. Form (see page 651) 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 641)	Returns the search (see page 514) result based on assets.
	WriteFilteredMetadataForEdit (see page 641)	Returns an HTML string for filtered metadata.
	WriteFilteredMetadataForView (see page 642)	Returns an HTML string for metadata view.
	WriteLibrarySearchExtended (see page 642)	Returns an HTML string for the library search (see page 514) custom metadata properties.
	WriteMetadataDefaultForEdit (see page 643)	Returns an HTML string for the default metadata.
	WriteMetadataTypeForView (see page 643)	Returns a metadata type for view.
	WritePublicSearch (see page 643)	Returns WritepublicsearchEx.

Legend

	Method

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(int parentfolderid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddNewEditableCustomFieldAssignments", Description = "")>
Public Function AddNewEditableCustomFieldAssignments(parentfolderid As Integer) As string
```

Parameters

Parameters	Description
parentfolderid	The parent folder's ID.

Returns

String

CustomFields.BoolToYesNo Method

Returns labels for yes or no based on locale.

C#

```
[SoapHeader("AuthHeader")]
[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 651) 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 (see page 534).

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, long LanguageID);
```

Visual Basic

```
Public Function GetFieldsByFolder(FolderID As long, LanguageID As long) As Collection
```

Parameters

Parameters	Description
FolderID	The folder's ID.
LanguageID	The language's ID.

Returns

Microsoft.VisualBasic.Collection

CustomFields.GetSearchProperties Method

Returns an HTML string for the search (see page 514) 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 (see page 514) 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 651) 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 (see page 514) 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 514) 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 667) 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 667) types.
FolderId	The folder's Id.
SearchableOnly	True or False

Returns

String

CustomFields.WriteLibrarySearchExtended Method

Returns an HTML string for the library search (see page 514) custom metadata properties.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "WriteLibrarySearchExtended", Description = "")]
public StringBuilder WriteLibrarySearchExtended(int folderId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "WriteLibrarySearchExtended", Description = "")>
Public Function WriteLibrarySearchExtended(folderId As Integer) As StringBuilder
```

Parameters

Parameters	Description
folderId	The folder's ID.

Returns

System.Text.StringBuilder

CustomFields.WriteMetadataDefaultForEdit Method

Returns an HTML string for the default metadata.

C#

```
public StringBuilder WriteMetadataDefaultForEdit(ContentMetaData MetadataType, int DataType,
Hashtable metaPermsObj);
```

Visual Basic

```
Public Function WriteMetadataDefaultForEdit(MetadataType As ContentMetaData, DataType As
Integer, metaPermsObj As Hashtable) As StringBuilder
```

Parameters

Parameters	Description
DataType	Data types
cMetadataType	Metadata (see page 667) 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 514) asset data. (Required)

Returns

System.Text.StringBuilder

Folder Class

Summary description for Folder

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Folder : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Folder
Inherits EktronBaseService
```

File

Folder.cs

Folder Methods

The methods of the Folder class are listed here.

Public Methods

	Name	Description
	AddFolder (see page 645)	Adds a folder.
	CreateFolder (see page 645)	Creates a folder.
	DeleteFolder (see page 645)	Deletes folder by folder id
	GetChildFolders (see page 646)	Loads all the subfolders for a given folder ID.
	GetFolder (see page 646)	Retrieves the folder's details.
	GetFolderID (see page 646)	
	GetPath (see page 647)	Find's a parent ID by using a given folder ID.
	RenameFolder (see page 648)	Renames folder to specified title
	UpdateFolder (see page 648)	Updates a folder.

Legend

	Method

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(FolderReq As ref FolderRequest) As void
```

Parameters

Parameters	Description
FolderReq	Folder (see page 644) 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(int FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteFolder", Description = "Deletes folder specified by folder
id")>
Public Function DeleteFolder(FolderID As Integer) As bool
```

Parameters

Parameters	Description
FolderID	Folder (see page 644) ID

Folder.GetChildFolders Method

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(int FolderID, bool Recursive, string 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 Integer, Recursive As bool, OrderBy As string) As
FolderData()
```

Parameters

Parameters	Description
FolderID	The folder ID
Recursive	True = recursive look up. False = none
OrderBy	An order by clause

Folder.GetFolder Method

Retrieves the folder's details.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolder", Description = "Retrieves the folder's details")]
public FolderData GetFolder(int FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolder", Description = "Retrieves the folder's details")>
Public Function GetFolder(FolderID As Integer) As FolderData
```

Parameters

Parameters	Description
FolderID	The folder's ID

GetFolderID Method

Overload List

	Name	Description
	Folder.GetFolderID (int) (see page 647)	Finds the folder ID by using the a content (see page 449)'s ID.

	Folder.GetFolderID (string, int) (see page 647)	Finds the folder ID by folder name and in a parent folder specified.

Folder.GetFolderID Method (int)

Finds the folder ID by using the a content (see page 449)'s ID.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderID", Description = "Finds the folder ID by using the a
content's ID")]
public int GetFolderID(int 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 Integer) As Integer
```

Parameters

Parameters	Description
ContentID	The content (see page 449)'s ID

Folder.GetFolderID Method (string, int)

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 int GetFolderID(string FolderName, int 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 Integer) As Integer
```

Parameters

Parameters	Description
FolderName	Name of the folder
ParentId	Look in a folder

Returns

Returns the folder id. Returns 0 if it does not exist.

Folder.GetPath Method

Find's a parent ID by using a given folder ID.

C#

```
[SoapHeader("AuthHeader")]
```

```
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetPath", Description = "Gets path to folder by folder id")]
public string GetPath(int FolderID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetPath", Description = "Gets path to folder by folder id")>
Public Function GetPath(FolderID As Integer) As string
```

Parameters

Parameters	Description
FolderID	The folder's ID

Folder.RenameFolder Method

Renames folder to specified title

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "RenameFolder", Description = "Renames the folder by id")]
public bool RenameFolder(int FolderID, string Title);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "RenameFolder", Description = "Renames the folder by id")>
Public Function RenameFolder(FolderID As Integer, Title As string) As bool
```

Parameters

Parameters	Description
FolderID	The Folder (see page 644)'s ID
Title	New Title

Folder.UpdateFolder Method

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 644) 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 649)	Adds a new font to the system.
	DeleteFont (see page 649)	Deletes a font from the system.
	GetAllFonts (see page 650)	Loads all of the fonts as a array of FontData.
	GetFont (see page 650)	Loads the required font using the font's ID.
	GetFontConfigList (see page 651)	Returns all of the font details in a drop down control.
	UpdateFont (see page 651)	Updates the font.

Legend

	Method

Font.AddFont Method

Adds a new font to the system.

C#

```
public void AddFont(Collection data);
```

Visual Basic

```
Public Function AddFont(data As Collection) As void
```

Parameters

Parameters	Description
data	Font (see page 648) 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 (see page 534) 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 706) 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(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFont", Description = "")>
Public Function GetFont(Id As Integer) As FontData
```

Parameters

Parameters	Description
Id	The font's ID

Returns

Integer

Font.GetFontConfigList Method

Returns all of the font details in a drop down control.

C#

```
[SoapHeader("AuthHeader")]
[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 (see page 534) 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 652)	Assigns a form submission task (see page 519), as specified in FormInfo (see page 351), of the SubmittedDataXml.
	EmailFormFieldData (see page 653)	Sends an email, as specified in FormInfo (see page 351), of the SubmittedDataXml.
	GetAllForms (see page 653)	Retrieves an array of all the forms contained with the specified folder. This method is overloaded.
	GetForm (see page 654)	Retrieves information about the specified form.
	GetFormDataHistogram (see page 654)	
	GetFormFieldData (see page 655)	Retrieves the submitted data for all fields from a specified form. This method is overloaded.
	GetFormFieldList (see page 656)	Retrieves the list of form fields defined in the specified form.
	GetFormFieldQuestions (see page 657)	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 (see page 534) in the descriptive name field in the form element dialog during design time.
	GetFormTitle (see page 657)	Same as GetFormById, but returns only the form's title.
	SubmitFormFieldData (see page 657)	Adds form data as if it was submitted during a browser post to the web server.
	UpdateFormFieldData (see page 658)	Updates a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

Legend

	Method

Form.AssignFormTask Method

Assigns a form submission task (see page 519), as specified in FormInfo (see page 351), 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 (Ektron.Cms.FormData (see page 781))
`SubmittedDataXml`	The form data submitted in XML format. Value Type: String (System.String)

Returns

Returns True if the task (see page 519) was successfully assigned. Boolean (System.Boolean)

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 657) if SendNotification is True and the form is configured to assign a task (see page 519) when data is submitted. The SubmittedDataXml is typically retrieved by calling SubmitFormFieldData (see page 657).

Form.EmailFormFieldData Method

Sends an email, as specified in FormInfo (see page 351), of the SubmittedDataXml.

C#

```
public void EmailFormFieldData(FormData FormInfo, string SubmittedDataXml);
```

Visual Basic

```
Public Function EmailFormFieldData(FormInfo As FormData, SubmittedDataXml As string) As void
```

Parameters

Parameters	Description
FormInfo	The properties of the form associated with the data. This information may be retrieved by calling GetFormById. Value Type: Cms.FormData (Ektron.Cms.FormData (see page 781))
SubmittedDataXml	The form data submitted in XML format. Value Type: String (System.String)

Remarks

This routine is called automatically when calling SubmitFormFieldData (see page 657) 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 657).

GetAllForms Method

Retrieves an array of all the forms contained with the specified folder. This method is overloaded.

Overload List

	Name	Description
	Form.GetAllForms () (see page 653)	Returns all of the forms as an array of FormData (see page 781).
	Form.GetAllForms (int, string, int, ContentAPI.FormResultType) (see page 654)	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 781).

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 781)()

Remarks

User (see page 706) login required to perform this operation.

Form.GetAllForms Method (int, string, int, ContentAPI.FormResultType)

Retrieves an array of all the forms contained with the specified folder. Forms may be limited to a single language and type.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllForms By Id, OrderBy, LanguageID and Type", Description = "")]
public FormData[] GetAllForms(int Id, string OrderBy, int LanguageId,
ContentAPI.FormResultType Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllForms By Id, OrderBy, LanguageID and Type", Description = "")>
Public Function GetAllForms(Id As Integer, OrderBy As string, LanguageId As Integer, Type As
ContentAPI.FormResultType) As FormData()
```

Parameters

Parameters	Description
Id	Specifies the folder.
OrderBy	Specifies how to sort the forms
LanguageId	Specifies a language or EkConstants.g_AllContentLanguages (-1) for all languages.
Type	Specifies whether only active forms or archived forms should be returned.

Remarks

Requires authentication.

Form.GetForm Method

Retrieves information about the specified form.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetForm", Description = "")]
public FormData GetForm(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetForm", Description = "")>
Public Function GetForm(Id As Integer) As FormData
```

Parameters

Parameters	Description
Id	Specifies the form.

Returns

Returns information about the form.

Remarks

Does not require authentication.

Form.GetFormDataHistogram Method

C#

```
[SoapHeader("AuthHeader")]
```

```
[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

Retrieves the submitted data for all fields from a specified form. This method is overloaded.

Overload List

	Name	Description
	Form.GetFormFieldData (int, int) (see page 655)	Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.
	Form.GetFormFieldData (int, string, string) (see page 656)	Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range.

Form.GetFormFieldData Method (int, int)

Retrieves a single instance of submitted data, as specified by FormDataId, for all fields from a specified form.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormFieldData By FormID and FormDataId", Description = "")]
public FormSubmittedData GetFormFieldData(int FormId, int FormDataId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormFieldData By FormID and FormDataId", Description = "")>
Public Function GetFormFieldData(FormId As Integer, FormDataId As Integer) As FormSubmittedData
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)
FormDataId	Specifies the instance of submitted data. Value Type: Int32 (System.Int32)

Returns

Returns the submitted data or null/Nothing, if none found. Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData (see page 799))

Remarks

Does not require authentication.

Form.GetFormFieldData Method (int, string, string)

Retrieves the submitted data for all fields from a specified form. All the data may be retrieved or just data submitted within a given date range.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormFieldData By FormId, StartDate and EndDate", Description =
"")]
public FormSubmittedData[] GetFormFieldData(int 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 Integer, StartDate As string, EndDate As string) As
FormSubmittedData()
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)
StartDate	Optionally limit the data returned to include only those submitted on or after the given date. Value Type: String (System.String)
EndDate	Optionally limit the data returned to include only those submitted on or before the given date. Value Type: String (System.String)

Returns

Returns an array of submitted data. Cms.FormSubmittedData (Ektron.Cms.FormSubmittedData (see page 799))

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(int FormId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormFieldList", Description = "")>
Public Function GetFormFieldList(FormId As Integer) As FormFieldList
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)

Returns

The list for defined fields in the given form. Cms.FormFieldList (Ektron.Cms.FormFieldList (see page 794))

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 (see page 534) in the descriptive name field in the form element dialog during design time.

C#

```
public Hashtable GetFormFieldQuestions(int FormId);
```

Visual Basic

```
Public Function GetFormFieldQuestions(FormId As Integer) As Hashtable
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)

Returns

a table of defined fields with the corresponding questions in a given form.

Description

[WebMethod(MessageName="GetFormFieldQuestions",Description="")]

Remarks

these questions is mainly used for reporting.

Form.GetFormTitle Method

Same as GetFormById, but returns only the form's title.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFormTitle", Description = "")]
public string GetFormTitle(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFormTitle", Description = "")>
Public Function GetFormTitle(Id As Integer) As string
```

Parameters

Parameters	Description
Id	Specifies the form.

Returns

The title of the specified form.

Remarks

Does not require authentication.

Form.SubmitFormFieldData Method

Adds form data as if it was submitted during a browser post to the web server.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "SubmitFormFieldData", Description = "")]
public void SubmitFormFieldData(int FormId, FormSubmittedData Data, bool SendNotification,
string SubmittedDataXml);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SubmitFormFieldData", Description = "")>
Public Function SubmitFormFieldData(FormId As Integer, Data As FormSubmittedData,
SendNotification As bool, SubmittedDataXml As string) As void
```

Parameters

Parameters	Description
FormId	Specifies the form that corresponds to the data. Value Type: Int32 (System.Int32)
Data	The data to submit. Value Type: Cms.FormSubmittedData (Ektron.Cms. FormSubmittedData (see page 799))
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 (see page 519) 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(int FormId, int FormDataId, FormSubmittedData SubmittedData);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UpdateFormFieldData", Description = "")>
Public Function UpdateFormFieldData(FormId As Integer, FormDataId As Integer, SubmittedData As
FormSubmittedData) As void
```

Parameters

Parameters	Description
FormId	Specifies the form. Value Type: Int32 (System.Int32)
FormDataId	Specifies the instance of submitted data. Value Type: Int32 (System.Int32)
SubmittedData	The field data values to be updated. Value Type: Cms.FormSubmittedData (Ektron.Cms. FormSubmittedData (see page 799))

Remarks

Does not require authentication.

Library Class

Summary description for Library

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Library : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Library
Inherits EktronBaseService
```

File

Library.cs

Library Methods

The methods of the Library class are listed here.

Public Methods

	Name	Description
	AddLoadBalanceItem (see page 659)	Adds a load balance path for files and images.
	DeleteLibraryItem (see page 660)	Deletes the library item by extension.
	DeleteLibraryItemById (see page 660)	Deletes the library items from the system.
	DeleteLoadBalanceItem (see page 660)	Deletes the load balance item from the system.
	GetAllChildLibItems (see page 661)	Loads all of the library items by type.
	GetAllLoadBalancePaths (see page 661)	Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.
	GetLibraryItem (see page 662)	Loads the library item's data. This method is overloaded.
	GetLibrarySettings (see page 664)	Loads the library settings.
	GetLibraryTypes (see page 664)	Loads the library types.
	GetLoadBalancePath (see page 665)	Gets the load balance path by ID.
	InternalLibrarySearch (see page 665)	Searches library items.
	UpdateLibrary (see page 665)	Updates a library item.
	UpdateLibrarySettings (see page 666)	Update the library settings.
	UpdateLoadBalanceSettings (see page 666)	Updates the load balance settings.
	UpdateQlinkTemplates (see page 666)	Updates the Quicklink template.

Legend

	Method

Library.AddLoadBalanceItem Method

Adds a load balance path for files and images.

C#

```
public void AddLoadBalanceItem(string Type, Collection pagedata);
```

Visual Basic

```
Public Function AddLoadBalanceItem(Type As string, pagedata As Collection) As void
```

Parameters

Parameters	Description
Type	The path type, either images or files
pagedata	Load balance item values for path and relative

Remarks

A user (see page 534) 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 658) item data

Remarks

A user (see page 534) 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 658) item data

Remarks

A user (see page 534) 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(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteLoadBalanceItem", Description = "")>
```

```
Public Function DeleteLoadBalanceItem(Id As Integer) As void
```

Parameters

Parameters	Description
Id	The load balance item ID

Remarks

A user (see page 534) 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, int 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 Integer, OrderBy As string,
currentPageNum As Integer, pageSize As Integer, totalPages As ref int) As LibraryData()
```

Parameters

Parameters	Description
Type	The type (Required)
ParentId	The folder's ID
OrderBy	The order to load the library items
currentPageNum	The current page number
pageSize	The page size
totalPages	The total pages

Returns

Ektron.Cms.LibraryData()

Remarks

User (see page 706) login required to perform this operation

GetAllLoadBalancePaths Method

Returns all the load balance paths as a array of LoadBalanceData. This method is overloaded.

Overload List

	Name	Description
	Library.GetAllLoadBalancePaths (int, string) (see page 662)	Returns an array of LoadBalanceData.
	Library.GetAllLoadBalancePaths (string) (see page 662)	Returns all the load balance paths as a array of LoadBalanceData.

Library.GetAllLoadBalancePaths Method (int, string)

Returns an array of LoadBalanceData.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetAllLoadBalancePaths By Id and Type", Description = "")]
public LoadBalanceData[] GetAllLoadBalancePaths(int Id, string Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllLoadBalancePaths By Id and Type", Description = "")>
Public Function GetAllLoadBalancePaths(Id As Integer, Type As string) As LoadBalanceData()
```

Parameters

Parameters	Description
Id	Load balance ID
Type	Use images or files

Returns

Ektron.Cms.LoadBalanceData()

Remarks

A user (see page 534) 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 (see page 534) with administrator privileges must be logged in to perform this operation.

GetLibraryItem Method

Loads the library item's data. This method is overloaded.

Overload List

	Name	Description
	Library.GetLibraryItem (int) (see page 663)	Loads the library item data.
	Library.GetLibraryItem (int, int) (see page 663)	Loads the library item's data.

Library.GetLibraryItem Method (int)

Loads the library item data.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibraryItem By Id", Description = "")]
public LibraryData GetLibraryItem(int LibID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibraryItem By Id", Description = "")>
Public Function GetLibraryItem(LibID As Integer) As LibraryData
```

Parameters

Parameters	Description
LibID	The library ID

Returns

Ektron.Cms.LibraryData

Library.GetLibraryItem Method (int, int)

Loads the library item's data.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibraryItem By Id And ParentID", Description = "")]
public LibraryData GetLibraryItem(int Id, int ParentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibraryItem By Id And ParentID", Description = "")>
Public Function GetLibraryItem(Id As Integer, ParentId As Integer) As LibraryData
```

Parameters

Parameters	Description
Id	The library's ID
ParentId	The folder's ID

Returns

Ektron.Cms.LibraryData

Remarks

Read permission to the folder is required to perform this operation

Library.GetLibrarySettings Method

Loads the library settings.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibrarySettings", Description = "")]
public LibraryConfigData GetLibrarySettings(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibrarySettings", Description = "")>
Public Function GetLibrarySettings(Id As Integer) As LibraryConfigData
```

Parameters

Parameters	Description
`Id`	The folder's ID

Returns

Ektron.Cms.LibraryConfigData

Remarks

Login (see page 229) required to perform this operation.

Library.GetLibraryTypes Method

Loads the library types.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetLibraryTypes", Description = "")]
public LibraryTypeData[] GetLibraryTypes(int Id, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetLibraryTypes", Description = "")>
Public Function GetLibraryTypes(Id As Integer, OrderBy As string) As LibraryTypeData()
```

Parameters

Parameters	Description
`Id`	The folder's ID
`OrderBy`	Order by clause

Returns

Object

Remarks

Library (see page 658) 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(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "", Description = "")>
Public Function GetLoadBalancePath(Id As Integer) As LoadBalanceData
```

Parameters

Parameters	Description
Id	th load balance path's ID

Returns

Ektron.Cms.LoadBalanceData

Library.InternalLibrarySearch Method

Searches library items.

C#

```
public SearchResultData[] InternalLibrarySearch(Collection pagedata,
Ektron.Cms.Common.SearchAssetRequest sar);
```

Visual Basic

```
Public Function InternalLibrarySearch(pagedata As Collection, sar As
Ektron.Cms.Common.SearchAssetRequest) As SearchResultData()
```

Parameters

Parameters	Description
pagedata	Search (see page 514) criteria
sar	Search (see page 514) asset criteria

Returns

Ektron.Cms.SearchResultData()

Remarks

User (see page 706) 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 658) 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 658) settings data

Remarks

A user (see page 534) 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 706) 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 (see page 534) with administrator privileges must be logged in to perform this operation.

Metadata Class

Summary description for Metadata

Class Hierarchy

C#

```
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.None)]
public class Metadata : EktronBaseService;
```

Visual Basic

```
<WebService(Namespace = "http://tempuri.org/")>
<WebServiceBinding(ConformsTo = WsiProfiles.None)>
Public Class Metadata
Inherits EktronBaseService
```

File

Metadata.cs

Metadata Methods

The methods of the Metadata class are listed here.

Public Methods

	Name	Description
	AddMetaDataType (see page 667)	Adds a new metadata type to the system
	DeleteMetadataType (see page 668)	Deletes a metadata type from the system.
	GetContentMetadataList (see page 668)	
	GetMetadataType (see page 669)	Loads all content (see page 449) metadata.
	GetMetaDataTypes (see page 670)	Loads all the metadata types.
	UpdateMetadataType (see page 670)	Updates the metadata type.

Legend

	Method

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(metaDataItem As ref ContentMetaData) As void
```

Parameters

Parameters	Description
metaDataItem	ContentMetadata object containing metadata definition details

Remarks

A user (see page 534) 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(int MetadataId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteMetadataType", Description = "")>
Public Function DeleteMetadataType(MetadataId As Integer) As void
```

Parameters

Parameters	Description
MetadataId	Metatype ID

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

GetContentMetadataList Method

Overload List

	Name	Description
	Metadata.GetContentMetadataList (int) (see page 668)	Loads all of the metadata for the given content (see page 449).
	Metadata.GetContentMetadataList (int, bool) (see page 669)	Loads all of the metadata for the given content (see page 449).

Metadata.GetContentMetadataList Method (int)

Loads all of the metadata for the given content (see page 449).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentMetadataList", Description = "")]
public CustomAttributeList GetContentMetadataList(int ContentId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentMetadataList", Description = "")>
Public Function GetContentMetadataList(ContentId As Integer) As CustomAttributeList
```

Parameters

Parameters	Description
ContentId	The contents ID.

Returns

Ektron.Cms.CustomAttributeList

Remarks

Read permissions required.

Metadata.GetContentMetadataList Method (int, bool)

Loads all of the metadata for the given content (see page 449).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetContentMetadataList with preview", Description = "")]
public CustomAttributeList GetContentMetadataList(int ContentId, bool preview);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetContentMetadataList with preview", Description = "")>
Public Function GetContentMetadataList(ContentId As Integer, preview As bool) As
CustomAttributeList
```

Parameters

Parameters	Description
ContentId	The contents ID.
preview	True implies get metadata from staging content (see page 449), else use published content (see page 449)

Returns

Ektron.Cms.CustomAttributeList

Remarks

Read permissions required.

Metadata.GetMetadataType Method

Loads all content (see page 449) metadata.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetMetadataType", Description = "")]
public ContentMetaData GetMetadataType(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetMetadataType", Description = "")>
```

```
Public Function GetMetadataType(Id As Integer) As ContentMetaData
```

Parameters

Parameters	Description
Id	The metadata type ID

Returns

Ektron.Cms.ContentMetaData

Remarks

A user (see page 534) 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 706) 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 667) types

Remarks

A user (see page 534) 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 671)	Adds the item permissions for a user (see page 534) or a usergroup.
	DeleteItemPermission (see page 672)	Deletes a user (see page 534) or a group from the item's permissions
	DisableContentInheritance (see page 672)	Disable permissions inheritance on the content (see page 449) and copies current permissions to the content (see page 449).
	DisableFolderInheritance (see page 673)	Disable permissions inheritance on the folder and copies current permissions to the folder.
	EnableContentInheritance (see page 673)	Deletes current permissions from the content (see page 449) and enable permissions inheritance on the content (see page 449).
	EnableFolderInheritance (see page 674)	Deletes current permissions from the folder and enable permissions inheritance on the folder.
	GetUserPermissions (see page 674)	
	LoadPermissions (see page 675)	Loads the permission for given object, such as content (see page 449), folder, task (see page 519), etc.
	UpdateItemPermission (see page 676)	Updates the item permissions for a user (see page 534) or a usergroup.

Legend

	Method

Permissions.AddItemPermission Method

Adds the item permissions for a user (see page 534) 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 706)'s Permissions (see page 671) Data

Returns

False if it's successful

Permissions.DeleteItemPermission Method

Deletes a user (see page 534) 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(int id, bool IsGroupId, int 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 Integer, IsGroupId As bool, itemId As Integer,
itemType As EkEnumeration.CMSObjectTypes) As bool
```

Parameters

Parameters	Description
id	User (see page 706) or Group Id
IsGroupId	Given ID is a user (see page 534) id or a group id?
itemId	ContentId or FolderId
itemType	Given item id is EkEnumeration.CMSObjectTypes.Content (see page 622) or EkEnumeration.CMSObjectTypes.Folder (see page 644)?

Returns

False if it's successful.

Permissions.DisableContentInheritance Method

Disable permissions inheritance on the content (see page 449) and copies current permissions to the content (see page 449).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DisableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool DisableContentInheritance(int 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 Integer) As bool
```

Parameters

Parameters	Description
id	Content (see page 622) 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(int 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 Integer) As bool
```

Parameters

Parameters	Description
id	Folder (see page 644) Id

Returns

False if it's successful.

Permissions.EnableContentInheritance Method

Deletes current permissions from the content (see page 449) and enable permissions inheritance on the content (see page 449).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EnableContentInheritance", Description = "Deletes a user or a group
from the item's permissions")]
public bool EnableContentInheritance(int 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 Integer) As bool
```

Parameters

Parameters	Description
id	Content (see page 622) 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(int 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 Integer) As bool
```

Parameters

Parameters	Description
id	Folder (see page 644) Id

Returns

False if it's successful.

GetUserPermissions Method

Overload List

	Name	Description
	Permissions.GetUserPermissions (int, string, int, string) (see page 674)	Loads the user (see page 534) permissions as a array of UserPermissionData.
	Permissions.GetUserPermissions (int, string, int, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType) (see page 675)	Loads the user (see page 534) permissions as a array of UserPermissionData.

Permissions.GetUserPermissions Method (int, string, int, string)

Loads the user (see page 534) 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(int Id, string ItemType, int 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 Integer, ItemType As string, UserId As Integer,
UserGroupList As string) As UserPermissionData()
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user (see page 534)'s ID
UserGroupList	the group list

Returns

Ektron.Cms.UserPermissionData()

Permissions.GetUserPermissions Method (int, string, int, string, ContentAPI.PermissionUserType, ContentAPI.PermissionRequestType)

Loads the user (see page 534) 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(int Id, string ItemType, int 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 Integer, ItemType As string, UserId As Integer,
UserGroupList As string, PermissionType As ContentAPI.PermissionUserType, PermissionRequest As
ContentAPI.PermissionRequestType) As UserPermissionData()
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user (see page 534)'s ID
UserGroupList	the group list
PermissionType	PermissionUserType
PermissionRequest	PermissionRequestType

Returns

Ektron.Cms.UserPermissionData()

Permissions.LoadPermissions Method

Loads the permission for given object, such as content (see page 449), folder, task (see page 519), 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(int 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 Integer, RequestType As string, Type As
ContentAPI.PermissionResultType) As PermissionData
```

Parameters

Parameters	Description
`Id`	The object ID
`RequestType`	The object type := content (see page 449) | folder | collections | tasks
`Type`	The permission result type

Returns

Ektron.Cms.PermissionData

Permissions.UpdateItemPermission Method

Updates the item permissions for a user (see page 534) 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 706)'s Permissions (see page 671) 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 677)	Returns the array of ContentData (see page 738). This method is overloaded.
	Search (see page 678)	
	SearchForCategories (see page 680)	Search (see page 678) based on taxonomy category.
	SearchTable (see page 680)	Search (see page 678) html, content (see page 449), images, multimedia, documents based on content (see page 449) or metadata.

Legend

	Method

ExecuteSearch Method

Returns the array of ContentData (see page 738). This method is overloaded.

Overload List

	Name	Description
	SearchManager.ExecuteSearch (ContentSearchCondition) (see page 677)	Returns the array of ContentData (see page 738) based on the criteria ContentSearchCondition.
	SearchManager.ExecuteSearch (UserSearchCondition) (see page 678)	Returns the array of UserData (see page 801) based on the criteria UserSearchCondition.

SearchManager.ExecuteSearch Method (ContentSearchCondition)

Returns the array of ContentData (see page 738) 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 738)

SearchManager.ExecuteSearch Method (UserSearchCondition)

Returns the array of UserData (see page 801) 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 801)

Search Method

Overload List

	Name	Description
	SearchManager.Search (MapRequestData, int, int, ref int) (see page 678)	Map (see page 232) search for retrieving geographical content (see page 449) from CMS400.Net site.
	SearchManager.Search (SearchRequestData, ref int) (see page 679)	Search html, content (see page 449), images, multimedia, documents based on content (see page 449) or metadata.
	SearchManager.Search (string, ref int) (see page 679)	Search the site for occurrence(s) of search text.

SearchManager.Search Method (MapRequestData, int, int, ref int)

Map (see page 232) search for retrieving geographical content (see page 449) 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,
resultCount As ref int) 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 (see page 449) satisfying location criteria

SearchManager.Search Method (SearchRequestData, ref int)

Search html, content (see page 449), images, multimedia, documents based on content (see page 449) 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, resultCount As ref int) 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 (see page 449) satisfying search criteria

SearchManager.Search Method (string, ref 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, resultCount As ref int) As SearchResponseData()
```

Parameters

Parameters	Description
resultCount	Number of records returned.
searchReqData	Search criteria bundled in a SearchRequestData object.

SearchManager.SearchForCategories Method

Search (see page 678) 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, resultCount As ref int) As String()
```

Parameters

Parameters	Description
searchReqData	Search (see page 678) 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 678) html, content (see page 449), images, multimedia, documents based on content (see page 449) 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,
resultCount As ref int) As DataSet
```

Parameters

Parameters	Description
searchReqData	Search (see page 678) criteria bundled in a SearchRequestData object.
resultCount	Number of records returned.
context	The current Http request.

Returns

Array of SearchResponseData - content (see page 449) satisfying search (see page 514) 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 682)	Removes the task (see page 519) permissions for a given user (see page 534).
	GetAllActiveLanguages (see page 682)	Gets all languages that have been activated in the CMS400 Workarea, Settings, Configuration, Language Settings area.
	GetAllLanguages (see page 683)	
	GetClientDayNames (see page 684)	Returns the day name of a given content (see page 449) language (or site language).
	GetClientMonthNames (see page 684)	Returns the month name of a given content (see page 449) language (or site language)
	GetCookieValue (see page 684)	Reads the cookie value.
	GetLanguage (see page 685)	Returns the details about the given language ID.

	GetLanguages (see page 685)	Gets a list of all languages that meet a given state, for example, active.
	GetSiteVariables (see page 685)	Returns all of the settings data.
	GetTaskPermission (see page 686)	Returns 1 if the user (see page 534) has one of the task (see page 519) permissions, otherwise 0.
	IsTaskPermissionInherited (see page 686)	Returns true if a user (see page 534) has already granted one of the task (see page 519) permissions.
	SetCookieValue (see page 687)	Resets the cookie value.
	SetTaskPermission (see page 687)	Updates the task (see page 519) permission(s) for the given user (see page 534) or group.
	UpdateSiteEnabledLanguages (see page 688)	Updates the languages in the given list to be enabled on the site. Languages not in the list are disabled.

Legend

	Method

Site.BreakTaskPermissionInherit Method

Removes the task (see page 519) permissions for a given user (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "BreakTaskPermissionInherit", Description = "")]
public void BreakTaskPermissionInherit(int UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "BreakTaskPermissionInherit", Description = "")>
Public Function BreakTaskPermissionInherit(UserId As Integer) As void
```

Parameters

Parameters	Description
`UserId`	User (see page 706) ID

Remarks

The user (see page 534) should be logged in to the system. Otherwise, the user (see page 534) 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

Overload List

	Name	Description
	Site.GetAllLanguages () (see page 683)	Gets a list of all the available languages sorted by language name in ascending order.
	Site.GetAllLanguages (LanguageDataSet.SortBy, LanguageDataSet.SortDirection) (see page 683)	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 (see page 449) 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 (see page 449) 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(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetSiteVariables", Description = "")>
Public Function GetSiteVariables(Id As Integer) As SettingsData
```

Parameters

Parameters	Description
Id	User (see page 706) ID

Returns

Ektron.Cms.SettingsData

Remarks

If user (see page 534) ID is passed it returns the additional value of MaxUsersAllowed, AdInstalled and AdAllowed properties.

Site.GetTaskPermission Method

Returns 1 if the user (see page 534) has one of the task (see page 519) permissions, otherwise 0.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetTaskPermission", Description = "")]
public int GetTaskPermission(int Id, int Type);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTaskPermission", Description = "")>
Public Function GetTaskPermission(Id As Integer, Type As Integer) As Integer
```

Parameters

Parameters	Description
Id	User (see page 706) or group ID
Type	User (see page 706) or group (1 = user (see page 534), 2 = group).

Returns

Integer

Remarks

The user (see page 534) should be logged in to the system. Otherwise, the user (see page 534) will get the permission denied error.

Site.IsTaskPermissionInherited Method

Returns true if a user (see page 534) has already granted one of the task (see page 519) permissions.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "IsTaskPermissionInherited", Description = "")]
public bool IsTaskPermissionInherited(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsTaskPermissionInherited", Description = "")>
Public Function IsTaskPermissionInherited(Id As Integer) As bool
```

Parameters

Parameters	Description
Id	User (see page 706) ID

Returns

Boolean

Remarks

To use this function, the user (see page 534) should be logged in to the system. Otherwise, the user (see page 534) 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 (see page 519) permission(s) for the given user (see page 534) or group.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "SetTaskPermission", Description = "")]
public void SetTaskPermission(int Id, int UserType, int PermissionType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SetTaskPermission", Description = "")>
Public Function SetTaskPermission(Id As Integer, UserType As Integer, PermissionType As
Integer) As void
```

Parameters

Parameters	Description
`Id`	User (see page 706) ID or Group ID
`UserType`	User (see page 706) (1) or Group (2)
`PermissionType`	Ektron.Cms.Common.TaskPermission (1 = Create Task (see page 576), 2 = Redirect Task (see page 576), 3 = Delete Task (see page 576), 4 = Edit Alias)

Remarks

To use this function, the user (see page 534) should be logged in to the system. Otherwise, user (see page 534) 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 689)	Get breadcrumb path of the folder in which content (see page 449) block exists.
	GetWebsiteMap (see page 689)	Creates the site map from folder breadcrumb.
	RenderFolderBreadcrumbHtml (see page 690)	Render folder breadcrumb HTML
	RenderWebsiteMap (see page 690)	Create HTML to render on the page.

Legend

	Method

SiteMap.GetFolderBreadcrumbPath Method

Get breadcrumb path of the folder in which content (see page 449) block exists.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetFolderBreadcrumbPath", Description = "")]
public SitemapPath[] GetFolderBreadcrumbPath(int id, bool isFolder);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetFolderBreadcrumbPath", Description = "")>
Public Function GetFolderBreadcrumbPath(id As Integer, isFolder As bool) As SitemapPath()
```

Parameters

Parameters	Description
`id`	Content (see page 622) or Folder (see page 644) Id
`isFolder`	set it true if the id is folder id

Returns

Array of node in the path

SiteMap.GetWebsiteMap Method

Creates the site map from folder breadcrumb.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetWebsiteMap", Description = "")]
public WebsiteMap GetWebsiteMap(int folderId, bool getSameLevelNodes);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetWebsiteMap", Description = "")>
Public Function GetWebsiteMap(folderId As Integer, getSameLevelNodes As bool) As WebsiteMap
```

Parameters

Parameters	Description
`folderId`	Build the web sitemap from the folder specified.
`getSameLevelNodes`	True indicates include all sites at same level as folderId

SiteMap.RenderFolderBreadcrumbHtml Method

Render folder breadcrumb HTML

C#

```
[SoapHeader("AuthHeader")]
[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	
pathSeparator	
linkNodes	
linkTarget	
displayVerticle	

Returns

HTML

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

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 692)	
	AddTaxonomySyncFolder (see page 692)	Adds a folder entry into the taxonomy.
	CreateTaxonomy (see page 693)	Creates a new taxonomy.
	DeleteTaxonomy (see page 693)	Deletes the existing taxonomy.
	GetTaxonomyIdByPath (see page 694)	Returns the taxonomy ID by a given path.
	LoadTaxonomy (see page 694)	Returns the TaxonomyData with all its subcategories and its items.
	ReadAllAssignedCategory (see page 694)	Returns all the assigned category for the content (see page 449) ID
	ReadTaxonomy (see page 695)	Read the taxonomy data.
	RemoveTaxonomyFolder (see page 695)	Removes the sync up folder from the taxonomy.
	RemoveTaxonomyItem (see page 696)	Removes an item from the taxonomy.
	ReOrderTaxonomyItems (see page 696)	Reorder the display of taxonomy items.
	UpdateTaxonomy (see page 696)	Updates the existing taxonomy.

Legend

	Method

AddTaxonomyItem Method

Overload List

	Name	Description
	Taxonomy.AddTaxonomyItem (TaxonomyContentRequest) (see page 692)	Inserts an item into the taxonomy.
	Taxonomy.AddTaxonomyItem (TaxonomyRequest) (see page 692)	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 int CreateTaxonomy(TaxonomyData TaxonomyReq);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "CreateTaxonomy", Description = "creates new taxonomy")>
Public Function CreateTaxonomy(TaxonomyReq As TaxonomyData) As Integer
```

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 int 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 Integer
```

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(TaxonomyReq As ref TaxonomyRequest) As TaxonomyData
```

Parameters

Parameters	Description
TaxonomyReq	TaxonomyRequest

Returns

TaxonomyData

Taxonomy.ReadAllAssignedCategory Method

Returns all the assigned category for the content (see page 449) ID

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "ReadAllAssignedCategory", Description = "returns all the assigned
category for the content id")]
public TaxonomyBaseData[] ReadAllAssignedCategory(int 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 Integer) 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(TaxonomyReq As ref 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 698)	Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.
	AddForum (see page 698)	Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum.
	AddTopic (see page 699)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.
	DeleteBoard (see page 700)	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 701)	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 701)	Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value.
	GetBoardCategories (see page 701)	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 702)	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 702)	Method to get a discussion board by ID or path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.
	GetDiscussionBoardString (see page 703)	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 704)	Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum (see page 167) array. If the forum does not exist, then a DiscussionBoard with a zero length Forums property will be returned.

	GetRepliesForTopics (see page 704)	Method returns a list of replies for a given forum ID. **ArrayList Structure** • ArrayListItem(j)(0) ' content_id • ArrayListItem(j)(1) ' comment_total • ArrayListItem(j)(2) ' comment_sum • ArrayListItem(j)(3) ' last_posted • ArrayListItem(j)(4) ' content_language
	GetTopic (see page 705)	Method to obtain a particular Topic by ID. Will return a discussion board with a 1 length Discussion Forum (see page 167) 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 705)	Method to obtain an array of topic replies matching search (see page 514) criteria. Will return an instance of EkTasks, which is effectively an array of EkTask objects. If no results are returned, then a zero length EkTasks will be returned.

Legend

	Method

ThreadedDiscussion.AddBoard Method

Adds a discussion board to a given folder. Accepts name, title, and other parameters. Returns integer ID of new discussion board.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddBoard", Description = "")]
public int AddBoard(int 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 Integer, Name As string, Title As string,
RequiresAuthentication As bool, StyleSheet As string, Categories As string[]) As Integer
```

Parameters

Parameters	Description
`ParentID`	Integer ID of folder to add discussion board to.
`Name`	Name of board.
`Title`	Title of board.
`RequiresAuthentication`	Does the Board require authentication?
`StyleSheet`	The stylesheet to apply to the board.
`Categories`	String array of categories for the board.

Returns

Integer ID of new discussion board.

ThreadedDiscussion.AddForum Method

Adds a discussion forum to a given board. Accepts name, description, and other parameters. Category ID must be valid for the board ID. Returns integer ID of new discussion forum.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "AddForum", Description = "")]
public int AddForum(int BoardID, string Name, string Description, bool ModeratePosts, bool
LockForum, int SortOrder, int CategoryID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddForum", Description = "")>
Public Function AddForum(BoardID As Integer, Name As string, Description As string,
ModeratePosts As bool, LockForum As bool, SortOrder As Integer, CategoryID As Integer) As
Integer
```

Parameters

Parameters	Description
BoardID	Integer 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	Integer of the Category ID it belongs to.

Returns

Integer ID of the new forum.

AddTopic Method

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer. This method is overloaded.

Overload List

	Name	Description
	ThreadedDiscussion.AddTopic (int, string, string) (see page 699)	Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer.
	ThreadedDiscussion.AddTopic (int, string, string, int) (see page 700)	Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as integer.

ThreadedDiscussion.AddTopic Method (int, string, string)

Method to add a topic based on a forum ID. Accepts title and topic starting message. Returns Topic ID as integer.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTopic", Description = "")]
public int AddTopic(int ForumID, string Title, string Message);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddTopic", Description = "")>
Public Function AddTopic(ForumID As Integer, Title As string, Message As string) As Integer
```

Parameters

Parameters	Description
ForumID	Integer value of forum to add topic to.
Title	String value of the topic title.
Message	String value of the topic message.

Returns

Integer value for the topic ID.

ThreadedDiscussion.AddTopic Method (int, string, string, int)

Overloaded method to add a topic based on a forum ID and language. Accepts title and topic starting message. Returns Topic ID as integer.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddTopic With Language ID", Description = "")]
public int AddTopic(int 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 Integer, Title As string, Message As string, LanguageId As
Integer) As Integer
```

Parameters

Parameters	Description
ForumID	Integer value of forum to add topic to.
Title	String value of the topic title.
Message	String value of the topic message.
LanguageId	Integer id of the Language ID.

Returns

Integer value for the topic ID.

ThreadedDiscussion.DeleteBoard Method

Deletes a board with a given ID. This will delete the board as well as all forums, topics and posts under it. BoardID must be greater than 0.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteBoard", Description = "")]
public void DeleteBoard(int BoardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteBoard", Description = "")>
Public Function DeleteBoard(BoardID As Integer) As void
```

Parameters

Parameters	Description
BoardID	The integer ID of the board. This corresponds to the folder ID.

ThreadedDiscussion.DeleteForum Method

Deletes a forum with a given ID. This will delete the forum as well as all topics and posts under it. forumID must be greater than 0.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteForum", Description = "")]
public void DeleteForum(int forumID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteForum", Description = "")>
Public Function DeleteForum(forumID As Integer) As void
```

Parameters

Parameters	Description
forumID	The integer ID of the forum. This corresponds to the folder ID.

ThreadedDiscussion.DeleteTopic Method

Deletes a topic by topicID. This method deletes all posts associated with the topic as well. Returns boolean value.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteTopic", Description = "")]
public bool DeleteTopic(int topicid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteTopic", Description = "")>
Public Function DeleteTopic(topicid As Integer) As bool
```

Parameters

Parameters	Description
post_id	Integer value of the topic you wish to delete.

Returns

Boolean

ThreadedDiscussion.GetBoardCategories Method

Method to get the Categories for a board by ID Returns either zero length or populated array. See DiscussionCategory class for more information.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBoardCategories", Description = "")]
public DiscussionCategory[] GetBoardCategories(int boardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBoardCategories", Description = "")>
```

```
Public Function GetBoardCategories(boardID As Integer) As DiscussionCategory()
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns an array of type DiscussionCategory.

ThreadedDiscussion.GetBoardForums Method

Method to get the Forums for a board by ID Returns either zero length or populated array. See DiscussionForum class for more information.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetBoardForums", Description = "")]
public DiscussionForum[] GetBoardForums(int boardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetBoardForums", Description = "")>
Public Function GetBoardForums(boardID As Integer) As DiscussionForum()
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns an array of type DiscussionForum.

GetDiscussionBoard Method

Method to get a discussion board by ID or path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

Overload List

	Name	Description
	ThreadedDiscussion.GetDiscussionBoard (int) (see page 702)	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 703)	Method to get a discussion board by path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

ThreadedDiscussion.GetDiscussionBoard Method (int)

Method to get a board by ID. Returns a DiscussionBoard class for a given ID value. See DiscussionBoard class for more information on the properties of the returned object.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetDiscussionBoard By Board ID", Description = "")]
```

```
public DiscussionBoard GetDiscussionBoard(int boardID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetDiscussionBoard By Board ID", Description = "")>
Public Function GetDiscussionBoard(boardID As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

Returns

Returns a class of type DiscussionBoard.

ThreadedDiscussion.GetDiscussionBoard Method (string)

Method to get a discussion board by path. Returns a DiscussionBoard class for a given path. See DiscussionBoard class for more information on the properties of the returned object.

C#

```
[SoapHeader("AuthHeader")]
[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: EktronCMS400.NET.

Returns

Returns a class of type DiscussionBoard.

ThreadedDiscussion.GetDiscussionBoardString Method

Method to get the rendered view of a discussion board by ID. Returns a string of the output based upon the input parameters. Display of the board can be controlled by using CSS stylesheets.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "", Description = "")]
public string GetDiscussionBoardString(int boardID, string URLpath);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "", Description = "")>
Public Function GetDiscussionBoardString(boardID As Integer, URLpath As string) As string
```

Parameters

Parameters	Description
boardID	Integer value of the board ID.

URLpath	String value of the URL path used. This is generally the Server Variable "URL". Ex: "/forum.aspx".

Returns

Returns a string of the rendered Discussion Board, suitable for the server control or a Web service.

ThreadedDiscussion.GetForum Method

Method to obtain a particular forum by ID. Will return a discussion board with a 1 length Discussion Forum (see page 167) 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(int forumid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetForum", Description = "")>
Public Function GetForum(forumid As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
forumid	Integer value of the forum ID.

Returns

An instance of type DiscussionBoard.

ThreadedDiscussion.GetRepliesForTopics Method

Method returns a list of replies for a given forum ID.

ArrayList Structure

- ArrayListItem(j)(0) ' content_id
- ArrayListItem(j)(1) ' comment_total
- ArrayListItem(j)(2) ' comment_sum
- ArrayListItem(j)(3) ' last_posted
- ArrayListItem(j)(4) ' content_language

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetRepliesForTopics", Description = "")]
public ArrayList GetRepliesForTopics(int ForumID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetRepliesForTopics", Description = "")>
Public Function GetRepliesForTopics(ForumID As Integer) As ArrayList
```

Parameters

Parameters	Description
ForumID	Integer value of the forum ID.

Returns

An instance of type ArrayList.

Remarks

This is an older method and is deprecated. The information returned herein was merged into the higher performance GetForum (see page 704)(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 167) 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(int topicid);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetTopic", Description = "")>
Public Function GetTopic(topicid As Integer) As DiscussionBoard
```

Parameters

Parameters	Description
`topicid`	Integer value of the Topic ID.

Returns

An instance of type DiscussionBoard.

ThreadedDiscussion.SearchReplies Method

Method to obtain an array of topic replies matching search (see page 514) 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(int[] ForumIDs, string[] SearchTerms, EkEnumeration.SearchTypes
stType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "SearchReplies", Description = "")>
Public Function SearchReplies(ForumIDs As int[], SearchTerms As string[], stType As
EkEnumeration.SearchTypes) As EkTasks
```

Parameters

Parameters	Description
`ForumIDs`	Integer array of the Forum (see page 167) IDs you want to search (see page 514). Use zero length array, or 0, to search (see page 514) all.
`SearchTerms`	String array of the search (see page 514) terms (case insensitive).
`stType`	EkEnumeration.SearchTypes enumeration value for the search (see page 514) 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 708)	Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters. This method is overloaded.
	AddMembershipUser (see page 709)	Creates new membership user (see page 534).
	AddNewUser (see page 709)	Adds a new user (see page 534) to the CMS.
	AddRolePermission (see page 710)	Adds permission for the given role.
	AddUser (see page 710)	Creates new cms user (see page 534)
	AddUserGroup (see page 711)	Creates new user (see page 534) group.
	AddUserToGroup (see page 711)	Adds an existing user (see page 534) to an existing group.
	AutoLogInUser (see page 712)	Login (see page 229) the given user (see page 534) credentials.
	CreateBlankUserObject (see page 712)	This method is overloaded.
	DeleteUserByID (see page 713)	Removes the user (see page 534) from the system. This will not purge the data, but it disables the user (see page 534) from further accessing the system.
	DeleteUserByIds (see page 713)	Removes the user (see page 534) from the system. This will not purge the data, but it disables the user (see page 534) from further accessing the system.
	DeleteUserFromGroup (see page 714)	Removes a user (see page 534) from the designated group.
	DeleteUserGroup (see page 714)	Removes the group from the system permanently.
	EditUserCustomProperties (see page 715)	This method creates the editable fields(UI) for user (see page 534)'s custom properties.

	GetActiveUser (see page 715)	Returns all of the user (see page 534) properties with or without preference.
	GetADMapping (see page 715)	Loads the CMS and Active Directory (see page 118) name.
	GetAllCmsUsers (see page 716)	Loads all the CMS users.
	GetAllCustomProperties (see page 717)	Returns array of all the custom properties defined in the system.
	GetAllUserByCustomProperty (see page 717)	Returns all the users based on the given properties.
	GetAllUserGroups (see page 717)	Loads all of the user groups from the system. This method is overloaded.
	GetAllUsers (see page 719)	This method is overloaded.
	GetGroupsUserIsIn (see page 720)	Loads all the groups by user (see page 534).
	GetMembershipUser (see page 720)	Returns user (see page 534) data based on the user (see page 534)'s login name (username).
	GetUser (see page 721)	Loads the user (see page 534) details for the given user (see page 534).
	GetUserbyUsername (see page 721)	Loads the user (see page 534) details for the given user (see page 534).
	GetUserCustomProperties (see page 722)	Returns an array of custom properties for a given user (see page 534)'s ID.
	GetUserEmailInfo (see page 722)	Returns user (see page 534) data for a given user (see page 534)'s ID.
	GetUserForFolderAdmin (see page 723)	
	GetUserGroup (see page 723)	Loads a given group's details.
	GetUserGroupForFolderAdmin (see page 724)	Returns all the groups data for the administrator.
	GetUserPermissions (see page 724)	Loads the user (see page 534) permissions as a array of UserPermissionData.
	GetUserPreference (see page 725)	Loads the preferences for the given user (see page 534).
	GetUsers (see page 725)	An Overloaded function that allows you to get the users from a specific group or load and array of user group data. This method is overloaded.
	GetUsersForTask (see page 726)	Loads the users that have permissions to the given content (see page 449).
	GetUsersNotInGroup (see page 727)	Loads all the users that are not in the given group.
	IsAdmin (see page 728)	Determines the if the current user (see page 534) is an Administrator.
	IsARoleMember (see page 728)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method is overloaded.
	IsARoleMemberForContent (see page 731)	Determines whether a Role Member is allowed to work with a given piece of content (see page 449) as an administrator or not. Administrations privileges are for the given piece of content (see page 449) 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 (see page 534).
	IsARoleMemberForFolder (see page 731)	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 (see page 534).
	LogInUser (see page 732)	Used to log in the user (see page 534).
	ResetMembershipUserPassword (see page 732)	
	UnsubscribeUser (see page 733)	Unsubscribes a user (see page 534) from the system.
	UpdateCustomPropertiesItemOrder (see page 734)	Updates the order of available custom properties.
	UpdateUser (see page 734)	
	UpDateUserGroup (see page 735)	Updates the user (see page 534) group details.
	UpdateUserPreferences (see page 736)	Updates the given user (see page 534)'s preference data.
	UserObject (see page 736)	Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Legend

	Method

ActivateUserAccount Method

Activates a deleted (deactivated) user into the current/active state using the UserId or the Username and GUID (AccountID) parameters.

This method is overloaded.

Overload List

	Name	Description
	User.ActivateUserAccount (int) (see page 708)	Activates a deleted (deactivated) user (see page 534) into the current/active state using the UserId. This function is overloaded. See Also: ActivateUserAccount(String, String)
	User.ActivateUserAccount (string, string) (see page 708)	Activates the deleted (deactivated) user (see page 534) into the current/active state using the Username and GUID (AccountID) parameter. This function is overloaded. See Also: ActivateUserAccount(Integer)

User.ActivateUserAccount Method (int)

Activates a deleted (deactivated) user (see page 534) 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(int UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "ActivateUserAccount By UserID", Description = "")>
Public Function ActivateUserAccount(UserId As Integer) As bool
```

Parameters

Parameters	Description
`UserId`	A unique ID for the required user (see page 534)

Returns

Boolean

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.ActivateUserAccount Method (string, string)

Activates the deleted (deactivated) user (see page 534) into the current/active state using the Username and GUID (AccountID) parameter.

This function is overloaded. See Also: ActivateUserAccount(Integer)

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 (see page 534)
`AccountID`	The GUID for the user (see page 534)

Returns

Boolean

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.AddMembershipUser Method

Creates new membership user (see page 534).

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 (see page 534) elements

Returns

Ektron.Cms.UserData (see page 801)

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.AddNewUser Method

Adds a new user (see page 534) 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 (see page 534) elements

Returns

Ektron.Cms.UserData (see page 801)

Remarks

A user (see page 534) 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 (see page 534)

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 of User (see page 706)'s info.

Returns

Ektron.Cms.UserData (see page 801)

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.AddUserGroup Method

Creates new user (see page 534) 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 118), otherwise CMS group assumed.
GroupDomain	Group domain, usually required for Active Directory (see page 118).

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.AddUserToGroup Method

Adds an existing user (see page 534) to an existing group.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "AddUserToGroup", Description = "")]
public void AddUserToGroup(int UserId, int GroupId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "AddUserToGroup", Description = "")>
Public Function AddUserToGroup(UserId As Integer, GroupId As Integer) As void
```

Parameters

Parameters	Description
UserId	The user (see page 534)'s ID.
GroupId	The required group's ID.

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.AutoLogInUser Method

Login (see page 229) the given user (see page 534) 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 (see page 534)'s name.
domain	The domain.
ServerName	The server name.

CreateBlankUserObject Method

This method is overloaded.

Overload List

	Name	Description
	User.CreateBlankUserObject () (see page 712)	Returns the instance of the user (see page 534) data. This function is overloaded. See Also: CreateBlankUserObject(UserData (see page 801))
	User.CreateBlankUserObject (ref UserData) (see page 713)	Sets the given user (see page 534) data values into the UserData (see page 801) class. This function is overloaded. See Also: CreateBlankUserObject

User.CreateBlankUserObject Method ()

Returns the instance of the user (see page 534) data.

This function is overloaded. See Also: CreateBlankUserObject(UserData (see page 801))

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 801)

User.CreateBlankUserObject Method (ref UserData)

Sets the given user (see page 534) data values into the UserData (see page 801) 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(UserData As ref UserData) As void
```

Parameters

Parameters	Description
UserData	an Object of user (see page 534) properties

User.DeleteUserByID Method

Removes the user (see page 534) from the system. This will not purge the data, but it disables the user (see page 534) from further accessing the system.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserByID", Description = "")]
public void DeleteUserByID(int UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserByID", Description = "")>
Public Function DeleteUserByID(UserId As Integer) As void
```

Parameters

Parameters	Description
UserId	The user (see page 534)'s ID.

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.DeleteUserByIds Method

Removes the user (see page 534) from the system. This will not purge the data, but it disables the user (see page 534) 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 (see page 534) IDs(comma separated values).

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.DeleteUserFromGroup Method

Removes a user (see page 534) from the designated group.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "DeleteUserFromGroup", Description = "")]
public void DeleteUserFromGroup(int UserId, int GroupId, bool ReturnDeleted);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserFromGroup", Description = "")>
Public Function DeleteUserFromGroup(UserId As Integer, GroupId As Integer, ReturnDeleted As
bool) As void
```

Parameters

Parameters	Description
UserId	The user (see page 534)'s ID.
GroupId	The group's ID.
ReturnDeleted	

Remarks

A user (see page 534) 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(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "DeleteUserGroup", Description = "")>
Public Function DeleteUserGroup(Id As Integer) As void
```

Parameters

Parameters	Description
Id	The group's ID.

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.EditUserCustomProperties Method

This method creates the editable fields(UI) for user (see page 534)'s custom properties.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "EditUserCustomProperties", Description = "")]
public string EditUserCustomProperties(int UserId, bool ReadOnly);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "EditUserCustomProperties", Description = "")>
Public Function EditUserCustomProperties(UserId As Integer, ReadOnly As bool) As string
```

Parameters

Parameters	Description
UserId	User (see page 706)'s ID. If the user (see page 534) ID is not specified then it will return empty fields.
ReadOnly	

Returns

String

User.GetActiveUser Method

Returns all of the user (see page 534) properties with or without preference.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetActiveUser", Description = "")]
public UserData GetActiveUser(int Id, bool Preference);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetActiveUser", Description = "")>
Public Function GetActiveUser(Id As Integer, Preference As bool) As UserData
```

Parameters

Parameters	Description
Id	The user (see page 534)'s ID.
Preference	Preference is returned only if the value true, otherwise nothing is returned for preference.

Returns

Ektron.Cms.UserData (see page 801)

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetADMapping Method

Loads the CMS and Active Directory (see page 118) name.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "GetADMapping", Description = "")]
public AdMappingData[] GetADMapping(int CallerId, string MapType, int Map, int Exclude, int
Editable);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetADMapping", Description = "")>
Public Function GetADMapping(CallerId As Integer, MapType As string, Map As Integer, Exclude
As Integer, Editable As Integer) As AdMappingData()
```

Parameters

Parameters	Description
`CallerId`	Who is calling this function.
`MapType`	Possible value is “userprop” (user (see page 534) properties).
`Map`	Ektron.Cms.Common.EkEnumeration.MapTypes (include, exclude and dontconsider are the possible values).
`Exclude`	Flag (0 – false, 1-true).
`Editable`	Flag(0-false,1-true).

Returns

Ektron.Cms.AdMappingData()

Remarks

A user (see page 534) 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. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData (see page 801)()

Description

Remarks

A user (see page 534) 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 (see page 534) with administrator privileges must be logged in to perform this operation.

GetAllUserGroups Method

Loads all of the user groups from the system. This method is overloaded.

Overload List

	Name	Description
	User.GetAllUserGroups (EkEnumeration.UserTypes, string) (see page 718)	Loads all of the user (see page 534) groups from the system. This function is overloaded. See Also: GetAllUserGroups(String)
	User.GetAllUserGroups (string) (see page 718)	Loads all of the user (see page 534) groups from the system. This function is overloaded. See Also: GetAllUserGroups(EkEnumeration.UserTypes, String)

User.GetAllUserGroups Method (EkEnumeration.UserTypes, string)

Loads all of the user (see page 534) 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 240); 0 = CMSUser
OrderBy	Arranges data in a specific order. Possible values are GroupName or GroupID.

Returns

Ektron.Cms.UserGroupData()

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetAllUserGroups Method (string)

Loads all of the user (see page 534) 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 (see page 534) with administrator privileges must be logged in to perform this operation.

GetAllUsers Method

This method is overloaded.

Overload List

	Name	Description
	User.GetAllUsers (int, int) (see page 719)	Gets the User (see page 706) Group Data of all the allowed users in the folder specified.
	User.GetAllUsers (ref UserRequestData) (see page 719)	Returns the list of users from the database based on the UserRequestData class.

User.GetAllUsers Method (int, int)

Gets the User (see page 706) 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(int CallerID, int FolderId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetAllUsers By CallerId and FolderID", Description = "")>
Public Function GetAllUsers(CallerID As Integer, FolderId As Integer) As UserGroupData()
```

Parameters

Parameters	Description
`CallerID`	The user (see page 534) ID that is logged on to the system.
`FolderId`	The ID of the folder.

Returns

Ektron.Cms.UserGroupData()

Remarks

User (see page 706) login with administrator privileges are required to perform this operation.

User.GetAllUsers Method (ref UserRequestData)

Returns the list of users from the database based on the UserRequestData class.

C#

```
[SoapHeader("AuthHeader")]
[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(userRequest As ref UserRequestData) As UserData()
```

Parameters

Parameters	Description
`userRequest`	list of constraints to load users

Returns

An array of users as UserData (see page 801).

User.GetGroupsUserIsIn Method

Loads all the groups by user (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetGroupsUserIsIn", Description = "")]
public GroupData[] GetGroupsUserIsIn(int UserId, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetGroupsUserIsIn", Description = "")>
Public Function GetGroupsUserIsIn(UserId As Integer, OrderBy As string) As GroupData()
```

Parameters

Parameters	Description
`UserId`	The user (see page 534)'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 (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetMembershipUser Method

Returns user (see page 534) data based on the user (see page 534)'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 706)'s login name (username). This is required

Returns

Ektron.Cms.UserData (see page 801)

Remarks

User (see page 706) login with administrator privileges are required to perform this operation.

User.GetUser Method

Loads the user (see page 534) details for the given user (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUser", Description = "")]
public UserData GetUser(int Id, bool Preference, bool ReturnDeleted);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUser", Description = "")>
Public Function GetUser(Id As Integer, Preference As bool, ReturnDeleted As bool) As UserData
```

Parameters

Parameters	Description
Id	The user (see page 534) ID.
Preference	Loads preference only if required (true = load preferences as a part of user (see page 534) data).

Returns

Ektron.Cms.UserData (see page 801)

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetUserbyUsername Method

Loads the user (see page 534) details for the given user (see page 534).

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 (see page 534) username.

Returns

Ektron.Cms.UserData (see page 801)

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetUserCustomProperties Method

Returns an array of custom properties for a given user (see page 534)'s ID.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserCustomProperties", Description = "")]
public CustomAttribute[] GetUserCustomProperties(int UserId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserCustomProperties", Description = "")>
Public Function GetUserCustomProperties(UserId As Integer) As CustomAttribute()
```

Parameters

Parameters	Description
`UserId`	User (see page 706)'s ID

Returns

Ektron.Cms.CustomAttribute()

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetUserEmailInfo Method

Returns user (see page 534) data for a given user (see page 534)'s ID.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserEmailInfo", Description = "")]
public UserData GetUserEmailInfo(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserEmailInfo", Description = "")>
Public Function GetUserEmailInfo(Id As Integer) As UserData
```

Parameters

Parameters	Description
`Id`	The user (see page 534)'s ID.

Returns

Object of UserData (see page 801)

Remarks

A user (see page 534) 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(int FolderId, int Id, bool Preference, bool
ReturnDeleted);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserForFolderAdmin", Description = "")>
Public Function GetUserForFolderAdmin(FolderId As Integer, Id As Integer, Preference As bool,
ReturnDeleted As bool) As UserData
```

Parameters

Parameters	Description
FolderId	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(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserGroup", Description = "")>
Public Function GetUserGroup(Id As Integer) As UserGroupData
```

Parameters

Parameters	Description
Id	The group's ID.

Returns

Ektron.Cms.UserGroupData

Remarks

A user (see page 534) 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(int FolderId, int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserGroupForFolderAdmin", Description = "")>
Public Function GetUserGroupForFolderAdmin(FolderId As Integer, Id As Integer) As UserGroupData
```

Parameters

Parameters	Description
FolderId	The ID of the folder.
Id	The ID of the group.

Returns

UserGroupData

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetUserPermissions Method

Loads the user (see page 534) permissions as a array of UserPermissionData.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPermissions", Description = "")]
public UserPermissionData[] GetUserPermissions(int Id, string ItemType, int UserId, string
UserGroupList, EkContent.PermissionUserType PermissionType, EkContent.PermissionRequestType
PermissionRequest);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPermissions", Description = "")>
Public Function GetUserPermissions(Id As Integer, ItemType As string, UserId As Integer,
UserGroupList As string, PermissionType As EkContent.PermissionUserType, PermissionRequest As
EkContent.PermissionRequestType) As UserPermissionData()
```

Parameters

Parameters	Description
Id	the item's ID
ItemType	The item's type
UserId	The user (see page 534)'s ID
UserGroupList	the group list
PermissionType	PermissionUserType
PermissionRequest	PermissionRequestType

Returns

Ektron.Cms.UserPermissionData()

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetUserPreference Method

Loads the preferences for the given user (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUserPreference", Description = "")]
public UserPreferenceData GetUserPreference(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUserPreference", Description = "")>
Public Function GetUserPreference(Id As Integer) As UserPreferenceData
```

Parameters

Parameters	Description
`Id`	The user (see page 534)'s ID.

Returns

Ektron.Cms.UserPreferenceData

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

GetUsers Method

An Overloaded function that allows you to get the users from a specific group or load and array of user group data. This method is overloaded.

Overload List

	Name	Description
	User.GetUsers (int, string) (see page 725)	Loads the all the users from the given group.
	User.GetUsers (string, string) (see page 726)	Returns an array of user (see page 534) group data based on the userName and userNameType.

User.GetUsers Method (int, string)

Loads the all the users from the given group.

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsers By GroupID and OrderBy", Description = "")]
public UserData[] GetUsers(int GroupId, string OrderBy);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsers By GroupID and OrderBy", Description = "")>
```

```
Public Function GetUsers(GroupId As Integer, OrderBy As string) As UserData()
```

Parameters

Parameters	Description
GroupId	The group's ID
OrderBy	The order of the users. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData (see page 801)()

Description

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetUsers Method (string, string)

Returns an array of user (see page 534) 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 (see page 534)'s name. (Required)
userNameTypes	The user (see page 534) type possible values are user (see page 534) or membership user (see page 534).

Returns

Ektron.Cms.UserGroupData()

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.GetUsersForTask Method

Loads the users that have permissions to the given content (see page 449).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "GetUsersForTask", Description = "")]
public UserGroupData[] GetUsersForTask(int CallerID, int ContentID);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsersForTask", Description = "")>
Public Function GetUsersForTask(CallerID As Integer, ContentID As Integer) As UserGroupData()
```

Parameters

Parameters	Description
`CallerID`	The user (see page 534) ID that is logged on to the system.
`ContentID`	The ID of the content (see page 449). (Required)

Returns

Ektron.Cms.UserGroupData()

Remarks

A user (see page 534) 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(int Id, string OrderBy, int GroupType);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "GetUsersNotInGroup", Description = "")>
Public Function GetUsersNotInGroup(Id As Integer, OrderBy As string, GroupType As Integer) As
UserData()
```

Parameters

Parameters	Description
`Id`	The group ID.
`OrderBy`	The order of the users. The string value can be: • username • firstname • lastname • userid • language • datecreated • lastlogindate

Returns

Ektron.Cms.UserData (see page 801)()

Description

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.IsAdmin Method

Determines the if the current user (see page 534) 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 (see page 534) is administrator. Otherwise, returns false.

IsARoleMember Method

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user. This method is overloaded.

Overload List

	Name	Description
	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) (see page 729)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 534). 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)
	User.IsARoleMember (Ektron.Cms.Common.EkEnumeration.CmsRoleIds, int, bool) (see page 729)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 534). This function is overloaded. See Also: • CommonApi.IsARoleMember Method (Ektron.Cms.Common.EkEnumeration.CmsRoleIds) • CommonApi.IsARoleMember Method (Integer, Integer, Boolean) • CommonApi.IsARoleMember Method (Integer)

	User.IsARoleMember (int) (see page 730)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 534). 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)
	User.IsARoleMember (int, int, bool) (see page 730)	Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 534). 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)

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 (see page 534).

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, int, bool)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 534).

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, int 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
Integer, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
RoleId	_nt_
GroupFlag	_nt_
nUserId	_nt_

User.IsARoleMember Method (int)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 534).

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(int RoleId);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by RoleID", Description = "")>
Public Function IsARoleMember(RoleId As Integer) As bool
```

Parameters

Parameters	Description
RoleId	_nt_

User.IsARoleMember Method (int, int, bool)

Roles provide the ability to offload/delegate some of the administrator's tasks onto other users, without giving full administration privileges to a user (see page 534).

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(int RoleId, int UserId, bool GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMember by RoleID, UserId and GroupFlag", Description = "")>
Public Function IsARoleMember(RoleId As Integer, UserId As Integer, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
RoleId	_nt_
GroupFlag	_nt_
nUserId	_nt_

User.IsARoleMemberForContent Method

Determines whether a Role Member is allowed to work with a given piece of content (see page 449) as an administrator or not. Administrations privileges are for the given piece of content (see page 449) 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 (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
[WebMethod(MessageName = "IsARoleMemberForContent", Description = "")]
public bool IsARoleMemberForContent(int RoleId, int ContentId, int UserId, bool GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMemberForContent", Description = "")>
Public Function IsARoleMemberForContent(RoleId As Integer, ContentId As Integer, UserId As
Integer, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
RoleId	
ContentId	The numeric content (see page 449) 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 (see page 534).

C#

```
[SoapHeader("AuthHeader")]
[SoapHeader("RequestInfoHeader")]
```

```
[WebMethod(MessageName = "IsARoleMemberForFolder", Description = "")]
public bool IsARoleMemberForFolder(int RoleId, int FolderId, int UserId, bool GroupFlag);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "IsARoleMemberForFolder", Description = "")>
Public Function IsARoleMemberForFolder(RoleId As Integer, FolderId As Integer, UserId As
Integer, GroupFlag As bool) As bool
```

Parameters

Parameters	Description
RoleId	
FolderId	The numeric folder ID for which you are giving the Role Member privileges.
UserId	
GroupFlag	

User.LogInUser Method

Used to log in the user (see page 534).

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 (see page 534)'s domain, required only when users login using Active Directory (see page 118).
Protocol	Required only when using a secure login or when you are using any valid Active Directory (see page 118) protocols.

Returns

Ektron.Cms.UserData (see page 801)

ResetMembershipUserPassword Method

Overload List

	Name	Description
	User.ResetMembershipUserPassword (string) (see page 733)	Reset the password for the given membership user (see page 534).
	User.ResetMembershipUserPassword (string, string) (see page 733)	Reset the password for the given membership user (see page 534).

User.ResetMembershipUserPassword Method (string)

Reset the password for the given membership user (see page 534).

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 706)'s username to be reset

Returns

String

Remarks

User (see page 706) login is required to perform this operation.

User.ResetMembershipUserPassword Method (string, string)

Reset the password for the given membership user (see page 534).

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 706)'s username to be reset
AccountId	If verification is turned on, specify verification id(email to a user (see page 534))

Returns

String

Remarks

User (see page 706) login is required to perform this operation.

User.UnsubscribeUser Method

Unsubscribes a user (see page 534) 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 (see page 534)'s username
`Password`	The user (see page 534)'s password

Returns

Boolean

Remarks

A user (see page 534) 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 (see page 534).

Returns

Boolean

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

UpdateUser Method

Overload List

	Name	Description
	User.UpdateUser (Collection) (see page 735)	Updates the user (see page 534) details.

	User.UpdateUser (UserData) (see page 735)	Updates the user (see page 534) details.

User.UpdateUser Method (Collection)

Updates the user (see page 534) 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 (see page 534) data.

Description

[WebMethod(MessageName = "UpdateUser", Description = "")]

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.UpdateUser Method (UserData)

Updates the user (see page 534) 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 801).

Remarks

User (see page 706) login with administrator privileges are required to perform this operation.

User.UpDateUserGroup Method

Updates the user (see page 534) 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 801).

Description

[WebMethod(MessageName = "UpDateUserGroup", Description = "")]

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

User.UpdateUserPreferences Method

Updates the given user (see page 534)'s preference data.

C#

```
public void UpdateUserPreferences(int Id, Collection data);
```

Visual Basic

```
Public Function UpdateUserPreferences(Id As Integer, data As Collection) As void
```

Parameters

Parameters	Description
Id	The user (see page 534)'s ID.
data	A collection of user (see page 534) preferences.

Description

[WebMethod(MessageName = "UpdateUserPreferences", Description = "")]

Remarks

A user (see page 534) with administrator privileges must be logged in to perform this operation.

UserObject Method

Returns user data based on the user's ID or by the user's login or username. This function is overloaded.

Overload List

	Name	Description
	User.UserObject (int) (see page 736)	Returns user (see page 534) data based on the user (see page 534)'s ID. This function is overloaded. See Also: UserObject(String)
	User.UserObject (string) (see page 737)	Returns user (see page 534) data based on the user (see page 534)'s login or username. This function is overloaded. See Also: UserObject(Integer)

User.UserObject Method (int)

Returns user (see page 534) data based on the user (see page 534)'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(int Id);
```

Visual Basic

```
<SoapHeader("AuthHeader")>
<SoapHeader("RequestInfoHeader")>
<WebMethod(MessageName = "UserObject By User ID", Description = "")>
Public Function UserObject(Id As Integer) As UserData
```

Parameters

Parameters	Description
Id	The user (see page 534)'s ID.

Returns

Ektron.Cms.UserData (see page 801)

User.UserObject Method (string)

Returns user (see page 534) data based on the user (see page 534)'s login or username.

This function is overloaded. See Also: UserObject(Integer)

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 706)'s login or username.

Returns

Ektron.Cms.UserData (see page 801)

Data Classes

Classes

	Name	Description
	ContentData (see page 738)	A data object that contains the properties that make up a content (see page 449) item.
	FolderData (see page 755)	This is class FolderData.
	FolderRequest (see page 770)	This is class FolderRequest.
	FormData (see page 781)	Describes a form design. Declaration Example: `Dim formItem As Ektron.Cms.FormData`
	FormFieldDataItem (see page 785)	Describes an instance of a data value from an individual form field. The data is a name/value pair.
	FormFieldDatalist (see page 786)	Describes an individual datalist suitable as an enumeration for a select list.
	FormFieldDatalistItem (see page 789)	Describes an individual datalist item (most importantly its value and display value).
	FormFieldDefinition (see page 791)	Describes the definition of an individual form field.
	FormFieldList (see page 794)	Describes the definition of all the fields in a specific form.
	FormRequestData (see page 796)	A data object that contains information about a form.
	FormSubmittedData (see page 799)	Describes the data of all the fields submitted from a form at a specific time.
	UserData (see page 801)	A data object that contains the properties that make up a user (see page 534).

Legend

	Class

ContentData Class

A data object that contains the properties that make up a content (see page 449) item.

Class Hierarchy

ContentData

C#

```
public class ContentData;
```

Visual Basic

```
Public Class ContentData
```

File

ContentData.vb

ContentData Properties

The properties of the ContentData class are listed here.

Public Properties

	Name	Description
	ApprovalMethod (see page 741)	Gets or sets the approval method of a content (see page 449) item for the ContentData (see page 738) object. Choices are: • **1** = Force All Approvers - content (see page 449) checked-out during the approval process will force the approval chain to reset to the beginning. • **0** = Do Not Force All Approvers - content (see page 449) checked-out during the approval process will continue through the approval chain.
	Approver (see page 742)	Gets or sets the approvers of a content (see page 449) item for the ContentData (see page 738) object.
	AssetData (see page 742)	Gets or sets a content (see page 449) item's asset data information when the content (see page 449) item is a managed file, Office Document, Multimedia file or Open Office document. This is for the ContentData (see page 738) object.
	Comment (see page 742)	Gets or sets comment information associated with a content (see page 449) item for the ContentData (see page 738) object.
	ContType (see page 742)	Gets or sets the content (see page 449) type as an integer for the ContentData (see page 738) object. Choices are: • **-1** = AllTypes • **1** = Content (see page 449) • **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
	DateCreated (see page 743)	Gets or sets the creation date of a content (see page 449) item for the ContentData (see page 738) object.
	DisplayDateCreated (see page 743)	Gets or sets the creation date of a content (see page 449) item formatted for display for the ContentData (see page 738) object. For example, `June 11, 2007`.
	DisplayEndDate (see page 743)	Gets or sets the end date of a content (see page 449) item formatted for display. This is for the ContentData (see page 738) object. For example, `June 11, 2007`.
	DisplayGoLive (see page 744)	Gets or sets the start date a content (see page 449) item goes live on the site formatted for display. This is for the ContentData (see page 738) object. For example, `June 11, 2007`.
	DisplayLastEditDate (see page 744)	Gets or sets the content (see page 449)'s last edit date formatted for display for the ContentData (see page 738) object. For example, `June 11, 2007`.
	EditorFirstName (see page 744)	Gets or sets the content (see page 449) editor's first name for the ContentData (see page 738) object.
	EditorLastName (see page 745)	Gets or sets the content (see page 449) editor's last name for the ContentData (see page 738) object.
	EditorUserNames (see page 745)	Gets or sets the username of the last editor of the content (see page 449) item for the ContentData (see page 738) object.

	EndDate (see page 745)	Gets or sets the end date of a content (see page 449) item. This is for the ContentData (see page 738) object.
	EndDateAction (see page 745)	Gets or sets the action to perform on a content (see page 449) item once the End Date has been reached. This is for the ContentData (see page 738) object. Choices are: • **1** = Archive and remove from site (expire) • **2** = Archive and remain on site • **3** = Add to the CMS Refresh Report
	FolderId (see page 746)	Gets or Sets the content (see page 449) item's folder ID for the ContentData (see page 738) object.
	FolderName (see page 746)	Gets or sets the content (see page 449) item's folder name for the ContentData (see page 738) object.
	GoLive (see page 746)	Gets or sets the start date a content (see page 449) item goes live on the site. This is for the ContentData (see page 738) object.
	HistoryId (see page 746)	Gets or sets the content (see page 449) item's history ID for the ContentData (see page 738) object.
	Html (see page 747)	Gets or sets the content (see page 449) item's HTML for the ContentData (see page 738) object.
	HyperLink (see page 747)	Gets or sets the content (see page 449) item's Hyperlink in the CMS400.NET for the ContentData (see page 738) object.
	Id (see page 747)	Gets or sets the ID of the content (see page 449) item for the ContentData (see page 738) object.
	Image (see page 748)	Gets or sets the path to an image in the library that you want to associate with the content (see page 449) for the ContentData (see page 738) object.
	ImageThumbnail (see page 748)	Gets or sets the path to a thumbnail image in the library that you want to associate with the content (see page 449) for the ContentData (see page 738) object.
	InheritedFrom (see page 748)	Gets or sets the folder ID from which to inherit permissions for the ContentData (see page 738) object.
	IsInherited (see page 748)	Gets or sets whether the content (see page 449) item inherits permissions from the folder settings for the ContentData (see page 738) object.
	IsPrivate (see page 749)	Gets or sets whether a content (see page 449) item is private and not viewable on the public Web site. Set to True to make a content (see page 449) item private for the ContentData (see page 738) object.
	IsPublished (see page 749)	Gets or sets whether the content (see page 449) is in a published state for the ContentData (see page 738) object.
	IsSearchable (see page 749)	Gets or sets whether the content (see page 449) is searchable for the ContentData (see page 738) object. True = the content (see page 449) searchable.
	IsXmlInherited (see page 750)	Gets or sets whether the XML Smart Form (see page 493) configuration is inherited from the folder. True = XML Smart Form (see page 493) configuration is inherited from the folder. This is for the ContentData (see page 738) object.
	LanguageId (see page 750)	Get or sets the language ID of the content (see page 449) item for the ContentData (see page 738) object.
	LastEditDate (see page 750)	Gets or sets the content (see page 449)'s last edit date for the ContentData (see page 738) object.
	ManualAlias (see page 750)	Gets or sets the manual alias by its name for the ContentData (see page 738) object. For example, `/~YourSite~/text.aspx`.
	ManualAliasId (see page 751)	Gets or sets the ID of the of a manual alias for the ContentData (see page 738) object.
	MetaData (see page 751)	Gets or sets the Metadata of a content (see page 449) item as an array of ContentMetadata. This is for the ContentData (see page 738) object.
	Path (see page 751)	Gets or sets the CMS400.NET path to content (see page 449) item for the ContentData (see page 738) object.
	Permissions (see page 751)	Gets or sets permission data for the ContentData (see page 738) object.
	Quicklink (see page 752)	Gets or sets the content (see page 449) item's Quicklink in the CMS400.NET for the ContentData (see page 738) object.

	Status (see page 752)	Gets or sets the letter value representing the status of a content (see page 449) item for the ContentData (see page 738) 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
	StyleSheet (see page 752)	Gets or sets the style sheet to use with a content (see page 449) item for the ContentData (see page 738) object.
	Teaser (see page 753)	Gets or sets a content (see page 449)'s summary information for the ContentData (see page 738) object.
	TemplateConfiguration (see page 753)	Gets or sets the content (see page 449) item's template configuration for the ContentData (see page 738) object.
	Title (see page 753)	Gets or sets the content (see page 449) item's title for the ContentData (see page 738) object.
	UserId (see page 754)	Gets or sets the ID of the user (see page 534) for the ContentData (see page 738) object.
	UserName (see page 754)	Gets or sets the content (see page 449)'s username for the ContentData (see page 738) object.
	XmlConfiguration (see page 754)	Gets or sets the content (see page 449) item's XML configuration for the ContentData (see page 738) object.
	XmlInheritedFrom (see page 754)	Gets or sets the ID of the folder from which the XML Smart Form (see page 493) configuration is inherited. This is for the ContentData (see page 738) object.

Legend

	Property

ContentData.ApprovalMethod Property

Gets or sets the approval method of a content (see page 449) item for the ContentData (see page 738) object. Choices are:

- **1** = Force All Approvers - content (see page 449) checked-out during the approval process will force the approval chain to reset to the beginning.
- **0** = Do Not Force All Approvers - content (see page 449) 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 (see page 449) is inheriting the approval method from the folder, the IsInherited (see page 748) property must be false to change this property.

ContentData.Approver Property

Gets or sets the approvers of a content (see page 449) item for the ContentData (see page 738) object.

C#

```
public String Approver;
```

Visual Basic

```
Public Property Approver() As String
```

Returns

A string value representing the approver(s) of a content (see page 449) item.

ContentData.AssetData Property

Gets or sets a content (see page 449) item's asset data information when the content (see page 449) item is a managed file, Office Document, Multimedia file or Open Office document. This is for the ContentData (see page 738) object.

C#

```
public Ektron.Cms.Common.AssetData AssetData;
```

Visual Basic

```
Public Property AssetData() As Ektron.Cms.Common.AssetData
```

Returns

Returns content (see page 449) item's asset data information as Ektron.Cms.Common.AssetData.

ContentData.Comment Property

Gets or sets comment information associated with a content (see page 449) item for the ContentData (see page 738) object.

C#

```
public String Comment;
```

Visual Basic

```
Public Property Comment() As String
```

Returns

Comment information associated with a content (see page 449) item.

ContentData.ContType Property

Gets or sets the content (see page 449) type as an integer for the ContentData (see page 738) object. Choices are:

- **-1** = AllTypes
- **1** = Content (see page 449)
- **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

C#

```
public int ContType;
```

Visual Basic

```
Public Property ContType() As Integer
```

Returns

An integer representing the content (see page 449) type.

ContentData.DateCreated Property

Gets or sets the creation date of a content (see page 449) item for the ContentData (see page 738) object.

C#

```
public DateTime DateCreated;
```

Visual Basic

```
Public Property DateCreated() As DateTime
```

Returns

The creation date of a content (see page 449) item as DateTime.

ContentData.DisplayDateCreated Property

Gets or sets the creation date of a content (see page 449) item formatted for display for the ContentData (see page 738) object. For example, `June 11, 2007`.

C#

```
public String DisplayDateCreated;
```

Visual Basic

```
Public Property DisplayDateCreated() As String
```

Returns

Creation date of a content (see page 449) item formatted for display.

ContentData.DisplayEndDate Property

Gets or sets the end date of a content (see page 449) item formatted for display. This is for the ContentData (see page 738) object. For example, `June 11, 2007`.

C#

```
public String DisplayEndDate;
```

Visual Basic

```
Public Property DisplayEndDate() As String
```

Returns

End date of a content (see page 449) item formatted for display.

ContentData.DisplayGoLive Property

Gets or sets the start date a content (see page 449) item goes live on the site formatted for display. This is for the ContentData (see page 738) object. For example, `June 11, 2007`.

C#

```
public String DisplayGoLive;
```

Visual Basic

```
Public Property DisplayGoLive() As String
```

Returns

Start date a content (see page 449) item goes live on the site formatted for display.

ContentData.DisplayLastEditDate Property

Gets or sets the content (see page 449)'s last edit date formatted for display for the ContentData (see page 738) object. For example, `June 11, 2007`.

C#

```
public String DisplayLastEditDate;
```

Visual Basic

```
Public Property DisplayLastEditDate() As String
```

Returns

Content (see page 449)'s last edit date formatted for display.

ContentData.EditorFirstName Property

Gets or sets the content (see page 449) editor's first name for the ContentData (see page 738) 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 (see page 449) editor's last name for the ContentData (see page 738) 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 (see page 449) item for the ContentData (see page 738) object.

C#

```
public String EditorUserNames;
```

Visual Basic

```
Public Property EditorUserNames() As String
```

Returns

Username of the last editor of the content (see page 449) item.

ContentData.EndDate Property

Gets or sets the end date of a content (see page 449) item. This is for the ContentData (see page 738) object.

C#

```
public String EndDate;
```

Visual Basic

```
Public Property EndDate() As String
```

Returns

A string value that represents the end date of a content (see page 449) item.

ContentData.EndDateAction Property

Gets or sets the action to perform on a content (see page 449) item once the End Date has been reached. This is for the ContentData (see page 738) 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 (see page 449) item's folder ID for the ContentData (see page 738) object.

C#

```
public int FolderId;
```

Visual Basic

```
Public Property FolderId() As Integer
```

Returns

Folder (see page 383) ID as an Integer.

ContentData.FolderName Property

Gets or sets the content (see page 449) item's folder name for the ContentData (see page 738) object.

C#

```
public String FolderName;
```

Visual Basic

```
Public Property FolderName() As String
```

Returns

The content (see page 449) item's folder name.

ContentData.GoLive Property

Gets or sets the start date a content (see page 449) item goes live on the site. This is for the ContentData (see page 738) object.

C#

```
public String GoLive;
```

Visual Basic

```
Public Property GoLive() As String
```

Returns

A string value that represents the date a content (see page 449) item goes live.

ContentData.HistoryId Property

Gets or sets the content (see page 449) item's history ID for the ContentData (see page 738) object.

C#

```
public int HistoryId;
```

Visual Basic

```
Public Property HistoryId() As Integer
```

Returns

Integer ID of the content (see page 449) item's history.

ContentData.Html Property

Gets or sets the content (see page 449) item's HTML for the ContentData (see page 738) object.

C#

```
public String Html;
```

Visual Basic

```
Public Property Html() As String
```

Returns

A string value that represents the HTML that makes up a content (see page 449) item.

ContentData.HyperLink Property

Gets or sets the content (see page 449) item's Hyperlink in the CMS400.NET for the ContentData (see page 738) object.

C#

```
public String HyperLink;
```

Visual Basic

```
Public Property HyperLink() As String
```

Returns

A content (see page 449) item's Hyperlink information.

ContentData.Id Property

Gets or sets the ID of the content (see page 449) item for the ContentData (see page 738) object.

C#

```
public int Id;
```

Visual Basic

```
Public Property Id() As Integer
```

Returns

Integer ID of the content (see page 449) item.

ContentData.Image Property

Gets or sets the path to an image in the library that you want to associate with the content (see page 449) for the ContentData (see page 738) object.

C#

```
public String Image;
```

Visual Basic

```
Public Property Image() As String
```

Returns

Path (see page 751) to the image in the library.

ContentData.ImageThumbnail Property

Gets or sets the path to a thumbnail image in the library that you want to associate with the content (see page 449) for the ContentData (see page 738) object.

C#

```
public String ImageThumbnail;
```

Visual Basic

```
Public Property ImageThumbnail() As String
```

Returns

Path (see page 751) to the thumbnail image in the library.

ContentData.InheritedFrom Property

Gets or sets the folder ID from which to inherit permissions for the ContentData (see page 738) object.

C#

```
public int InheritedFrom;
```

Visual Basic

```
Public Property InheritedFrom() As Integer
```

Returns

An integer ID representing the folder from which permissions are inherited.

ContentData.IsInherited Property

Gets or sets whether the content (see page 449) item inherits permissions from the folder settings for the ContentData (see page 738) object.

C#

```
public Boolean IsInherited;
```

Visual Basic

```
Public Property IsInherited() As Boolean
```

Returns

A boolean that returns true when the content (see page 449) item inherits permissions from the folder settings.

ContentData.IsPrivate Property

Gets or sets whether a content (see page 449) item is private and not viewable on the public Web site. Set to True to make a content (see page 449) item private for the ContentData (see page 738) object.

C#

```
public Boolean IsPrivate;
```

Visual Basic

```
Public Property IsPrivate() As Boolean
```

Returns

Returns true if the content (see page 449) item is set to private.

ContentData.IsPublished Property

Gets or sets whether the content (see page 449) is in a published state for the ContentData (see page 738) object.

C#

```
public Boolean IsPublished;
```

Visual Basic

```
Public Property IsPublished() As Boolean
```

Returns

Returns a boolean based on whether the content (see page 449) is in a published state. True = Published

ContentData.IsSearchable Property

Gets or sets whether the content (see page 449) is searchable for the ContentData (see page 738) object. True = the content (see page 449) searchable.

C#

```
public Boolean IsSearchable;
```

Visual Basic

```
Public Property IsSearchable() As Boolean
```

Returns

A boolean based on whether the content (see page 449) is searchable.

- **True** = content (see page 449) is searchable.
- **False** = content (see page 449) is not searchable.

Description

ContentData.IsXmlInherited Property

Gets or sets whether the XML Smart Form (see page 493) configuration is inherited from the folder. True = XML Smart Form (see page 493) configuration is inherited from the folder. This is for the ContentData (see page 738) object.

C#

```
public Boolean IsXmlInherited;
```

Visual Basic

```
Public Property IsXmlInherited() As Boolean
```

Returns

A boolean based on whether the XML Smart Form (see page 493) configuration is inherited from the folder.

- **True** = content (see page 449) XML Smart Form (see page 493) configuration is inherited from the folder.
- **False** = content (see page 449) XML Smart Form (see page 493) configuration is not inherited from the folder.

ContentData.LanguageId Property

Get or sets the language ID of the content (see page 449) item for the ContentData (see page 738) object.

C#

```
public int LanguageId;
```

Visual Basic

```
Public Property LanguageId() As Integer
```

Returns

Language ID as an integer. For example, 1033 (English).

ContentData.LastEditDate Property

Gets or sets the content (see page 449)'s last edit date for the ContentData (see page 738) object.

C#

```
public String LastEditDate;
```

Visual Basic

```
Public Property LastEditDate() As String
```

Returns

Content (see page 449)'s last edit date as a string.

ContentData.ManualAlias Property

Gets or sets the manual alias by its name for the ContentData (see page 738) 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 738) object.

C#

```
public int ManualAliasId;
```

Visual Basic

```
Public Property ManualAliasId() As Integer
```

Returns

Integer ID representing a manual alias.

ContentData.MetaData Property

Gets or sets the Metadata of a content (see page 449) item as an array of ContentMetadata. This is for the ContentData (see page 738) 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 (see page 449) item for the ContentData (see page 738) object.

C#

```
public String Path;
```

Visual Basic

```
Public Property Path() As String
```

Returns

A string value representing the CMS400.NET path to the content (see page 449) item.

ContentData.Permissions Property

Gets or sets permission data for the ContentData (see page 738) object.

C#

```
public PermissionData Permissions;
```

Visual Basic

```
Public Property Permissions() As PermissionData
```

Returns

Permissions as a PermissionData object.

ContentData.Quicklink Property

Gets or sets the content (see page 449) item's Quicklink in the CMS400.NET for the ContentData (see page 738) object.

C#

```
public String Quicklink;
```

Visual Basic

```
Public Property Quicklink() As String
```

Returns

A content (see page 449) item's Quicklink information.

ContentData.Status Property

Gets or sets the letter value representing the status of a content (see page 449) item for the ContentData (see page 738) 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 (see page 449) item.

ContentData.StyleSheet Property

Gets or sets the style sheet to use with a content (see page 449) item for the ContentData (see page 738) 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 (see page 449) item.

ContentData.Teaser Property

Gets or sets a content (see page 449)'s summary information for the ContentData (see page 738) object.

C#

```
public String Teaser;
```

Visual Basic

```
Public Property Teaser() As String
```

Returns

Summary information for the content (see page 449).

ContentData.TemplateConfiguration Property

Gets or sets the content (see page 449) item's template configuration for the ContentData (see page 738) object.

C#

```
public TemplateData TemplateConfiguration;
```

Visual Basic

```
Public Property TemplateConfiguration() As TemplateData
```

Returns

A content (see page 449) item's template configuration as TemplateData.

ContentData.Title Property

Gets or sets the content (see page 449) item's title for the ContentData (see page 738) object.

C#

```
public String Title;
```

Visual Basic

```
Public Property Title() As String
```

Returns

Title of the content (see page 449) item.

ContentData.UserId Property

Gets or sets the ID of the user (see page 534) for the ContentData (see page 738) object.

C#

```
public int UserId;
```

Visual Basic

```
Public Property UserId() As Integer
```

Returns

User (see page 534) ID as an integer.

ContentData.UserName Property

Gets or sets the content (see page 449)'s username for the ContentData (see page 738) object.

C#

```
public String UserName;
```

Visual Basic

```
Public Property UserName() As String
```

Returns

The content (see page 449)'s username.

ContentData.XmlConfiguration Property

Gets or sets the content (see page 449) item's XML configuration for the ContentData (see page 738) object.

C#

```
public XmlConfigData XmlConfiguration;
```

Visual Basic

```
Public Property XmlConfiguration() As XmlConfigData
```

Returns

A content (see page 449) 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 493) configuration is inherited. This is for the ContentData (see page 738) object.

C#

```
public int XmlInheritedFrom;
```

Visual Basic

```
Public Property XmlInheritedFrom() As Integer
```

Returns

An integer ID of the folder from which the XML Smart Form (see page 493) configuration is inherited.

FolderData Class

Class Hierarchy

FolderData

C#

```
public class FolderData;
```

Visual Basic

```
Public Class FolderData
```

File

FolderData.vb

Description

This is class FolderData.

FolderData Properties

The properties of the FolderData class are listed here.

Public Properties

	Name	Description
	ApprovalMethod (see page 757)	Gets or set whether or not to force all approvers to approve the content (see page 449) that is added to the folder. This is for the FolderData (see page 755) object. **0** = Do Not Force All Approvers - content (see page 449) checked-out during the approval process will continue through the approval chain. **1** = Force All Approvers - content (see page 449) checked-out during the approval process forces the approval chain to reset to the beginning.
	CategoryRequired (see page 758)	Gets or sets whether at least one taxonomy category selection is required when creating or editing content (see page 449). This is for the FolderData (see page 755) object. **True** = one taxonomy category is required when creating or editing content (see page 449). **False** = a taxonomy category is not required when creating or editing content (see page 449).
	ChildFolders (see page 758)	Gets or sets the child folders of a parent folder as an array of FolderData (see page 755). This is for the FolderData (see page 755) object.
	Description (see page 758)	Gets or sets a folder's description information for the FolderData (see page 755) object. This is for the FolderData (see page 755) object.
	DomainProduction (see page 759)	Gets or sets the domain of the production server when using the site replication feature. For example: `www.yourektroncmssite.com` This is for the FolderData (see page 755) object.
	DomainStaging (see page 759)	Gets or sets the domain of the staging server when using the site replication feature. For example: `staging.yourektroncmssite.com` This is for the FolderData (see page 755) object.
	FolderTaxonomy (see page 759)	Gets or sets the taxonomy for a folder as an array of TaxonomyBaseData. This is for the FolderData (see page 755) object.

	FolderType (see page 760)	Gets or sets the type of folder in the Workarea. The `FolderType` property can be used to verify if a given folder accepts the type of content (see page 449) being submitted. This is for the FolderData (see page 755) object. Available Types: • **0** = Content (see page 449) - a content (see page 449) folder that contains assets, documents and HTML content (see page 449). • **1** = Blog (see page 454) - designates the folder for use with the Blogging feature. • **2** = Domain - designates the folder as a domain folder when using the Multi-site feature. • **3** = Discussion Board - designates the folder for use as a Threaded Discussion folder. • **4** = Discussion Forum (see page 167)... more (see page 760)
	HasChildren (see page 760)	Gets or sets whether the folder has child folders. This is for the FolderData (see page 755) object. **True** = the folder has child folders. **False** = the folder does not child folders.
	Id (see page 760)	Gets or sets the ID of the Folder (see page 383) for the FolderData (see page 755) object.
	Inherited (see page 761)	Gets or sets whether the folder's permission are inherited from the the parent folder. This is for the FolderData (see page 755) object. **True** = inherit the parent folder's permissions. **False** = do not inherit the parent folder's permissions.
	InheritedFrom (see page 761)	Gets or sets the ID of the parent folder from which the folder is inheriting. This is for the FolderData (see page 755) object.
	IsCommunityFolder (see page 761)	Marks a folder as being a community folder. A community folder allows the editing of content (see page 449) by membership users. This is for the FolderData (see page 755) object. **True** = folder is a community folder. **False** = folder is not community folder.
	IsDomainFolder (see page 762)	Gets or sets whether or not the folder is a domain folder. This is for the FolderData (see page 755) object. **True** = folder is a domain folder. **False** = folder is not a domain folder.
	MetaInherited (see page 762)	Get or set whether the metadata is inherited from the parent folder. This is for the FolderData (see page 755) object. **True** = the folder inherits metadata from the parent folder. **False** = the folder does not inherit metadata from the parent folder.
	MetaInheritedFrom (see page 762)	Gets or sets the parent folder ID from which to inherit metadata information. This is for the FolderData (see page 755) object.
	Name (see page 762)	Gets or sets the folder's name for the FolderData (see page 755) object.
	NameWithPath (see page 763)	Gets or sets the folder's name with its path. This path is the tree structure in the Workarea, not a folders file system path. For example: If you want the **Medication Management** folder in the Ektron Medical demo site Workarea, it would appear as `\~YourSite~\Products\Medication Management`
	ParentId (see page 763)	Gets or sets the folder's parent folder ID for the FolderData (see page 755) object.
	Permissions (see page 763)	Gets or sets a list of folder permissions for the logged in user (see page 534). This is for the FolderData (see page 755) object. If you use Ektron.Cms.API (see page 363) to access this information, there is an internal Admin user (see page 534) and all permissions are returned. As an alternative, use Ektron.Cms.ContentAPI.
	PrivateContent (see page 764)	Gets or sets whether the content (see page 449) in a folder is private. This is for the FolderData (see page 755) object. **True** = the folder's contents private. **False** = the folder's contents are public.
	PublishHtmlActive (see page 764)	Get or set whether assets in the folder are published as HTML. This is for the FolderData (see page 755) object. **True** = assets in the folder are published as HTML. **False** = assets are not published as HTML.
	PublishPdfActive (see page 764)	Get or set whether assets in the folder are published as a PDF. This is for the FolderData (see page 755) object. **True** = assets in the folder are published as a PDF. **False** = assets are not published as a PDF.
	PublishPdfEnabled (see page 765)	Get or set whether the folder allows assets to be published in another format. For example, as PDF or HTML. This is for the FolderData (see page 755) object. **True** = assets in the folder can be published in another format. **False** = assets are not published in another format.

	ReplicationMethod (see page 765)	Gets or sets whether or not the content (see page 449) in the folder is replicated via quick deploy. The default is 0 (zero). This is for the FolderData (see page 755) object. • **0** = do not replicate content (see page 449). This is the default. • **1** = replicate content (see page 449) via the Quick Deploy feature.
	SitemapInherited (see page 765)	Gets or sets whether or not the Sitemap (see page 301) information for this folder is inherited. This is for the FolderData (see page 755) object. • **0** (Zero) = False • **1** = True
	SitemapInheritedFrom (see page 766)	Gets or sets the integer ID of the folder from which the Sitemap (see page 301) information is inherited. This is for the FolderData (see page 755) object.
	SitemapPath (see page 766)	Gets or sets the SitemapPath as an array of Common.SitemapPath. This is for the FolderData (see page 755) object.
	StyleSheet (see page 766)	Gets or sets a style sheet that defines the styles of the content (see page 449) within the content (see page 449) folder. This is for the FolderData (see page 755) object.
	StyleSheetInherited (see page 767)	Gets or sets the whether the style sheet is inherited from the parent folder. This is for the FolderData (see page 755) object. **True** = the folder inherits style sheet information from the parent folder. **False** = the folder does not inherit style sheet information from the parent folder.
	TaxonomyInherited (see page 767)	Gets or sets whether taxonomy is inherited from the parent folder. This is for the FolderData (see page 755) object. **True** = taxonomy is inherited from the parent folder. **False** = taxonomy is not inherited from the parent folder.
	TaxonomyInheritedFrom (see page 767)	Gets or sets the ID of the parent folder from which the taxonomy is inherited. This is for the FolderData (see page 755) object.
	TemplateFileName (see page 768)	Gets or sets a template name for the folder. Content (see page 449) in this folder appears on your Web site within the template you choose. This is for the FolderData (see page 755) object.
	TemplateId (see page 768)	Gets or sets the folder template's ID for the FolderData (see page 755) object.
	TemplateInherited (see page 768)	Gets or sets the whether the template is inherited from the parent folder. This is for the FolderData (see page 755) object. **True** = the folder inherits the template from the parent folder. **False** = the folder does not inherit the template from the parent folder.
	TotalContent (see page 768)	Gets or sets the total number of content (see page 449) items in the folder. This is for the FolderData (see page 755) object.
	UpdatedContent (see page 769)	Gets or sets the total number of content (see page 449) items that have been updated. This is for the FolderData (see page 755) object.
	XmlConfiguration (see page 769)	Gets or sets the XML configuration for a folder. This is for the FolderData (see page 755) object.
	XmlInherited (see page 769)	Gets or sets whether the folder inherits the XML configuration from the parent folder. This is for the FolderData (see page 755) object. **True** = the folder inherits the XML configuration from the parent folder. **False** = the folder does not inherit the XML configuration from the parent folder.
	XmlInheritedFrom (see page 770)	Gets or sets the parent folder ID from which the XML is inherited. This is for the FolderData (see page 755) object.

Legend

	Property

FolderData.ApprovalMethod Property

Gets or set whether or not to force all approvers to approve the content (see page 449) that is added to the folder. This is for the FolderData (see page 755) object.

0 = Do Not Force All Approvers - content (see page 449) checked-out during the approval process will continue through the approval chain.

1 = Force All Approvers - content (see page 449) checked-out during the approval process forces the approval chain to reset to the beginning.

C#

```
public int ApprovalMethod;
```

Visual Basic

```
Public Property ApprovalMethod() As Integer
```

Returns

Returns 0 (zero) if you are not forcing all approvers.

Returns 1 (one) if you are forcing all approvers.

FolderData.CategoryRequired Property

Gets or sets whether at least one taxonomy category selection is required when creating or editing content (see page 449). This is for the FolderData (see page 755) object.

True = one taxonomy category is required when creating or editing content (see page 449).

False = a taxonomy category is not required when creating or editing content (see page 449).

C#

```
public Boolean CategoryRequired;
```

Visual Basic

```
Public Property CategoryRequired() As Boolean
```

Returns

Returns true if a taxonomy category is required when creating of publishing content (see page 449).

FolderData.ChildFolders Property

Gets or sets the child folders of a parent folder as an array of FolderData (see page 755). This is for the FolderData (see page 755) object.

C#

```
public FolderData() ChildFolders;
```

Visual Basic

```
Public Property ChildFolders() As FolderData()
```

Returns

Child folders as an array of FolderData (see page 755).

FolderData.Description Property

Gets or sets a folder's description information for the FolderData (see page 755) object. This is for the FolderData (see page 755) object.

C#

```
public String Description;
```

Visual Basic

```
Public Property Description() As String
```

Returns

The folder's description.

FolderData.DomainProduction Property

Gets or sets the domain of the production server when using the site replication feature. For example:

`www.yourektroncmssite.com`

This is for the FolderData (see page 755) object.

C#

```
public String DomainProduction;
```

Visual Basic

```
Public Property DomainProduction() As String
```

Returns

The production server domain.

FolderData.DomainStaging Property

Gets or sets the domain of the staging server when using the site replication feature. For example:

`staging.yourektroncmssite.com`

This is for the FolderData (see page 755) object.

C#

```
public String DomainStaging;
```

Visual Basic

```
Public Property DomainStaging() As String
```

Returns

The staging server domain.

FolderData.FolderTaxonomy Property

Gets or sets the taxonomy for a folder as an array of TaxonomyBaseData. This is for the FolderData (see page 755) object.

C#

```
public TaxonomyBaseData() FolderTaxonomy;
```

Visual Basic

```
Public Property FolderTaxonomy() As TaxonomyBaseData()
```

Returns

An array of TaxononmyBaseData.

FolderData.FolderType Property

Gets or sets the type of folder in the Workarea. The `FolderType` property can be used to verify if a given folder accepts the type of content (see page 449) being submitted. This is for the FolderData (see page 755) object.

Available Types:

- **0** = Content (see page 449) - a content (see page 449) folder that contains assets, documents and HTML content (see page 449).
- **1** = Blog (see page 454) - designates the folder for use with the Blogging feature.
- **2** = Domain - designates the folder as a domain folder when using the Multi-site feature.
- **3** = Discussion Board - designates the folder for use as a Threaded Discussion folder.
- **4** = Discussion Forum (see page 167) - a subfolder of the Threaded Discussion folder.
- **5** = Root - Specifies a folder as root folder. This option is not available.
- **6** = Community - Membership (see page 240) users can add and edit content (see page 449) contained in these folders.

C#

```
public int FolderType;
```

Visual Basic

```
Public Property FolderType() As Integer
```

Returns

An integer value that equals a type of folder. Folder (see page 383) types are listed above.

FolderData.HasChildren Property

Gets or sets whether the folder has child folders. This is for the FolderData (see page 755) object.

True = the folder has child folders.

False = the folder does not child folders.

C#

```
public Boolean HasChildren;
```

Visual Basic

```
Public Property HasChildren() As Boolean
```

Returns

Returns true if folder has child folders. Otherwise, returns false.

FolderData.Id Property

Gets or sets the ID of the Folder (see page 383) for the FolderData (see page 755) object.

C#

```
public int Id;
```

Visual Basic

```
Public Property Id() As Integer
```

Returns

The folder's Integer ID.

FolderData.Inherited Property

Gets or sets whether the folder's permission are inherited from the the parent folder. This is for the FolderData (see page 755) object.

True = inherit the parent folder's permissions.

False = do not inherit the parent folder's permissions.

C#

```
public Boolean Inherited;
```

Visual Basic

```
Public Property Inherited() As Boolean
```

Returns

Returns true if the folder is inheriting from its parent. Otherwise, returns false.

FolderData.InheritedFrom Property

Gets or sets the ID of the parent folder from which the folder is inheriting. This is for the FolderData (see page 755) object.

C#

```
public int InheritedFrom;
```

Visual Basic

```
Public Property InheritedFrom() As Integer
```

Returns

The integer ID of the parent folder.

FolderData.IsCommunityFolder Property

Marks a folder as being a community folder. A community folder allows the editing of content (see page 449) by membership users. This is for the FolderData (see page 755) object.

True = folder is a community folder.

False = folder is not community folder.

C#

```
public Boolean IsCommunityFolder;
```

Visual Basic

```
Public Property IsCommunityFolder() As Boolean
```

Returns

Returns true if the folder is a community folder. Otherwise, returns false.

FolderData.IsDomainFolder Property

Gets or sets whether or not the folder is a domain folder. This is for the FolderData (see page 755) object.

True = folder is a domain folder.

False = folder is not a domain folder.

C#

```
public Boolean IsDomainFolder;
```

Visual Basic

```
Public Property IsDomainFolder() As Boolean
```

Returns

Returns true if the folder is a domain folder. Otherwise, returns false.

FolderData.MetaInherited Property

Get or set whether the metadata is inherited from the parent folder. This is for the FolderData (see page 755) object.

True = the folder inherits metadata from the parent folder.

False = the folder does not inherit metadata from the parent folder.

C#

```
public int MetaInherited;
```

Visual Basic

```
Public Property MetaInherited() As Integer
```

Returns

Returns true if metadata is inherited from the parent folder. Otherwise, returns false.

FolderData.MetaInheritedFrom Property

Gets or sets the parent folder ID from which to inherit metadata information. This is for the FolderData (see page 755) object.

C#

```
public int MetaInheritedFrom;
```

Visual Basic

```
Public Property MetaInheritedFrom() As Integer
```

Returns

The parent folder's integer ID.

FolderData.Name Property

Gets or sets the folder's name for the FolderData (see page 755) object.

C#

```
public String Name;
```

Visual Basic

```
Public Property Name() As String
```

Returns

The folder's name

FolderData.NameWithPath Property

Gets or sets the folder's name with its path. This path is the tree structure in the Workarea, not a folders file system path. For example:

If you want the **Medication Management** folder in the Ektron Medical demo site Workarea, it would appear as `\~YourSite~\Products\Medication Management`

C#

```
public String NameWithPath;
```

Visual Basic

```
Public Property NameWithPath() As String
```

Returns

The folder's name with its path. For example, `\~YourSite~\Products\Medication Management`

FolderData.ParentId Property

Gets or sets the folder's parent folder ID for the FolderData (see page 755) object.

C#

```
public int ParentId;
```

Visual Basic

```
Public Property ParentId() As Integer
```

Returns

The folder's parent folder integer ID.

FolderData.Permissions Property

Gets or sets a list of folder permissions for the logged in user (see page 534). This is for the FolderData (see page 755) object. If you use Ektron.Cms.API (see page 363) to access this information, there is an internal Admin user (see page 534) and all permissions are returned. As an alternative, use Ektron.Cms.ContentAPI.

C#

```
public PermissionData Permissions;
```

Visual Basic

```
Public Property Permissions() As PermissionData
```

Returns

A list of folder permissions for the logged in user (see page 534).

FolderData.PrivateContent Property

Gets or sets whether the content (see page 449) in a folder is private. This is for the FolderData (see page 755) object.

True = the folder's contents private.

False = the folder's contents are public.

C#

```
public Boolean PrivateContent;
```

Visual Basic

```
Public Property PrivateContent() As Boolean
```

Returns

Returns true if the folder's contents are private. Otherwise, returns false.

FolderData.PublishHtmlActive Property

Get or set whether assets in the folder are published as HTML. This is for the FolderData (see page 755) object.

True = assets in the folder are published as HTML.

False = assets are not published as HTML.

C#

```
public Boolean PublishHtmlActive;
```

Visual Basic

```
Public Property PublishHtmlActive() As Boolean
```

Returns

Returns true if assets in the folder are published as HTML. Otherwise, returns false.

Remarks

The PublishPdfEnable property must be set to true.

FolderData.PublishPdfActive Property

Get or set whether assets in the folder are published as a PDF. This is for the FolderData (see page 755) object.

True = assets in the folder are published as a PDF.

False = assets are not published as a PDF.

C#

```
public Boolean PublishPdfActive;
```

Visual Basic

```
Public Property PublishPdfActive() As Boolean
```

Returns

Returns true if assets in the folder are published as a PDF. Otherwise, returns false.

Remarks

The PublishPdfEnable property must be set to true.

FolderData.PublishPdfEnabled Property

Get or set whether the folder allows assets to be published in another format. For example, as PDF or HTML. This is for the FolderData (see page 755) object.

True = assets in the folder can be published in another format.

False = assets are not published in another format.

C#

```
public Boolean PublishPdfEnabled;
```

Visual Basic

```
Public Property PublishPdfEnabled() As Boolean
```

Returns

Returns true if assets in the folder can be published in another format. Otherwise, returns false.

Remarks

If set to true, either the PublishPdfActive (see page 764) property or the PublishHtmlActive (see page 764) property must be set to true.

FolderData.ReplicationMethod Property

Gets or sets whether or not the content (see page 449) in the folder is replicated via quick deploy. The default is 0 (zero). This is for the FolderData (see page 755) object.

- **0** = do not replicate content (see page 449). This is the default.
- **1** = replicate content (see page 449) via the Quick Deploy feature.

C#

```
public int ReplicationMethod;
```

Visual Basic

```
Public Property ReplicationMethod() As Integer
```

Returns

An integer representing whether content (see page 449) in the folder is replicated via quick deploy.

- **0** = do not replicate content (see page 449). This is the default.
- **1** = replicate content (see page 449) via the Quick Deploy feature.

FolderData.SitemapInherited Property

Gets or sets whether or not the Sitemap (see page 301) information for this folder is inherited. This is for the FolderData (

see page 755) object.

- **0** (Zero) = False
- **1** = True

C#

```
public int SitemapInherited;
```

Visual Basic

```
Public Property SitemapInherited() As Integer
```

Returns

Returns an integer representing whether the Sitemap (see page 301) is inherited.

- **0** (Zero) = False
- **1** = True

FolderData.SitemapInheritedFrom Property

Gets or sets the integer ID of the folder from which the Sitemap (see page 301) information is inherited. This is for the FolderData (see page 755) object.

C#

```
public int SitemapInheritedFrom;
```

Visual Basic

```
Public Property SitemapInheritedFrom() As Integer
```

Returns

The integer ID of the folder from which the Sitemap (see page 301) information is inherited.

FolderData.SitemapPath Property

Gets or sets the SitemapPath as an array of Common.SitemapPath. This is for the FolderData (see page 755) object.

C#

```
public Common.SitemapPath() SitemapPath;
```

Visual Basic

```
Public Property SitemapPath() As Common.SitemapPath()
```

Returns

An array of Common.SitemapPath.

FolderData.StyleSheet Property

Gets or sets a style sheet that defines the styles of the content (see page 449) within the content (see page 449) folder. This is for the FolderData (see page 755) object.

C#

```
public String StyleSheet;
```

Visual Basic

```
Public Property StyleSheet() As String
```

Returns

The folder's style sheet.

FolderData.StyleSheetInherited Property

Gets or sets the whether the style sheet is inherited from the parent folder. This is for the FolderData (see page 755) object.

True = the folder inherits style sheet information from the parent folder.

False = the folder does not inherit style sheet information from the parent folder.

C#

```
public Boolean StyleSheetInherited;
```

Visual Basic

```
Public Property StyleSheetInherited() As Boolean
```

Returns

Returns true if the style sheet information is inherited from the parent folder. Otherwise, returns false.

FolderData.TaxonomyInherited Property

Gets or sets whether taxonomy is inherited from the parent folder. This is for the FolderData (see page 755) object.

True = taxonomy is inherited from the parent folder.

False = taxonomy is not inherited from the parent folder.

C#

```
public Boolean TaxonomyInherited;
```

Visual Basic

```
Public Property TaxonomyInherited() As Boolean
```

Returns

Returns true if taxonomy is inherited from the parent folder.

FolderData.TaxonomyInheritedFrom Property

Gets or sets the ID of the parent folder from which the taxonomy is inherited. This is for the FolderData (see page 755) object.

C#

```
public int TaxonomyInheritedFrom;
```

Visual Basic

```
Public Property TaxonomyInheritedFrom() As Integer
```

Returns

The integer ID of the parent folder.

Remarks

The TaxonomyInherited (see page 767) property must be set to True, otherwise it will be ignored.

FolderData.TemplateFileName Property

Gets or sets a template name for the folder. Content (see page 449) in this folder appears on your Web site within the template you choose. This is for the FolderData (see page 755) object.

C#

```
public String TemplateFileName;
```

Visual Basic

```
Public Property TemplateFileName() As String
```

Returns

The template name.

FolderData.TemplateId Property

Gets or sets the folder template's ID for the FolderData (see page 755) object.

C#

```
public int TemplateId;
```

Visual Basic

```
Public Property TemplateId() As Integer
```

Returns

The folder template's integer ID.

FolderData.TemplateInherited Property

Gets or sets the whether the template is inherited from the parent folder. This is for the FolderData (see page 755) object.

True = the folder inherits the template from the parent folder.

False = the folder does not inherit the template from the parent folder.

C#

```
public Boolean TemplateInherited;
```

Visual Basic

```
Public Property TemplateInherited() As Boolean
```

Returns

Returns true if the style sheet information is inherited from the parent folder. Otherwise, returns false.

FolderData.TotalContent Property

Gets or sets the total number of content (see page 449) items in the folder. This is for the FolderData (see page 755) object.

C#

```
public int TotalContent;
```

Visual Basic

```
Public Property TotalContent() As Integer
```

Returns

The total number of content (see page 449) items in the folder.

FolderData.UpdatedContent Property

Gets or sets the total number of content (see page 449) items that have been updated. This is for the FolderData (see page 755) object.

C#

```
public int UpdatedContent;
```

Visual Basic

```
Public Property UpdatedContent() As Integer
```

Returns

The total number of content (see page 449) items that have been updated.

FolderData.XmlConfiguration Property

Gets or sets the XML configuration for a folder. This is for the FolderData (see page 755) object.

C#

```
public XmlConfigData() XmlConfiguration;
```

Visual Basic

```
Public Property XmlConfiguration() As XmlConfigData()
```

Returns

An array of XmlConfigData.

FolderData.XmlInherited Property

Gets or sets whether the folder inherits the XML configuration from the parent folder. This is for the FolderData (see page 755) object.

True = the folder inherits the XML configuration from the parent folder.

False = the folder does not inherit the XML configuration from the parent folder.

C#

```
public Boolean XmlInherited;
```

Visual Basic

```
Public Property XmlInherited() As Boolean
```

Returns

Returns true if the XML is inherited from the parent folder. Otherwise, returns false.

FolderData.XmlInheritedFrom Property

Gets or sets the parent folder ID from which the XML is inherited. This is for the FolderData (see page 755) object.

C#

```
public int XmlInheritedFrom;
```

Visual Basic

```
Public Property XmlInheritedFrom() As Integer
```

Returns

The parent folder's integer ID.

FolderRequest Class

Class Hierarchy

FolderRequest

C#

```
public class FolderRequest;
```

Visual Basic

```
Public Class FolderRequest
```

File

FolderData.vb

Description

This is class FolderRequest.

FolderRequest Properties

The properties of the FolderRequest class are listed here.

Public Properties

	Name	Description
	BreakInheritButton (see page 772)	Gets or sets whether to break the folder's permission inheritance for the FolderRequest (see page 770) data object. **True** = break a folder's permission inheritance. **False** = Do not break a folder's permission inheritance.
	CategoryRequired (see page 772)	Flag to indicate if a category selection is required while adding/editing content (see page 449). This is for the FolderRequest (see page 770) data object.
	ContentSubAssignments (see page 773)	Gets or sets the folder's content (see page 449) subscription assignments. This is for the FolderRequest (see page 770) data object.

	DomainProduction (see page 773)	Gets or sets the production domain for the folder when the folder is a domain folder and the site is a production site. This is for the FolderRequest (see page 770) data object.
	DomainStaging (see page 773)	Gets or sets the staging domain for the folder when the folder is a domain folder and the site is a staging site. This is for the FolderRequest (see page 770) data object.
	EnableReplication (see page 773)	Gets or sets whether the folder is replicated during Quick Deploy. This is for the FolderRequest (see page 770) data object. **True** = replicate the folder during Quick Deploy. **False** = Do not replicate the folder during Quick Deploy.
	FileDirectory (see page 774)	Gets or sets the physical path to the folder's file directory for the FolderRequest (see page 770) data object.
	FolderCfldAssignments (see page 774)	Gets or set a list of metadata assigned to the folder. This is for the FolderRequest (see page 770) data object.
	FolderDescription (see page 774)	Gets or sets the folder description for the FolderRequest (see page 770) data object.
	FolderId (see page 775)	Gets or sets the FolderId for the FolderRequest (see page 770) data object.
	FolderName (see page 775)	Gets or sets the FolderName for the FolderRequest (see page 770) data object.
	FolderType (see page 775)	Gets or sets the folder type of the folder. This enumeration is available at Ektron.Cms.Common.EkEnumeration.FolderType. Available Types: • **0** = Content (see page 449) - a content (see page 449) folder that contains assets, documents and HTML content (see page 449). • **1** = Blog (see page 454) - designates the folder for use with the Blogging feature. • **2** = Domain - designates the folder as a domain folder when using the Multi-site feature. • **3** = Discussion Board - designates the folder for use as a Threaded Discussion folder. • **4** = Discussion Forum (see page 167) - a subfolder of the Threaded Discussion folder. • **5** = Root - Specifies a folder as root folder. This option... more (see page 775)
	ImageDirectory (see page 776)	Gets or sets the physical path to the folder's image directory for the FolderRequest (see page 770) data object.
	IsDomainFolder (see page 776)	Gets or sets whether the folder is a domain folder for the FolderRequest (see page 770) data object. **True** = designate a folder as a domain folder. **False** = folder is not a domain folder.
	MetaInherited (see page 776)	Gets or sets whether the folder inherits metadata from another folder. The folder ID must be specified in MetaInheritedFrom (see page 777) property. • **1** = inherits metadata from a folder. • **0** = does not inherit metadata from a folder.
	MetaInheritedFrom (see page 777)	Gets or Sets the folder ID from which metadata is inherited. This is for the FolderRequest (see page 770) data object.
	ParentId (see page 777)	Gets or sets the folder's parent folder ID for the FolderRequest (see page 770) data object.
	PublishActive (see page 777)	Gets or sets whether assets in the folder are published in another format. It also determines in which format the assets are published. For example, HTML or PDF. This is for the FolderRequest (see page 770) data object. • **PublishPdfActive** = publish asset as a PDF. • **PublishHtmlActive** = publish asset as HTML. • **""** (Empty) - assets in the folder are not published.
	SiteMapPath (see page 777)	Gets or sets a folder's SiteMap (see page 430) path for the FolderRequest (see page 770) data object.
	SiteMapPathInherit (see page 778)	Gets or sets whether the folder's SiteMap (see page 430) path is inherited for the FolderRequest (see page 770) data object. **True** = SiteMap (see page 430) path is inherited. **False** = SiteMap (see page 430) path is not inherited.
	StyleSheet (see page 778)	Gets or sets the folder's style sheet for the FolderRequest (see page 770) data object.
	SubscriptionProperties (see page 778)	Gets or sets the folder's Subscriptions properties for the FolderRequest (see page 770) data object.

	suppressNotification (see page 779)	Gets or sets whether to disable Web alerts. This is for the FolderRequest (see page 770) data object. **True** = suppress Web Alert notifications. **False** = allow Web Alert notifications.
	TaxonomyIdList (see page 779)	Gets or sets a list of taxonomy IDs values separated by comma. This is for the FolderRequest (see page 770) data object.
	TaxonomyInherited (see page 779)	Set the property to true to inherit taxonomy from a folder. The folder ID must be specified in TaxonomyInheritedFrom (see page 780) property. This is for the FolderRequest (see page 770) data object. **1** = this folder inherits taxonomy from a folder. **0** = this folder does not inherit taxonomy from a folder.
	TaxonomyInheritedFrom (see page 780)	Gets or sets the numeric folder ID from which taxonomy is inherited. This is for the FolderRequest (see page 770) data object.
	TemplateFileName (see page 780)	Gets or sets the folder's template file for the FolderRequest (see page 770) data object.
	XmlConfiguration (see page 780)	Gets or sets the XML configuration for the folder. This is for the FolderRequest (see page 770) data object.
	XmlInherited (see page 781)	Gets or sets whether the XML configuration for the folder is inherited from the parent folder. This is for the FolderRequest (see page 770) data object. **True** = inherit the XML configuration from the parent folder. **False** = Do not inherit the XML configuration from the parent folder.

Legend

	Property

FolderRequest.BreakInheritButton Property

Gets or sets whether to break the folder's permission inheritance for the FolderRequest (see page 770) data object.

True = break a folder's permission inheritance.

False = Do not break a folder's permission inheritance.

C#

```
public Boolean BreakInheritButton;
```

Visual Basic

```
Public Property BreakInheritButton() As Boolean
```

Returns

Returns True if the folder's permission inheritance is broken.

FolderRequest.CategoryRequired Property

Flag to indicate if a category selection is required while adding/editing content (see page 449). This is for the FolderRequest (see page 770) data object.

C#

```
public Boolean CategoryRequired;
```

Visual Basic

```
Public Property CategoryRequired() As Boolean
```

Returns

Boolean value. If true, a category selection is required. Otherwise, false

FolderRequest.ContentSubAssignments Property

Gets or sets the folder's content (see page 449) subscription assignments. This is for the FolderRequest (see page 770) data object.

C#

```
public String ContentSubAssignments;
```

Visual Basic

```
Public Property ContentSubAssignments() As String
```

FolderRequest.DomainProduction Property

Gets or sets the production domain for the folder when the folder is a domain folder and the site is a production site. This is for the FolderRequest (see page 770) data object.

C#

```
public String DomainProduction;
```

Visual Basic

```
Public Property DomainProduction() As String
```

Returns

The folder's existing production site domain.

Remarks

Example: www.ektron.com

FolderRequest.DomainStaging Property

Gets or sets the staging domain for the folder when the folder is a domain folder and the site is a staging site. This is for the FolderRequest (see page 770) data object.

C#

```
public String DomainStaging;
```

Visual Basic

```
Public Property DomainStaging() As String
```

Returns

The folder's existing staging site domain.

Remarks

Example: dev.ektron.com

FolderRequest.EnableReplication Property

Gets or sets whether the folder is replicated during Quick Deploy. This is for the FolderRequest (see page 770) data object.

True = replicate the folder during Quick Deploy.

False = Do not replicate the folder during Quick Deploy.

C#

```
public Boolean EnableReplication;
```

Visual Basic

```
Public Property EnableReplication() As Boolean
```

Returns

Returns True if the folder is replicated during Quick Deploy.

Remarks

This property is used with the Quick Deploy feature in CMS400.NET. It does not affect Site (see page 424) Replication.

FolderRequest.FileDirectory Property

Gets or sets the physical path to the folder's file directory for the FolderRequest (see page 770) data object.

C#

```
public String FileDirectory;
```

Visual Basic

```
Public Property FileDirectory() As String
```

Returns

An existing folder's file directory.

FolderRequest.FolderCfldAssignments Property

Gets or set a list of metadata assigned to the folder. This is for the FolderRequest (see page 770) data object.

C#

```
public String FolderCfldAssignments;
```

Visual Basic

```
Public Property FolderCfldAssignments() As String
```

FolderRequest.FolderDescription Property

Gets or sets the folder description for the FolderRequest (see page 770) data object.

C#

```
public String FolderDescription;
```

Visual Basic

```
Public Property FolderDescription() As String
```

Returns

An existing folder's description.

FolderRequest.FolderId Property

Gets or sets the FolderId for the FolderRequest (see page 770) data object.

C#

```
public int FolderId;
```

Visual Basic

```
Public Property FolderId() As Integer
```

Returns

An existing folder's integer ID.

FolderRequest.FolderName Property

Gets or sets the FolderName for the FolderRequest (see page 770) data object.

C#

```
public String FolderName;
```

Visual Basic

```
Public Property FolderName() As String
```

Returns

An existing folder's name.

FolderRequest.FolderType Property

Gets or sets the folder type of the folder. This enumeration is available at Ektron.Cms.Common.EkEnumeration.FolderType.

Available Types:

- **0** = Content (see page 449) - a content (see page 449) folder that contains assets, documents and HTML content (see page 449).
- **1** = Blog (see page 454) - designates the folder for use with the Blogging feature.
- **2** = Domain - designates the folder as a domain folder when using the Multi-site feature.
- **3** = Discussion Board - designates the folder for use as a Threaded Discussion folder.
- **4** = Discussion Forum (see page 167) - a subfolder of the Threaded Discussion folder.
- **5** = Root - Specifies a folder as root folder. This option is not available.
- **6** = Community - Membership (see page 240) users can add and edit content (see page 449) contained in these folders.

C#

```
public int FolderType;
```

Visual Basic

```
Public Property FolderType() As Integer
```

Returns

The integer value of the folder type.

FolderRequest.ImageDirectory Property

Gets or sets the physical path to the folder's image directory for the FolderRequest (see page 770) data object.

C#

```
public String ImageDirectory;
```

Visual Basic

```
Public Property ImageDirectory() As String
```

Returns

An existing folder's image directory.

FolderRequest.IsDomainFolder Property

Gets or sets whether the folder is a domain folder for the FolderRequest (see page 770) data object.

True = designate a folder as a domain folder.

False = folder is not a domain folder.

C#

```
public Boolean IsDomainFolder;
```

Visual Basic

```
Public Property IsDomainFolder() As Boolean
```

Returns

Returns True if the folder is a domain folder.

FolderRequest.MetaInherited Property

Gets or sets whether the folder inherits metadata from another folder. The folder ID must be specified in MetaInheritedFrom (see page 777) property.

- **1** = inherits metadata from a folder.
- **0** = does not inherit metadata from a folder.

C#

```
public int MetaInherited;
```

Visual Basic

```
Public Property MetaInherited() As Integer
```

Returns

An integer value representing whether the folder inherits metadata from another folder.

Remarks

If set to true, the folder ID must be specified in MetaInheritedFrom (see page 777) property.

FolderRequest.MetaInheritedFrom Property

Gets or Sets the folder ID from which metadata is inherited. This is for the FolderRequest (see page 770) data object.

C#

```
public int MetaInheritedFrom;
```

Visual Basic

```
Public Property MetaInheritedFrom() As Integer
```

Returns

An integer value representing the folder ID

Remarks

InheritMetadata must be set to 1, otherwise it will be ignored.

FolderRequest.ParentId Property

Gets or sets the folder's parent folder ID for the FolderRequest (see page 770) data object.

C#

```
public int ParentId;
```

Visual Basic

```
Public Property ParentId() As Integer
```

Returns

An existing folder's parent folder ID.

FolderRequest.PublishActive Property

Gets or sets whether assets in the folder are published in another format. It also determines in which format the assets are published. For example, HTML or PDF. This is for the FolderRequest (see page 770) data object.

- **PublishPdfActive** = publish asset as a PDF.
- **PublishHtmlActive** = publish asset as HTML.
- **""** (Empty) - assets in the folder are not published.

C#

```
public String PublishActive;
```

Visual Basic

```
Public Property PublishActive() As String
```

Returns

A string based on whether to publish contents as HTML, PDF or not at all.

FolderRequest.SiteMapPath Property

Gets or sets a folder's SiteMap (see page 430) path for the FolderRequest (see page 770) data object.

C#

```
public Ektron.Cms.Common.SitemapPath() SiteMapPath;
```

Visual Basic

```
Public Property SiteMapPath() As Ektron.Cms.Common.SitemapPath()
```

Returns

An existing folder's SiteMap (see page 430) path.

FolderRequest.SiteMapPathInherit Property

Gets or sets whether the folder's SiteMap (see page 430) path is inherited for the FolderRequest (see page 770) data object.

True = SiteMap (see page 430) path is inherited.

False = SiteMap (see page 430) path is not inherited.

C#

```
public Boolean SiteMapPathInherit;
```

Visual Basic

```
Public Property SiteMapPathInherit() As Boolean
```

Returns

True or False depending whether an existing folder's SiteMap (see page 430) path is inherited or not.

FolderRequest.StyleSheet Property

Gets or sets the folder's style sheet for the FolderRequest (see page 770) data object.

C#

```
public String StyleSheet;
```

Visual Basic

```
Public Property StyleSheet() As String
```

Returns

An existing folder's style sheet name.

FolderRequest.SubscriptionProperties Property

Gets or sets the folder's Subscriptions properties for the FolderRequest (see page 770) data object.

C#

```
public SubscriptionPropertiesData SubscriptionProperties;
```

Visual Basic

```
Public Property SubscriptionProperties() As SubscriptionPropertiesData
```

FolderRequest.suppressNotification Property

Gets or sets whether to disable Web alerts. This is for the FolderRequest (see page 770) data object.

True = suppress Web Alert notifications.

False = allow Web Alert notifications.

C#

```
public Boolean suppressNotification;
```

Visual Basic

```
Public Property suppressNotification() As Boolean
```

Description

True = suppress Web Alert notifications.

False = allow Web Alert notifications.

Default = False, to override this use the BreakInheritance property of the FolderRequest.SubscriptionProperties (see page 778) object (set it to True).

Remarks

Boolean value. If true, suppress Web Alert notifications. Otherwise, false.

FolderRequest.TaxonomyIdList Property

Gets or sets a list of taxonomy IDs values separated by comma. This is for the FolderRequest (see page 770) data object.

C#

```
public String TaxonomyIdList;
```

Visual Basic

```
Public Property TaxonomyIdList() As String
```

Returns

Comma separated list of taxonomy IDs.

FolderRequest.TaxonomyInherited Property

Set the property to true to inherit taxonomy from a folder. The folder ID must be specified in TaxonomyInheritedFrom (see page 780) property. This is for the FolderRequest (see page 770) data object.

1 = this folder inherits taxonomy from a folder.

0 = this folder does not inherit taxonomy from a folder.

C#

```
public Boolean TaxonomyInherited;
```

Visual Basic

```
Public Property TaxonomyInherited() As Boolean
```

Returns

An integer value representing whether the folder inherits taxonomy from a folder.

Remarks

If set to true, the folder ID must be specified in TaxonomyInheritedFrom (see page 780) property.

FolderRequest.TaxonomyInheritedFrom Property

Gets or sets the numeric folder ID from which taxonomy is inherited. This is for the FolderRequest (see page 770) data object.

C#

```
public int TaxonomyInheritedFrom;
```

Visual Basic

```
Public Property TaxonomyInheritedFrom() As Integer
```

Returns

An integer representing the folder ID from which taxonomy is inherited.

Remarks

The TaxonomyInherited (see page 779) property must be set to 1 or True, otherwise it will be ignored.

FolderRequest.TemplateFileName Property

Gets or sets the folder's template file for the FolderRequest (see page 770) data object.

C#

```
public String TemplateFileName;
```

Visual Basic

```
Public Property TemplateFileName() As String
```

Returns

An existing folder's template name.

FolderRequest.XmlConfiguration Property

Gets or sets the XML configuration for the folder. This is for the FolderRequest (see page 770) data object.

C#

```
public String XmlConfiguration;
```

Visual Basic

```
Public Property XmlConfiguration() As String
```

Returns

The existing XML configuration for the folder.

FolderRequest.XmlInherited Property

Gets or sets whether the XML configuration for the folder is inherited from the parent folder. This is for the FolderRequest (see page 770) data object.

True = inherit the XML configuration from the parent folder.

False = Do not inherit the XML configuration from the parent folder.

C#

```
public Boolean XmlInherited;
```

Visual Basic

```
Public Property XmlInherited() As Boolean
```

Returns

Returns True if the XML configuration is inherited from the parent folder.

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 782)	Specifies whether the form data should be automatically filled if the user (see page 534) is known and has previously entered data for this form.
	Description (see page 782)	Gets or sets a description of the form. The description must be 255 characters or less.
	MailCc (see page 783)	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 783)	Gets or sets the email address assigned as the sender when email is sent upon submittal of form data.
	MailPreamble (see page 783)	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 783)	Gets or sets the subject of the email sent when form data is submitted. Maximum length is 128 characters.
	MailTo (see page 784)	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 784)	Gets or sets the revision number of the poll as a string.
	SendXmlPacket (see page 784)	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.
	StoreDataTo (see page 785)	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.
	Validation (see page 785)	Gets or sets the form's validation information as an array of FormValidationData.

Legend

	Property

FormData.Autofill Property

Specifies whether the form data should be automatically filled if the user (see page 534) 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 785) 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 786)	Gets or sets the value of the form field.
	FieldName (see page 786)	Gets or sets the name of the form field.

Legend

	Property

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

FormFieldDatalist

C#

```
public class FormFieldDatalist;
```

Visual Basic

```
Public Class FormFieldDatalist
```

File

FormFieldDatalist.vb

Remarks

Typically used as an array in class FormFieldList (see page 794).

Typically referenced by name in class FormFieldDefinition (see page 791).

FormFieldDatalist Properties

The properties of the FormFieldDatalist class are listed here.

Public Properties

	Name	Description
	CaptionXPath (see page 787)	Gets or sets the CaptionXPath of the pre-defined datalist.
	DatalistItems (see page 788)	Gets or sets the array of items defined in the datalist.
	DataNamespaces (see page 788)	Gets or sets the Data Namespace of the pre-defined datalist.
	DataSelect (see page 788)	Gets or sets the Data Select of the pre-defined datalist.
	DataSrc (see page 788)	Gets or sets the Data Source of the pre-defined datalist.
	Language (see page 788)	Gets or sets the language code of the datalist. For example, en or en-US.
	Name (see page 789)	Gets or sets the name of the datalist.
	ValueXPath (see page 789)	Gets or sets the Value XPath of the pre-defined datalist.

Legend

	Property

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

FormFieldDatalistItem

C#

```
public class FormFieldDatalistItem;
```

Visual Basic

```
Public Class FormFieldDatalistItem
```

File

FormFieldDatalistItem.vb

Remarks

Typically used as an array in class FormFieldDatalist (see page 786) Cms.FormFieldDatalist

FormFieldDatalistItem Properties

The properties of the FormFieldDatalistItem class are listed here.

Public Properties

	Name	Description
	DataValue (see page 790)	Gets or sets the value for this item in the data list.
	DisplayValue (see page 790)	Gets or sets the value to be displayed.
	IsDefault (see page 791)	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 791)	Gets or sets whether this item is enabled for selection. Set to **True** if this item is enabled.

Legend

	Property

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 794).

FormFieldDefinition Properties

The properties of the FormFieldDefinition class are listed here.

Public Properties

	Name	Description
	Basetype (see page 792)	Gets or sets the basic datatype of the field.
	ContentType (see page 792)	Gets or sets the type of content (see page 449) allowed within the field. This is loosely associated with the datatype.
	DatalistName (see page 793)	Gets or sets the name of the datalist, if one is defined for this field.
	Datatype (see page 793)	Gets or sets the datatype of the field.
	DisplayName (see page 793)	Gets or sets the name of the field to be displayed.
	FieldName (see page 794)	Gets or sets the name of the field.
	Language (see page 794)	Gets or sets the language code of the datalist. For example, en or en-US.
	XPath (see page 794)	Gets or sets the XPath expression that locates the field.

Legend

	Property

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 (see page 447) and checkbox.

FormFieldDefinition.ContentType Property

Gets or sets the type of content (see page 449) 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 (see page 449) 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 795)	Gets or sets an array of defined datalists.
	Fields (see page 795)	Gets or sets an array of field definitions.

Legend

	Property

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 786)

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 791)

FormRequestData Class

A data object that contains information about a form.

Class Hierarchy

FormRequestData

C#

```
public class FormRequestData;
```

Visual Basic

```
Public Class FormRequestData
```

File

FormRequestData.vb

FormRequestData Properties

The properties of the FormRequestData class are listed here.

Public Properties

	Name	Description
	AutoFill (see page 797)	Indicates whether a form's fields are pre-filled when the form is displayed. • **True** - a form's fields are pre-filled when the form is displayed. • **False** - a form's fields are empty when the form is displayed.
	FieldList (see page 797)	Gets or sets a list of the fields defined in the form.
	Id (see page 797)	The integer ID of the form.
	IsPoll (see page 797)	Gets whether the form is used as poll. • **True** - this form is a poll. • **False** - this form is not a poll.
	Language (see page 798)	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 798)	Gets or sets the data submitted from the form as a collection.
	PreFillFielddata (see page 798)	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 (see page 534) 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 799)	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.

Legend

	Property

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 794)

FormRequestData.Id Property

The integer ID of the form.

C#

```
public int Id;
```

Visual Basic

```
Public Property Id() As Integer
```

Returns

The ID of the form as an integer.

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 (see page 534) 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 797) 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

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 800)	Gets or sets the array of the data associated with each field defined in the form.
	DateSubmitted (see page 800)	Gets or sets the date the data in the DataItems (see page 800)() array was submitted.

	FormDataID (see page 800)	Gets or sets the integer ID for the form data.
	UserId (see page 800)	Gets or sets the user (see page 534) id of the person who submitted the data. A value of 0 means anonymous.
	UserInfo (see page 801)	Gets or sets the user (see page 534) information as an Ektron.Cms.UserData (see page 801) object.

Legend

	Property

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 800)() array was submitted.

C#

```
public Date DateSubmitted;
```

Visual Basic

```
Public Property DateSubmitted() As Date
```

FormSubmittedData.FormDataID Property

Gets or sets the integer ID for the form data.

C#

```
public int FormDataID;
```

Visual Basic

```
Public Property FormDataID() As Integer
```

Remarks

A value of -1 means the ID is undefined.

FormSubmittedData.UserId Property

Gets or sets the user (see page 534) id of the person who submitted the data. A value of 0 means anonymous.

C#

```
public int UserId;
```

Visual Basic

```
Public Property UserId() As Integer
```

Remarks

UserId and UserInfo (see page 801) are not kept synchronized in this class.

FormSubmittedData.UserInfo Property

Gets or sets the user (see page 534) information as an Ektron.Cms.UserData (see page 801) object.

C#

```
public UserData UserInfo;
```

Visual Basic

```
Public Property UserInfo() As UserData
```

Remarks

UserId (see page 800) and UserInfo are not kept synchronized in this class.

UserData Class

A data object that contains the properties that make up a user (see page 534).

Class Hierarchy

C#

```
public class UserData : CMSObject;
```

Visual Basic

```
Public Class UserData
Inherits CMSObject
```

File

UserData.vb

UserData Properties

The properties of the UserData class are listed here.

Public Properties

	Name	Description
	AcceptedTerms (see page 802)	Gets or set whether the user (see page 534) has accepted the Terms and Conditions associated with a Threaded Discussion. This is for the UserData (see page 801) object.
	AdUserName (see page 802)	Gets or sets the Active Directory (see page 118) user (see page 534) name for the UserData (see page 801) object.
	Avatar (see page 803)	Gets or sets the user (see page 534)'s avatar URL for a Threaded Discussion. This is for the UserData (see page 801) object. For Example, `www.example.com/images/smiley.gif`
	DateCreated (see page 803)	Gets or sets the creation date of a user (see page 534). This is for the UserData (see page 801) object.
	DisplayName (see page 803)	Gets or sets the user (see page 534)'s display alias for the UserData (see page 801) object, used in the forums and other locations.
	DisplayUserName (see page 803)	Gets or sets the user (see page 534)'s first name and last name for the UserData (see page 801) object. This is for the UserData (see page 801) object.
	Email (see page 804)	Gets or sets the user (see page 534)'s email address for the UserData (see page 801) object.

	FirstName (see page 804)	Gets or sets the first name of a user (see page 534) for the UserData (see page 801) object.
	GroupId (see page 804)	Gets or sets the integer ID of the group to which the user (see page 534) belongs. This is for the UserData (see page 801) object.
	GroupName (see page 805)	Gets or sets the name of the group to which the user (see page 534) belongs. This is for the UserData (see page 801) object.
	Id (see page 805)	Gets or sets the integer ID of a user (see page 534) for the UserData (see page 801) object.
	IsMemberShip (see page 805)	Gets or sets a boolean indicating whether the user (see page 534) is a Membership (see page 240) user (see page 534) or not. This is for UserData (see page 801) object.
	LanguageId (see page 805)	Gets or sets the integer ID of the language for the UserData (see page 801) object.
	LanguageName (see page 806)	Gets or set the name of the user (see page 534)'s language for the UserData (see page 801) object.
	LastLoginDate (see page 806)	Gets or sets the last date a user (see page 534) logged into CMS400.NET. This is for the UserData (see page 801) object.
	LastName (see page 806)	Gets or sets the last name of a user (see page 534) for the UserData (see page 801) object.
	LoginAttempts (see page 807)	Gets an Int16 representing the number of times a user (see page 534) has attempted to login for the UserData (see page 801) object. This number resets to 0 (zero) once a user (see page 534) successfully logs in.
	Password (see page 807)	Gets or sets the user (see page 534)'s password for the UserData (see page 801) object.
	Rank (see page 807)	Gets or sets the user (see page 534)'s rank for a Threaded Discussion for the UserData (see page 801) object.
	Signature (see page 807)	Gets or sets the user (see page 534)'s signature for the UserData (see page 801) object, used in the forums and other locations.
	Username (see page 808)	Gets or sets the username of a user (see page 534) for the UserData (see page 801) object.
	UserPreference (see page 808)	Gets or sets a user (see page 534)'s preferences as UserPreferenceData for the UserData (see page 801) object.

Legend

	Property

UserData.AcceptedTerms Property

Gets or set whether the user (see page 534) has accepted the Terms and Conditions associated with a Threaded Discussion. This is for the UserData (see page 801) object.

C#

```
public Boolean AcceptedTerms;
```

Visual Basic

```
Public Property AcceptedTerms() As Boolean
```

Returns

A boolean indicating whether the user (see page 534) has accepted the Terms and Conditions associated with a Threaded Discussion.

- **True** = user (see page 534) has accepted Terms and Conditions.
- **False** = user (see page 534) has not accepted Terms and Conditions.

Description

UserData.AdUserName Property

Gets or sets the Active Directory (see page 118) user (see page 534) name for the UserData (see page 801) object.

C#

```
public String AdUserName;
```

Visual Basic

```
Public Property AdUserName() As String
```

Returns

The Active Directory (see page 118) user (see page 534) name.

UserData.Avatar Property

Gets or sets the user (see page 534)'s avatar URL for a Threaded Discussion. This is for the UserData (see page 801) object. For Example, `www.example.com/images/smiley.gif`

C#

```
public String Avatar;
```

Visual Basic

```
Public Property Avatar() As String
```

Returns

The user (see page 534)'s avatar URL for a Threaded Discussion.

UserData.DateCreated Property

Gets or sets the creation date of a user (see page 534). This is for the UserData (see page 801) object.

C#

```
public String DateCreated;
```

Visual Basic

```
Public Property DateCreated() As String
```

Returns

The date the user (see page 534) was created.

UserData.DisplayName Property

Gets or sets the user (see page 534)'s display alias for the UserData (see page 801) object, used in the forums and other locations.

C#

```
public String DisplayName;
```

Visual Basic

```
Public Property DisplayName() As String
```

Returns

User (see page 534)'s display alias, used in the forums and other locations.

UserData.DisplayUserName Property

Gets or sets the user (see page 534)'s first name and last name for the UserData (see page 801) object. This is for the

UserData (see page 801) object.

C#

```
public String DisplayUserName;
```

Visual Basic

```
Public Property DisplayUserName() As String
```

Returns

User (see page 534)'s first name and last name.

UserData.Email Property

Gets or sets the user (see page 534)'s email address for the UserData (see page 801) object.

C#

```
public String Email;
```

Visual Basic

```
Public Property Email() As String
```

Returns

The user (see page 534)'s email address.

UserData.FirstName Property

Gets or sets the first name of a user (see page 534) for the UserData (see page 801) object.

C#

```
public String FirstName;
```

Visual Basic

```
Public Property FirstName() As String
```

Returns

The first name of a user (see page 534).

UserData.GroupId Property

Gets or sets the integer ID of the group to which the user (see page 534) belongs. This is for the UserData (see page 801) object.

C#

```
public int GroupId;
```

Visual Basic

```
Public Property GroupId() As Integer
```

Returns

The group ID to which the user (see page 534) belongs.

UserData.GroupName Property

Gets or sets the name of the group to which the user (see page 534) belongs. This is for the UserData (see page 801) object.

C#

```
public String GroupName;
```

Visual Basic

```
Public Property GroupName() As String
```

Returns

The name of the group to which the user (see page 534) belongs.

UserData.Id Property

Gets or sets the integer ID of a user (see page 534) for the UserData (see page 801) object.

C#

```
public int Id;
```

Visual Basic

```
Public Property Id() As Integer
```

Returns

An integer representing the ID of a user (see page 534).

UserData.IsMemberShip Property

Gets or sets a boolean indicating whether the user (see page 534) is a Membership (see page 240) user (see page 534) or not. This is for UserData (see page 801) object.

C#

```
public Boolean IsMemberShip;
```

Visual Basic

```
Public Property IsMemberShip() As Boolean
```

Returns

A boolean indicating whether the user (see page 534) is a Membership (see page 240) user (see page 534).

- **True** = user (see page 534) is a Membership (see page 240) user (see page 534).
- **False** = user (see page 534) is a CMS400.NET user (see page 534).

Description

UserData.LanguageId Property

Gets or sets the integer ID of the language for the UserData (see page 801) 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 (see page 534)'s language for the UserData (see page 801) object.

C#

```
public String LanguageName;
```

Visual Basic

```
Public Property LanguageName() As String
```

Returns

Name of the user (see page 534)'s language.

UserData.LastLoginDate Property

Gets or sets the last date a user (see page 534) logged into CMS400.NET. This is for the UserData (see page 801) object.

C#

```
public String LastLoginDate;
```

Visual Basic

```
Public Property LastLoginDate() As String
```

Returns

The last login date for a user (see page 534).

UserData.LastName Property

Gets or sets the last name of a user (see page 534) for the UserData (see page 801) object.

C#

```
public String LastName;
```

Visual Basic

```
Public Property LastName() As String
```

Returns

The last name of a user (see page 534).

UserData.LoginAttempts Property

Gets an Int16 representing the number of times a user (see page 534) has attempted to login for the UserData (see page 801) object. This number resets to 0 (zero) once a user (see page 534) successfully logs in.

C#

```
public Int16 LoginAttempts;
```

Visual Basic

```
Public Property LoginAttempts() As Int16
```

Returns

Int16 representing the number of times a user (see page 534) has attempted to login. This number resets to 0 (zero) once a user (see page 534) successfully logs in.

UserData.Password Property

Gets or sets the user (see page 534)'s password for the UserData (see page 801) object.

C#

```
public String Password;
```

Visual Basic

```
Public Property Password() As String
```

Returns

The user (see page 534)'s password.

UserData.Rank Property

Gets or sets the user (see page 534)'s rank for a Threaded Discussion for the UserData (see page 801) object.

C#

```
public UserRank Rank;
```

Visual Basic

```
Public Property Rank() As UserRank
```

Returns

The user (see page 534)'s rank for a Threaded Discussion.

UserData.Signature Property

Gets or sets the user (see page 534)'s signature for the UserData (see page 801) object, used in the forums and other locations.

C#

```
public String Signature;
```

Visual Basic

```
Public Property Signature() As String
```

Returns

The user (see page 534)'s signature used in the forums and other locations.

UserData.Username Property

Gets or sets the username of a user (see page 534) for the UserData (see page 801) object.

C#

```
public String Username;
```

Visual Basic

```
Public Property Username() As String
```

Returns

The username of a user (see page 534).

UserData.UserPreference Property

Gets or sets a user (see page 534)'s preferences as UserPreferenceData for the UserData (see page 801) object.

C#

```
public UserPreferenceData UserPreference;
```

Visual Basic

```
Public Property UserPreference() As UserPreferenceData
```

Returns

A user (see page 534)'s preferences as UserPreferenceData.

Index

.

.NET Assemblies 362

A

ActiveDirectoryUser class 593
- about ActiveDirectoryUser class 593
- ActiveDirectoryUser methods 594
- AddADGroupToCMS 594
- DeSynchUserGroups 595
- DeSynchUsers 595
- GetADMapping 596
- GetADStatus 596
- GetAvailableADGroups 596
- GetAvailableADUsers 597
- GetCMSGroupsToSync 597
- GetCMSRelationshipsToSync 598
- GetCMSUsersToSync 598
- GetDomains 599
- MapCMSUserGroupToAD 599
- MapCMSUserToAD 600
- UpdateADMapping 600
- UpdateADUsersGroups 601
- UpdateUsersInfoFromAD 601

ActiveTopics class 14
- about ActiveTopics class 14
- ActiveTopics methods 14
- ActiveTopics properties 15
- BoardID 15
- CacheInterval 16
- DisplayMode 16
- Fill 14
- MaxNumber 16
- ToString 14
- URLPath 16
- Wrap 15

ADUser class 534
- about ADUser class 534
- AddADGroupToCMS 535
- AddADUsersToCMSByUsername 536
- ADUser methods 535
- DeSynchUserGroups 536
- DeSynchUsers 536
- GetADMapping 536
- GetADStatus 537
- GetAvailableADGroups 537
- GetAvailableADUsers 537
- GetCMSGroupsToSync 538
- GetCMSRelationshipsToSync 538
- GetCMSUsersToSync 538
- GetDomains 539
- MapCMSUserGroupToAD 539
- MapCMSUserToAD 539
- SynchCMSGroupsToAD 540
- SynchCMSRelationshipsToAD 540
- SynchCMSUsersToAD 540
- UpdateADMapping 541
- UpdateADUsersGroups 541
- UpdateUsersInfoFromAD 541

Analytics class 443
- about Analytics class 443
- Analytics methods 443
- Analytics properties 446
- AnalyticsEnabled 446
- AppendToIISLog 446
- GetSessionRank 444
- HardDumpAnalyticsBuffer 444
- InMemoryContentHitsData 447
- QueryAnalytics 444
- RecordsBeforeWrite 447
- SoftDumpAnalyticsBuffer 444
- TimeBeforeWrite 447
- TrackVisit 445
- TrackVisit method 445

AnalyticsTracker class 17
- about AnalyticsTracker class 17

AnalyticsTracker methods 17
AnalyticsTracker properties 18
DefaultContentID 18
DynamicParameter 18
EnableAnalytics 19
Wrap 18
API 363
API namespace 363
Classes 364
API.Analytics 442
API.Analytics namespace 442
Classes 443
API.Analytics.Analytics 443
API.Analytics.Analytics.AnalyticsEnabled 446
API.Analytics.Analytics.AppendToIISLog 446
API.Analytics.Analytics.GetSessionRank 444
API.Analytics.Analytics.HardDumpAnalyticsBuffer 444
API.Analytics.Analytics.InMemoryContentHitsData 447
API.Analytics.Analytics.QueryAnalytics 444
API.Analytics.Analytics.RecordsBeforeWrite 447
API.Analytics.Analytics.SoftDumpAnalyticsBuffer 444
API.Analytics.Analytics.TimeBeforeWrite 447
API.Analytics.Analytics.TrackVisit 445
API.BusinessRule 365
API.BusinessRule.GetBusinessRules 366
API.Calendar 447
API.Calendar namespace 447
Classes 448
API.Calendar.Calendar 448
API.Calendar.CalendarEvent 448
API.Calendar.CalendarEventType 449
API.Content 449
API.Content namespace 449
Classes 449
API.Content.Asset 450
API.Content.Asset.GetAssetMgtConfigInfo 450
API.Content.Asset.GetAssetMgtInfo 451
API.Content.Asset.GetAssetSupertypes 451
API.Content.Asset.GetCreateDragDropSnippet 451
API.Content.Asset.GetCreateEktronExplorerSnippet 452
API.Content.Asset.GetCreateSnippet 452
API.Content.Asset.GetDragDropWOSnippet 452
API.Content.Asset.GetEditSnippet 453
API.Content.Asset.GetMyAssetMgtID 453
API.Content.Asset.GetViewUrl 453
API.Content.Asset.MakeCopyDMSAsset 454
API.Content.Asset.SetAssetMgtConfigInfo 454
API.Content.Blog 454
API.Content.Blog.AddBlog 455
API.Content.Blog.AddCommentForPost 456
API.Content.Blog.AddPost 456
API.Content.Blog.DeleteBlog 458
API.Content.Blog.DeletePost 458
API.Content.Blog.GetBlankBlogPostData 459
API.Content.Blog.GetBlog 459, 462
API.Content.Blog.GetBlogData 463, 464, 465
API.Content.Blog.GetBlogDataForPost 465
API.Content.Blog.GetBlogPostData 466
API.Content.Blog.GetBlogRoll 466, 467
API.Content.Blog.GetPost 467
API.Content.Blog.GetPostbyID 468
API.Content.Content 468
API.Content.Content.AddContent 470, 472, 473
API.Content.Content.AddXmlConfiguration 474
API.Content.Content.CheckOutContent 474
API.Content.Content.CopyContentToFolder 474
API.Content.Content.DeleteContentItem 475
API.Content.Content.DeleteXmlConfiguration 476
API.Content.Content.GetAddViewLanguage 476
API.Content.Content.GetAllComments 477
API.Content.Content.GetAllTemplates 477
API.Content.Content.GetAllUnassignedItemApprovals 477
API.Content.Content.GetAllXmlConfigurations 478
API.Content.Content.GetChildContent 478
API.Content.Content.GetContent 479, 480
API.Content.Content.GetContentByHistory 481
API.Content.Content.GetContentDisplayLink 481
API.Content.Content.GetContentForEditing 482

API.Content.Content.GetContentReport 482
API.Content.Content.GetContentState 482
API.Content.Content.GetContentStatus 483
API.Content.Content.GetDomain 483
API.Content.Content.GetExpireContent 483
API.Content.Content.MoveContentToFolder 483
API.Content.Content.PublishContent 484
API.Content.Content.RelicateXmlConfiguration 484
API.Content.Content.RenameContent 485
API.Content.Content.ReplicateXmlConfiguration 485
API.Content.Content.RestoreHistoryContent 485
API.Content.Content.SaveContent 486
API.Content.Content.SubmitForDelete 486
API.Content.Content.UndoCheckout 487
API.Content.Content.UpdateContentMetaData 487
API.Content.Content.UpdatexmlCollectionPackage 487
API.Content.Content.UpdateXmlConfiguration 487
API.Content.ContentRating 488
API.Content.ContentRating.GetContentRating 488
API.Content.ContentRating.GetContentRatingAddRating 490
API.Content.ContentRating.GetContentRatingResults 491
API.Content.ContentRating.GetContentRatingStatistics 491, 492
API.Content.ContentRating.PurgeContentRatings 492
API.Content.Form 493
API.Content.Form.AssignFormTask 494
API.Content.Form.EmailFormFieldData 494
API.Content.Form.GetAllForms 495
API.Content.Form.GetForm 495
API.Content.Form.GetFormFieldData 496
API.Content.Form.GetFormFieldList 497
API.Content.Form.GetFormFieldQuestions 497
API.Content.Form.GetFormTitle 497
API.Content.Form.SubmitFormFieldData 498
API.Content.Form.UpdateFormFieldData 498
API.Content.Taxonomy 498
API.Content.Taxonomy.AddTaxonomyItem 499, 500
API.Content.Taxonomy.AddTaxonomySyncFolder 500
API.Content.Taxonomy.CreateTaxonomy 500
API.Content.Taxonomy.DeleteTaxonomy 500
API.Content.Taxonomy.GetTaxonomyIdByPath 501
API.Content.Taxonomy.ImportTaxonomy 501
API.Content.Taxonomy.LoadTaxonomy 501
API.Content.Taxonomy.ReadAllAssignedCategory 502
API.Content.Taxonomy.ReadTaxonomy 502
API.Content.Taxonomy.RemoveTaxonomyFolder 502
API.Content.Taxonomy.RemoveTaxonomyItem 502
API.Content.Taxonomy.ReOrderTaxonomyItems 503
API.Content.Taxonomy.UpdateTaxonomy 503
API.Content.ThreadedDiscussion 503
API.Content.ThreadedDiscussion.AddBoard 504
API.Content.ThreadedDiscussion.AddForum 505
API.Content.ThreadedDiscussion.AddTopic 507, 508
API.Content.ThreadedDiscussion.DeleteBoard 508
API.Content.ThreadedDiscussion.DeleteForum 508
API.Content.ThreadedDiscussion.DeleteTopic 508
API.Content.ThreadedDiscussion.GetBoardCategories 509
API.Content.ThreadedDiscussion.GetBoardForums 509
API.Content.ThreadedDiscussion.GetDiscussionBoard 510
API.Content.ThreadedDiscussion.GetDiscussionBoardString 510
API.Content.ThreadedDiscussion.GetForum 511
API.Content.ThreadedDiscussion.GetRepliesForTopics 512
API.Content.ThreadedDiscussion.GetTopic 513
API.Content.ThreadedDiscussion.SearchReplies 513
API.CustomFields 367
API.CustomFields.AddNewEditableCustomFieldAssignments 369
API.CustomFields.BoolToYesNo 369
API.CustomFields.GetEditableCustomFieldAssignments 369
API.CustomFields.GetEnhancedMetadataArea 370
API.CustomFields.GetEnhancedMetadataScript 370
API.CustomFields.GetFieldsByFolder 370
API.CustomFields.GetSearchProperties 371, 372
API.CustomFields.Populate_AssetRequestObjectFromForm 372
API.CustomFields.ProcessCustomFields 373
API.CustomFields.SearchAllAssets 373
API.CustomFields.WriteFilteredMetadataForEdit 373
API.CustomFields.WriteFilteredMetadataForView 374
API.CustomFields.WriteLibrarySearchExtended 374

API.CustomFields.WriteMetadataDefaultForEdit 374
API.CustomFields.WriteMetadataTypeForView 375
API.CustomFields.WritePublicSearch 375
API.CustomFields.WritePublicSearchEx 376
API.DateTimeSelector 377
API.DateTimeSelector.ClientDateSeparator 379
API.DateTimeSelector.ClientLongDatePattern 379
API.DateTimeSelector.ClientLongTimePattern 380
API.DateTimeSelector.ClientShortDatePattern 380
API.DateTimeSelector.ClientShortTimePattern 380
API.DateTimeSelector.ClientTimeSeparator 380
API.DateTimeSelector.DisplayCultureDate 377
API.DateTimeSelector.DisplayCultureDateTime 378
API.DateTimeSelector.DisplayCultureTime 378
API.DateTimeSelector.ExtendedMeta 381
API.DateTimeSelector.FormElement 381
API.DateTimeSelector.FormName 381
API.DateTimeSelector.RequestInfo 381
API.DateTimeSelector.RoundMinutes 378
API.DateTimeSelector.ServerDateSeparator 382
API.DateTimeSelector.ServerShortDatePattern 382
API.DateTimeSelector.ServerShortTimePattern 382
API.DateTimeSelector.ServerTimeSeparator 382
API.DateTimeSelector.SpanAttributes 383
API.DateTimeSelector.SpanId 383
API.DateTimeSelector.TargetDate 383
API.DateTimeSelector.ValidationType 383
API.Folder 383
API.Folder.AddCommunityFolder 384
API.Folder.AddFolder 385
API.Folder.CreateFolder 386
API.Folder.DeleteFolderById 386
API.Folder.GetChildFolders 387
API.Folder.GetFolder 389, 390
API.Folder.GetFolderDataWithPermission 390
API.Folder.GetFolderId 390, 391
API.Folder.GetFolderInfo 391
API.Folder.GetPath 391
API.Folder.RenameFolder 392
API.Folder.UpdateFolder 392
API.Font 392
API.Font.AddFont 393
API.Font.DeleteFont 393
API.Font.GetAllFonts 393
API.Font.GetFont 394
API.Font.GetFontConfigList 394
API.Font.UpdateFont 394
API.Library 395
API.Library.AddLibraryItem 396
API.Library.AddLoadBalanceItem 396
API.Library.DeleteLibraryItem 396
API.Library.DeleteLibraryItemById 396
API.Library.DeleteLoadBalanceItem 397
API.Library.GetAllChildLibItems 397
API.Library.GetAllLoadBalancePaths 399, 400
API.Library.GetLibraryItem 400
API.Library.GetLibrarySettings 401
API.Library.GetLibraryTypes 401
API.Library.GetLoadBalancePath 401
API.Library.InternalLibrarySearch 402
API.Library.UpdateLibrary 402
API.Library.UpdateLibrarySettings 402
API.Library.UpdateLoadBalanceSettings 403
API.Library.UpdateQlinkTemplates 403
API.Localization 403
API.Localization.GetAddViewLanguage 404
API.Localization.GetAddViewLanguageForAllContent 405
API.Localization.GetAddViewLanguageForMenus 405
API.Localization.GetContentItems 405
API.Localization.GetFile 406
API.Localization.GetFiles 406
API.Localization.GetFilesInZipFile 406
API.Localization.GetFlagUrl 407, 408
API.Localization.GetInterchangeFile 408
API.Localization.GetInterchangeFiles 408
API.Localization.GetInterchangeFilesInZipFile 409
API.Localization.GetJobs 410
API.Localization.GetLocalizationUrl 410

API.Localization.GetTranslationUploadDirectory 410
API.Localization.StartExportContentForTranslation 411
API.Localization.StartExportFolderForTranslation 411
API.Localization.StartExportMenusForTranslation 412
API.Localization.StartImportTranslation 412
API.Metadata 412
API.Metadata.AddMetaDataType 413, 414
API.Metadata.DeleteMetadataType 415
API.Metadata.GetContentMetadataList 416, 417
API.Metadata.GetIntegratedSearchFolder 418
API.Metadata.GetIntegratedSearchFolders 418
API.Metadata.GetMetadataType 418
API.Metadata.GetMetaDataTypes 419
API.Metadata.UpdateIntegratedSearchFolder 419
API.Metadata.UpdateMetadataType 419
API.Permissions 419
API.Permissions.AddItemPermission 420
API.Permissions.DeleteItemPermission 420
API.Permissions.DisableContentInheritance 421
API.Permissions.DisableFolderInheritance 421
API.Permissions.EnableContentInheritance 421
API.Permissions.EnableFolderInheritance 422
API.Permissions.GetUserPermissions 422, 423
API.Permissions.LoadPermissions 423
API.Permissions.UpdateItemPermission 424
API.Search 514
API.Search namespace 514
Classes 514
API.Search.SearchManager 515
API.Search.SearchManager.Execute 516
API.Search.SearchManager.Search 517, 518
API.Search.SearchManager.SearchForCategories 519
API.Search.SearchManager.SearchTable 519
API.Site 424
API.Site.BreakTaskPermissionInherit 425
API.Site.DeleteCookie 425
API.Site.GetAllActiveLanguages 425
API.Site.GetAllLanguages 426
API.Site.GetClientDayNames 426
API.Site.GetClientMonthNames 427
API.Site.GetCookieValue 427
API.Site.GetLanguage 427
API.Site.GetLanguages 428
API.Site.GetSiteVariables 428
API.Site.GetTaskPermission 428
API.Site.IsTaskPermissionInherited 429
API.Site.SetCookieValue 429
API.Site.SetTaskPermission 429
API.Site.UpdateSiteEnabledLanguages 429
API.Site.UpdateSiteVariables 430
API.SiteMap 430
API.SiteMap.GetFolderBreadcrumbPath 431
API.SiteMap.GetWebsiteMap 432
API.SiteMap.RenderFolderBreadcrumbHtml 434
API.SiteMap.RenderWebsiteMap 434
API.Subscription 435
API.Subscription.AddEmailFrom 436
API.Subscription.AddSubscription 436
API.Subscription.AddSubscriptionMessage 436
API.Subscription.DeleteEmailFrom 437
API.Subscription.DeleteSubscription 437
API.Subscription.DeleteSubscriptionMessage 437
API.Subscription.GetAllActiveSubscriptions 437
API.Subscription.GetAllEmailFrom 438
API.Subscription.GetAllSubscriptionMessages 438
API.Subscription.GetAllSubscriptions 438
API.Subscription.GetFolderInheritedFrom 438
API.Subscription.GetSubscription 439
API.Subscription.GetSubscriptionMessage 439
API.Subscription.GetSubscriptionMessagesForType 439
API.Subscription.GetSubscriptionPropertiesForContent 439
API.Subscription.GetSubscriptionPropertiesForFolder 440
API.Subscription.GetSubscriptionsForContent 440
API.Subscription.GetSubscriptionsForFolder 440
API.Subscription.UpdateEmailFrom 441
API.Subscription.UpdateSubscription 441
API.Subscription.UpdateSubscriptionMessage 441
API.Task 519

API.Task namespace 519
Classes 520
API.Task.Task 520
API.Task.Task.AddTask 521
API.Task.Task.AssignTaskToUser 521, 522
API.Task.Task.AssignTaskToUsergroup 522, 523
API.Task.Task.DeleteTask 523, 524
API.Task.Task.DeleteTasks 524, 525
API.Task.Task.GetTask 525
API.Task.Task.GetTasks 525, 526
API.Task.Task.GetTasksAssignedByUser 526
API.Task.Task.GetTasksAssignedToUser 527
API.Task.Task.GetTasksAssignedToUsergroup 528
API.Task.Task.GetTaskState 529
API.Task.Task.SetStateForTask 529
API.Task.Task.UpdateTask 529
API.Task.TaskCategory 530
API.Task.TaskCategory.AddTaskCategory 530
API.Task.TaskCategory.DeleteTaskCategory 531
API.Task.TaskCategory.GetAllCategories 531
API.Task.TaskCategory.GetTaskCategory 531
API.Task.TaskCategory.UpdateTaskCategory 531
API.Task.TaskCategoryType 532
API.Task.TaskCategoryType.AddTaskType 532
API.Task.TaskCategoryType.DeleteTaskType 533
API.Task.TaskCategoryType.GetTaskType 533
API.Task.TaskCategoryType.UpdateTaskType 533
API.User 534
API.User namespace 534
Classes 534
API.User.ADUser 534
API.User.ADUser.AddADGroupToCMS 535
API.User.ADUser.AddADUsersToCMSByUsername 536
API.User.ADUser.DeSynchUserGroups 536
API.User.ADUser.DeSynchUsers 536
API.User.ADUser.GetADMapping 536
API.User.ADUser.GetADStatus 537
API.User.ADUser.GetAvailableADGroups 537
API.User.ADUser.GetAvailableADUsers 537
API.User.ADUser.GetCMSGroupsToSync 538
API.User.ADUser.GetCMSRelationshipsToSync 538
API.User.ADUser.GetCMSUsersToSync 538
API.User.ADUser.GetDomains 539
API.User.ADUser.MapCMSUserGroupToAD 539
API.User.ADUser.MapCMSUserToAD 539
API.User.ADUser.SynchCMSGroupsToAD 540
API.User.ADUser.SynchCMSRelationshipsToAD 540
API.User.ADUser.SynchCMSUsersToAD 540
API.User.ADUser.UpdateADMapping 541
API.User.ADUser.UpdateADUsersGroups 541
API.User.ADUser.UpdateUsersInfoFromAD 541
API.User.User 542
API.User.User.ActivateUserAccount 544
API.User.User.AddMembershipUser 545
API.User.User.AddNewUser 547
API.User.User.AddRolePermission 547
API.User.User.AddUser 548
API.User.User.AddUserGroup 548
API.User.User.AddUserToGroup 549
API.User.User.AutoLogInUser 549
API.User.User.CreateBlankUserObject 550
API.User.User.DeleteUserByID 550
API.User.User.DeleteUserByIds 550
API.User.User.DeleteUserFromGroup 551
API.User.User.DeleteUserGroup 551
API.User.User.EditUserCustomProperties 551
API.User.User.GetActiveUser 552
API.User.User.GetADMapping 552
API.User.User.GetAllCmsUsers 552
API.User.User.GetAllCustomProperties 553
API.User.User.GetAllUserByCustomProperty 553
API.User.User.GetAllUserGroups 554
API.User.User.GetAllUsers 555
API.User.User.GetGroupsUserIsIn 555
API.User.User.GetMembershipUser 556
API.User.User.GetUser 556
API.User.User.GetUserByUsername 556
API.User.User.GetUserCustomProperties 558

API.User.User.GetUserEmailInfo 558
API.User.User.GetUserForFolderAdmin 559
API.User.User.GetUserGroup 559
API.User.User.GetUserGroupByName 559
API.User.User.GetUserGroupForFolderAdmin 559
API.User.User.GetUserPermissions 560
API.User.User.GetUserPreference 560
API.User.User.GetUsers 561
API.User.User.GetUsersForTask 562
API.User.User.GetUsersNotInGroup 562
API.User.User.IsAdmin 563
API.User.User.IsARoleMember 564, 565
API.User.User.IsARoleMemberForContent 566
API.User.User.IsARoleMemberForContent_FolderUserAdmin 566
API.User.User.IsARoleMemberForFolder 566
API.User.User.IsARoleMemberForFolder_FolderUserAdmin 567
API.User.User.IsARoleMemberForFolder_MetadataAdmin 567
API.User.User.LogInUser 568
API.User.User.ReadCustomProperties 570
API.User.User.ResetMembershipUserPassword 570
API.User.User.SetAuthenticationCookie 571
API.User.User.UnsubscribeUser 571
API.User.User.UpdateCustomPropertiesItemOrder 571
API.User.User.UpdateUser 572
API.User.User.UpDateUserGroup 573
API.User.User.UpdateUserPreferences 573
API.User.User.UserIdAuthenticated 575
API.User.User.UserObject 574
API.Utilities 441
API.Utilities.LoadPermissions 442
Asset class 450, 601
about Asset class 450, 601
Asset methods 450, 602
GetAssetMgtConfigInfo 450, 602
GetAssetMgtInfo 451, 603
GetAssetSupertypes 451, 603
GetCreateDragDropSnippet 451, 603
GetCreateEktronExplorerSnippet 452, 604
GetCreateSnippet 452, 604
GetDragDropWOSnippet 452, 604
GetEditSnippet 453, 605
GetMyAssetMgtID 453, 605
GetViewUrl 453
MakeCopyDMSAsset 454, 606
SetAssetMgtConfigInfo 454, 606
AssetControl class 19
about AssetControl class 19
AssetControl methods 19
AssetControl properties 20
DefaultAssetID 20
DefaultFolderID 21
DynamicParameter 21
Fill 20
UploadType 21

B

Blog class 22, 454, 607
about Blog class 22, 454, 607
AddBlog 455, 608
AddCommentForPost 456, 608
AddPost 456, 609
ArchiveMode 24
Blog methods 22, 455, 607
Blog properties 23
BlogID 24
CacheInterval 24
DateToStart 25
DeleteBlog 458, 609
DeletePost 458, 610
DynamicParameter 25
EditorHeight 25
EditorWidth 26
Fill 22
GetBlankBlogPostData 459, 610
GetBlog 459, 462, 611, 612
GetBlog method 459, 610
GetBlogData 463, 464, 465, 613, 614, 615
GetBlogData method 462, 612

GetBlogDataForPost 465, 615
GetBlogPostData 466, 616
GetBlogRoll 466, 467, 617
GetBlogRoll method 466, 616
GetPost 467, 618
GetPostbyID 468, 618
JavascriptEditorHTMLMode 26
MaxResults 26
PostParameter 27
RecentPosts 27
ShowHeader 27
ShowRSS 27
ToString 23
Wrap 23
BlogArchive class 28
about BlogArchive class 28
ArchiveMode 29
BlogArchive methods 28
BlogArchive properties 29
BlogID 29
CacheInterval 30
Fill 28
Text 30
Wrap 29
WrapTag 30
BlogCalendar class 30
about BlogCalendar class 30
BlogCalendar methods 31
BlogCalendar properties 31
BlogID 32
CacheInterval 32
Fill 31
Text 32
Wrap 31
WrapTag 33
BlogCategories class 33
about BlogCategories class 33
BlogCategories methods 33
BlogCategories properties 34
BlogID 34
CacheInterval 35
Fill 34
Text 35
Wrap 34
WrapTag 35
BlogEntries class 36
about BlogEntries class 36
BlogEntries methods 36
BlogEntries properties 38
BlogID 38
BlogPostParameter 39
CacheInterval 39
DateToStart 39
DisplayXslt 40
Fill 36
GetRssFeed 37
GetView 37
GetViewNames 37
MaxResults 40
ShowHeader 40
ToString 37
Wrap 37
WrapTag 41
XmlDoc 41
BlogPost class 41
about BlogPost class 41
BlogPost methods 42
BlogPost properties 42
DefaultContentID 42
ShowType 43
Wrap 42
BlogRecentPosts class 43
about BlogRecentPosts class 43
BlogID 45
BlogRecentPosts methods 43
BlogRecentPosts properties 44
CacheInterval 45
Fill 44

NumberofPosts 45
Text 46
Wrap 44
WrapTag 46
BlogRoll class 46
about BlogRoll class 46
BlogID 47
BlogRoll methods 46
BlogRoll properties 47
CacheInterval 48
Fill 47
Text 48
Wrap 47
WrapTag 48
BlogRSS class 49
about BlogRSS class 49
BlogID 50
BlogRSS methods 49
BlogRSS properties 49
CacheInterval 50
Fill 49
Text 50
WrapTag 51
BreadCrumb class 51
about BreadCrumb class 51
BreadCrumb methods 51
BreadCrumb properties 53
CurrentPageIndicator 54
DefaultContentID 54
DisplayStyle 54
DisplayTitle 55
DynamicParameter 55
Fill 52
GenerateTrail_HTML 52
IconAlt 55
IconPath 55
LinkLastItem 56
LinkTarget 56
MaxItems 56
MetadataName 56
Mode 57
ResetTrail 52, 53
ResetTrail method 52
separator 57
BreadCrumbItem class 57
about BreadCrumbItem class 57
BreadCrumbItem 58
BreadCrumbItem constructor 58
BreadCrumbItem methods 58
BreadCrumbItem properties 59
CaseSensitiveUrlComparison 60
DisplayTitle 60
Equals 59
GetHashCode 59
IconAlt 60
IconPath 60
RawUrl 60
Target 61
ToString 59
BusinessRule class 365
about BusinessRule class 365
BusinessRule methods 365
GetBusinessRules 366
GetBusinessRules method 366
BusinessRules class 61, 619
about BusinessRules class 61, 619
BusinessRules methods 61, 619
BusinessRules properties 62
Fill 62
GetBusinessRules 619, 620
GetBusinessRules method 619
Parameters 62
Result 63
RuleID 63
Text 63
Wrap 62

Copyright (c) Ektron Inc. 2007

C

Calendar class 64, 448, 621
about Calendar class 64, 448, 621
Calendar methods 64
Calendar properties 65
DefaultCalendarID 65
DisplayType 66
DynamicParameter 66
EndDate 66
Fill 64
HighlightEventID 67
HighlightEventTypeID 67
ShowXML 67
StartDate 68
Text 68
Wrap 64
CalendarEvent class 448, 621
about CalendarEvent class 448, 621
CalendarEventType class 449, 621
about CalendarEventType class 449, 621
Collection class 68
about Collection class 68
AddText 72
CacheInterval 73
Collection methods 69
Collection properties 72
CollectionDescription 73
CollectionTitle 73
ContentParameter 74
CopyTo 69
Count 74
Current 74
DefaultCollectionID 74
DynamicParameter 75
EkItems 75
EnablePaging 75
Fill 69
GetEnumerator 70
GetHtml 76
GetRssFeed 70
GetView 70
GetViewNames 70
IncludeIcons 76
IsSynchronized 76
LinkTarget 77
MarkupLanguage 77
MaxResults 77
MoveNext 71
Random 77
Reset 71
SuppressWrapperTags 78
SyncRoot 78
this 71
ToString 71
Wrap 71
XmlDoc 78
Content class 468, 622
about Content class 468, 622
AddContent 470, 472, 473, 623, 624, 625
AddContent method 470, 623
AddXmlConfiguration 474
CheckOutContent 474, 625
Content methods 468, 622
CopyContentToFolder 474, 626
DeleteContentItem 475, 626
DeleteXmlConfiguration 476
GetAddViewLanguage 476, 627
GetAllComments 477, 627
GetAllTemplates 477, 628
GetAllUnassignedItemApprovals 477, 628
GetAllXmlConfigurations 478
GetChildContent 478, 628
GetContent 479, 480, 629
GetContent method 478
GetContentByHistory 481, 629
GetContentDisplayLink 481
GetContentForEditing 482, 629

GetContentReport 482
GetContentState 482, 630
GetContentStatus 483, 630
GetDomain 483, 631
GetExpireContent 483
MoveContentToFolder 483, 631
PublishContent 484, 631
RelicateXmlConfiguration 484
RenameContent 485, 632
ReplicateXmlConfiguration 485
RestoreHistoryContent 485
SaveContent 486, 632
SubmitForDelete 486, 633
UndoCheckout 487, 633
UpdateContentMetaData 487, 633
UpdatexmlCollectionPackage 487
UpdateXmlConfiguration 487
ContentBlock class 78
about ContentBlock class 78
CacheInterval 81
ContentBlock methods 79
ContentBlock properties 81
CopyTo 79
Count 82
Current 82
DefaultContentID 82
DisplayXslt 83
DynamicParameter 83
EkItem 83
Fill 80
GetEnumerator 80
IsSynchronized 84
MoveNext 80
OverrideXslt 84
Reset 80
StartingFolderID 84
SyncRoot 85
this 80
ToString 80
Wrap 81
ContentData 738
ContentData class 738
about ContentData class 738
ApprovalMethod 741
Approver 742
AssetData 742
Comment 742
ContentData properties 739
ContType 742
DateCreated 743
DisplayDateCreated 743
DisplayEndDate 743
DisplayGoLive 744
DisplayLastEditDate 744
EditorFirstName 744
EditorLastName 745
EditorUserNames 745
EndDate 745
EndDateAction 745
FolderId 746
FolderName 746
GoLive 746
HistoryId 746
Html 747
HyperLink 747
Id 747
Image 748
ImageThumbnail 748
InheritedFrom 748
IsInherited 748
IsPrivate 749
IsPublished 749
IsSearchable 749
IsXmlInherited 750
LanguageId 750
LastEditDate 750
ManualAlias 750
ManualAliasId 751

MetaData 751
Path 751
Permissions 751
Quicklink 752
Status 752
StyleSheet 752
Teaser 753
TemplateConfiguration 753
Title 753
UserId 754
UserName 754
XmlConfiguration 754
XmlInheritedFrom 754
ContentData.ApprovalMethod 741
ContentData.Approver 742
ContentData.AssetData 742
ContentData.Comment 742
ContentData.ContType 742
ContentData.DateCreated 743
ContentData.DisplayDateCreated 743
ContentData.DisplayEndDate 743
ContentData.DisplayGoLive 744
ContentData.DisplayLastEditDate 744
ContentData.EditorFirstName 744
ContentData.EditorLastName 745
ContentData.EditorUserNames 745
ContentData.EndDate 745
ContentData.EndDateAction 745
ContentData.FolderId 746
ContentData.FolderName 746
ContentData.GoLive 746
ContentData.HistoryId 746
ContentData.Html 747
ContentData.HyperLink 747
ContentData.Id 747
ContentData.Image 748
ContentData.ImageThumbnail 748
ContentData.InheritedFrom 748
ContentData.IsInherited 748
ContentData.IsPrivate 749
ContentData.IsPublished 749
ContentData.IsSearchable 749
ContentData.IsXmlInherited 750
ContentData.LanguageId 750
ContentData.LastEditDate 750
ContentData.ManualAlias 750
ContentData.ManualAliasId 751
ContentData.MetaData 751
ContentData.Path 751
ContentData.Permissions 751
ContentData.Quicklink 752
ContentData.Status 752
ContentData.StyleSheet 752
ContentData.Teaser 753
ContentData.TemplateConfiguration 753
ContentData.Title 753
ContentData.UserId 754
ContentData.UserName 754
ContentData.XmlConfiguration 754
ContentData.XmlInheritedFrom 754
ContentInfo class 327
about ContentInfo class 327
AssetID 328
Comments 328
ContentInfo 327
ContentInfo properties 327
ErrorMessage 328
FileAndPath 328
FileExtension 328
FileName 329
FolderID 329
Html 329
ID 329
Language 329
MimeType 329
Status 330
Title 330
Type 330

UserID 330
ContentList class 85
about ContentList class 85
CacheInterval 89
ContentIds 89
ContentList methods 85
ContentList properties 88
CopyTo 86
Count 89
Current 89
DefaultContentID 90
Direction 90
DynamicParameter 90
EkItems 91
Fill 86
GetEnumerator 86
GetHtml 91
GetRssFeed 86
IncludeIcons 91
IsSynchronized 91
LinkTarget 92
MarkupLanguage 92
MetaTag 92
MoveNext 87
OrderKey 93
Random 93
Reset 87
SuppressWrapperTags 93
SyncRoot 93
this 87
ToString 87
Wrap 88
XmlDoc 94
ContentRating class 94, 488, 634
about ContentRating class 94, 488, 634
AddContentRating 634
AlreadyRatedMessage 96
BadLabel 96
CharsRemainingLabel 96
CompleteGraphDisplay 97
ContentRating methods 94, 488, 634
ContentRating properties 95
CookiesRequiredMessage 97
DefaultContentID 98
DynamicParameter 98
FeedbackHeader 98
Fill 94
GetContentRating 488
GetContentRatingAddRating 490
GetContentRatingResults 491, 635
GetContentRatingStatistics 491, 492
GetContentRatingStatistics method 491
GoodLabel 99
GraphBarColor 99
InitialGraphDisplay 99
Padding 100
PurgeContentRatings 492, 635
RatingLevelLabel 100
ThankYouMessage 100
ToString 95
TotalRatingsLabel 101
UserCommentsHeader 101
ValidationMessage 102
Wrap 95
ContentReview class 102
about ContentReview class 102
ContentReview methods 102
ContentReview properties 103
DefaultContentID 104
DisplayXslt 105
DynamicParameter 105
Fill 103
GetReviews 106
MaxReviews 106
Moderate 106
RatingsMinimum 107
ToString 103
UserID 107

Wrap 103
XmlDoc 107
CustomFields class 367, 636
about CustomFields class 367, 636
AddNewEditableCustomFieldAssignments 369, 637
BoolToYesNo 369, 638
CustomFields methods 367, 636
GetEditableCustomFieldAssignments 369, 638
GetEnhancedMetadataArea 370, 639
GetEnhancedMetadataScript 370, 639
GetFieldsByFolder 370, 640
GetSearchProperties 371, 372, 640
GetSearchProperties method 371
Populate_AssetRequestObjectFromForm 372
ProcessCustomFields 373, 640
SearchAllAssets 373, 641
WriteFilteredMetadataForEdit 373, 641
WriteFilteredMetadataForView 374, 642
WriteLibrarySearchExtended 374, 642
WriteMetadataDefaultForEdit 374, 643
WriteMetadataTypeForView 375, 643
WritePublicSearch 375, 643
WritePublicSearchEx 376

D

Data Classes 738
DateTimeSelector class 377
about DateTimeSelector class 377
ClientDateSeparator 379
ClientLongDatePattern 379
ClientLongTimePattern 380
ClientShortDatePattern 380
ClientShortTimePattern 380
ClientTimeSeparator 380
DateTimeSelector methods 377
DateTimeSelector properties 379
DisplayCultureDate 377
DisplayCultureDateTime 378
DisplayCultureTime 378
ExtendedMeta 381
FormElement 381
FormName 381
RequestInfo 381
RoundMinutes 378
ServerDateSeparator 382
ServerShortDatePattern 382
ServerShortTimePattern 382
ServerTimeSeparator 382
SpanAttributes 383
SpanId 383
TargetDate 383
ValidationType 383
DesignTimeDiagnostic class 107
about DesignTimeDiagnostic class 107
Authenticated 109
DesignTimeDiagnostic methods 108
DesignTimeDiagnostic properties 108
Fill 108
Hide 109
Language 109
SuppressWrapperTags 109
WrapTag 109
DhtmlMenu class 110
about DhtmlMenu class 110
AccessKey 112
Align 112
BackColor 112
BorderColor 112
BorderStyle 112
BorderWidth 113
CssClass 113
DhtmlMenu 110
DhtmlMenu methods 110
DhtmlMenu properties 111
DisplayXslt 113
Enabled 113
EnableIE6FormsFix 113
Fill 110

Height 114
HorizontalDisplay 114
MenuHoverOverColor 114
MenuStyle 115
Stylesheet 115
SubMenuBackColor 115
SubMenuHoverOverColor 116
SubMenuItemHeight 116
SubMenuItemWidth 116
SubMenuTopOffset 117
TabIndex 117
ToolTip 117
Width 117
Wrap 111
Directory class 118
about Directory class 118
CacheInterval 122
CopyTo 119
Count 122
Current 123
Directory methods 118
Directory properties 121
DisplayXslt 123
DynamicParameter 123
EkItems 123
EnableAjax 124
EnablePaging 124
EnableSearch 124
Fill 119
FolderID 124
GetEnumerator 119
GetView 119
GetViewNames 119
IsSynchronized 125
LinkTarget 125
MarkupLanguage 125
MaxResults 126
MoveNext 120
Recursive 126
Reset 120
ShowAllChecked 126
ShowCount 126
SyncRoot 127
TaxonomyCols 127
TaxonomyDepth 127
TaxonomyDescription 127
TaxonomyId 128
TaxonomyItemCols 128
TaxonomyPath 128
TaxonomyTitle 128
TaxonomyTreeData 128
TaxonomyXml 129
this 120
ToString 120
Wrap 121
XmlDoc 129

E

EkEditorPart class 10
about EkEditorPart class 10
Ektron.Cms 7
Ektron.Cms namespace 7
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.Fill 14
Ektron.Cms.Controls.ActiveTopics.MaxNumber 16
Ektron.Cms.Controls.ActiveTopics.ToString 14
Ektron.Cms.Controls.ActiveTopics.URLPath 16
Ektron.Cms.Controls.ActiveTopics.Wrap 15
Ektron.Cms.Controls.AnalyticsTracker 17
Ektron.Cms.Controls.AnalyticsTracker.DefaultContentID 18
Ektron.Cms.Controls.AnalyticsTracker.DynamicParameter 18
Ektron.Cms.Controls.AnalyticsTracker.EnableAnalytics 19

Ektron.Cms.Controls.AnalyticsTracker.Wrap 18
Ektron.Cms.Controls.AssetControl 19
Ektron.Cms.Controls.AssetControl.DefaultAssetID 20
Ektron.Cms.Controls.AssetControl.DefaultFolderID 21
Ektron.Cms.Controls.AssetControl.DynamicParameter 21
Ektron.Cms.Controls.AssetControl.Fill 20
Ektron.Cms.Controls.AssetControl.UploadType 21
Ektron.Cms.Controls.Blog 22
Ektron.Cms.Controls.Blog.ArchiveMode 24
Ektron.Cms.Controls.Blog.BlogID 24
Ektron.Cms.Controls.Blog.CacheInterval 24
Ektron.Cms.Controls.Blog.DateToStart 25
Ektron.Cms.Controls.Blog.DynamicParameter 25
Ektron.Cms.Controls.Blog.EditorHeight 25
Ektron.Cms.Controls.Blog.EditorWidth 26
Ektron.Cms.Controls.Blog.Fill 22
Ektron.Cms.Controls.Blog.JavascriptEditorHTMLMode 26
Ektron.Cms.Controls.Blog.MaxResults 26
Ektron.Cms.Controls.Blog.PostParameter 27
Ektron.Cms.Controls.Blog.RecentPosts 27
Ektron.Cms.Controls.Blog.ShowHeader 27
Ektron.Cms.Controls.Blog.ShowRSS 27
Ektron.Cms.Controls.Blog.ToString 23
Ektron.Cms.Controls.Blog.Wrap 23
Ektron.Cms.Controls.BlogArchive 28
Ektron.Cms.Controls.BlogArchive.ArchiveMode 29
Ektron.Cms.Controls.BlogArchive.BlogID 29
Ektron.Cms.Controls.BlogArchive.CacheInterval 30
Ektron.Cms.Controls.BlogArchive.Fill 28
Ektron.Cms.Controls.BlogArchive.Text 30
Ektron.Cms.Controls.BlogArchive.Wrap 29
Ektron.Cms.Controls.BlogArchive.WrapTag 30
Ektron.Cms.Controls.BlogCalendar 30
Ektron.Cms.Controls.BlogCalendar.BlogID 32
Ektron.Cms.Controls.BlogCalendar.CacheInterval 32
Ektron.Cms.Controls.BlogCalendar.Fill 31
Ektron.Cms.Controls.BlogCalendar.Text 32
Ektron.Cms.Controls.BlogCalendar.Wrap 31
Ektron.Cms.Controls.BlogCalendar.WrapTag 33
Ektron.Cms.Controls.BlogCategories 33
Ektron.Cms.Controls.BlogCategories.BlogID 34
Ektron.Cms.Controls.BlogCategories.CacheInterval 35
Ektron.Cms.Controls.BlogCategories.Fill 34
Ektron.Cms.Controls.BlogCategories.Text 35
Ektron.Cms.Controls.BlogCategories.Wrap 34
Ektron.Cms.Controls.BlogCategories.WrapTag 35
Ektron.Cms.Controls.BlogEntries 36
Ektron.Cms.Controls.BlogEntries.BlogID 38
Ektron.Cms.Controls.BlogEntries.BlogPostParameter 39
Ektron.Cms.Controls.BlogEntries.CacheInterval 39
Ektron.Cms.Controls.BlogEntries.DateToStart 39
Ektron.Cms.Controls.BlogEntries.DisplayXslt 40
Ektron.Cms.Controls.BlogEntries.Fill 36
Ektron.Cms.Controls.BlogEntries.GetRssFeed 37
Ektron.Cms.Controls.BlogEntries.GetView 37
Ektron.Cms.Controls.BlogEntries.GetViewNames 37
Ektron.Cms.Controls.BlogEntries.MaxResults 40
Ektron.Cms.Controls.BlogEntries.ShowHeader 40
Ektron.Cms.Controls.BlogEntries.ToString 37
Ektron.Cms.Controls.BlogEntries.Wrap 37
Ektron.Cms.Controls.BlogEntries.WrapTag 41
Ektron.Cms.Controls.BlogEntries.XmlDoc 41
Ektron.Cms.Controls.BlogPost 41
Ektron.Cms.Controls.BlogPost.DefaultContentID 42
Ektron.Cms.Controls.BlogPost.ShowType 43
Ektron.Cms.Controls.BlogPost.Wrap 42
Ektron.Cms.Controls.BlogRecentPosts 43
Ektron.Cms.Controls.BlogRecentPosts.BlogID 45
Ektron.Cms.Controls.BlogRecentPosts.CacheInterval 45
Ektron.Cms.Controls.BlogRecentPosts.Fill 44
Ektron.Cms.Controls.BlogRecentPosts.NumberofPosts 45
Ektron.Cms.Controls.BlogRecentPosts.Text 46
Ektron.Cms.Controls.BlogRecentPosts.Wrap 44
Ektron.Cms.Controls.BlogRecentPosts.WrapTag 46
Ektron.Cms.Controls.BlogRoll 46
Ektron.Cms.Controls.BlogRoll.BlogID 47
Ektron.Cms.Controls.BlogRoll.CacheInterval 48
Ektron.Cms.Controls.BlogRoll.Fill 47

Ektron.Cms.Controls.BlogRoll.Text 48
Ektron.Cms.Controls.BlogRoll.Wrap 47
Ektron.Cms.Controls.BlogRoll.WrapTag 48
Ektron.Cms.Controls.BlogRSS 49
Ektron.Cms.Controls.BlogRSS.BlogID 50
Ektron.Cms.Controls.BlogRSS.CacheInterval 50
Ektron.Cms.Controls.BlogRSS.Fill 49
Ektron.Cms.Controls.BlogRSS.Text 50
Ektron.Cms.Controls.BlogRSS.WrapTag 51
Ektron.Cms.Controls.BreadCrumb 51
Ektron.Cms.Controls.BreadCrumb.CurrentPageIndicator 54
Ektron.Cms.Controls.BreadCrumb.DefaultContentID 54
Ektron.Cms.Controls.BreadCrumb.DisplayStyle 54
Ektron.Cms.Controls.BreadCrumb.DisplayTitle 55
Ektron.Cms.Controls.BreadCrumb.DynamicParameter 55
Ektron.Cms.Controls.BreadCrumb.Fill 52
Ektron.Cms.Controls.BreadCrumb.GenerateTrail_HTML 52
Ektron.Cms.Controls.BreadCrumb.IconAlt 55
Ektron.Cms.Controls.BreadCrumb.IconPath 55
Ektron.Cms.Controls.BreadCrumb.LinkLastItem 56
Ektron.Cms.Controls.BreadCrumb.LinkTarget 56
Ektron.Cms.Controls.BreadCrumb.MaxItems 56
Ektron.Cms.Controls.BreadCrumb.MetadataName 56
Ektron.Cms.Controls.BreadCrumb.Mode 57
Ektron.Cms.Controls.BreadCrumb.ResetTrail 52, 53
Ektron.Cms.Controls.BreadCrumb.separator 57
Ektron.Cms.Controls.BreadCrumbItem 57
Ektron.Cms.Controls.BreadCrumbItem.BreadCrumbItem 58
Ektron.Cms.Controls.BreadCrumbItem.CaseSensitiveUrlComparison 60
Ektron.Cms.Controls.BreadCrumbItem.DisplayTitle 60
Ektron.Cms.Controls.BreadCrumbItem.Equals 59
Ektron.Cms.Controls.BreadCrumbItem.GetHashCode 59
Ektron.Cms.Controls.BreadCrumbItem.IconAlt 60
Ektron.Cms.Controls.BreadCrumbItem.IconPath 60
Ektron.Cms.Controls.BreadCrumbItem.RawUrl 60
Ektron.Cms.Controls.BreadCrumbItem.Target 61
Ektron.Cms.Controls.BreadCrumbItem.ToString 59
Ektron.Cms.Controls.BusinessRules 61
Ektron.Cms.Controls.BusinessRules.Fill 62
Ektron.Cms.Controls.BusinessRules.Parameters 62
Ektron.Cms.Controls.BusinessRules.Result 63
Ektron.Cms.Controls.BusinessRules.RuleID 63
Ektron.Cms.Controls.BusinessRules.Text 63
Ektron.Cms.Controls.BusinessRules.Wrap 62
Ektron.Cms.Controls.Calendar 64
Ektron.Cms.Controls.Calendar.DefaultCalendarID 65
Ektron.Cms.Controls.Calendar.DisplayType 66
Ektron.Cms.Controls.Calendar.DynamicParameter 66
Ektron.Cms.Controls.Calendar.EndDate 66
Ektron.Cms.Controls.Calendar.Fill 64
Ektron.Cms.Controls.Calendar.HighlightEventID 67
Ektron.Cms.Controls.Calendar.HighlightEventTypeID 67
Ektron.Cms.Controls.Calendar.ShowXML 67
Ektron.Cms.Controls.Calendar.StartDate 68
Ektron.Cms.Controls.Calendar.Text 68
Ektron.Cms.Controls.Calendar.Wrap 64
Ektron.Cms.Controls.Collection 68
Ektron.Cms.Controls.Collection.AddText 72
Ektron.Cms.Controls.Collection.CacheInterval 73
Ektron.Cms.Controls.Collection.CollectionDescription 73
Ektron.Cms.Controls.Collection.CollectionTitle 73
Ektron.Cms.Controls.Collection.ContentParameter 74
Ektron.Cms.Controls.Collection.CopyTo 69
Ektron.Cms.Controls.Collection.Count 74
Ektron.Cms.Controls.Collection.Current 74
Ektron.Cms.Controls.Collection.DefaultCollectionID 74
Ektron.Cms.Controls.Collection.DynamicParameter 75
Ektron.Cms.Controls.Collection.EkItems 75
Ektron.Cms.Controls.Collection.EnablePaging 75
Ektron.Cms.Controls.Collection.Fill 69
Ektron.Cms.Controls.Collection.GetEnumerator 70
Ektron.Cms.Controls.Collection.GetHtml 76
Ektron.Cms.Controls.Collection.GetRssFeed 70
Ektron.Cms.Controls.Collection.GetView 70
Ektron.Cms.Controls.Collection.GetViewNames 70
Ektron.Cms.Controls.Collection.IncludeIcons 76
Ektron.Cms.Controls.Collection.IsSynchronized 76

Ektron.Cms.Controls.Collection.LinkTarget 77
Ektron.Cms.Controls.Collection.MarkupLanguage 77
Ektron.Cms.Controls.Collection.MaxResults 77
Ektron.Cms.Controls.Collection.MoveNext 71
Ektron.Cms.Controls.Collection.Random 77
Ektron.Cms.Controls.Collection.Reset 71
Ektron.Cms.Controls.Collection.SuppressWrapperTags 78
Ektron.Cms.Controls.Collection.SyncRoot 78
Ektron.Cms.Controls.Collection.this 71
Ektron.Cms.Controls.Collection.ToString 71
Ektron.Cms.Controls.Collection.Wrap 71
Ektron.Cms.Controls.Collection.XmlDoc 78
Ektron.Cms.Controls.ContentBlock 78
Ektron.Cms.Controls.ContentBlock.CacheInterval 81
Ektron.Cms.Controls.ContentBlock.CopyTo 79
Ektron.Cms.Controls.ContentBlock.Count 82
Ektron.Cms.Controls.ContentBlock.Current 82
Ektron.Cms.Controls.ContentBlock.DefaultContentID 82
Ektron.Cms.Controls.ContentBlock.DisplayXslt 83
Ektron.Cms.Controls.ContentBlock.DynamicParameter 83
Ektron.Cms.Controls.ContentBlock.EkItem 83
Ektron.Cms.Controls.ContentBlock.Fill 80
Ektron.Cms.Controls.ContentBlock.GetEnumerator 80
Ektron.Cms.Controls.ContentBlock.IsSynchronized 84
Ektron.Cms.Controls.ContentBlock.MoveNext 80
Ektron.Cms.Controls.ContentBlock.OverrideXslt 84
Ektron.Cms.Controls.ContentBlock.Reset 80
Ektron.Cms.Controls.ContentBlock.StartingFolderID 84
Ektron.Cms.Controls.ContentBlock.SyncRoot 85
Ektron.Cms.Controls.ContentBlock.this 80
Ektron.Cms.Controls.ContentBlock.ToString 80
Ektron.Cms.Controls.ContentBlock.Wrap 81
Ektron.Cms.Controls.ContentList 85
Ektron.Cms.Controls.ContentList.CacheInterval 89
Ektron.Cms.Controls.ContentList.ContentIds 89
Ektron.Cms.Controls.ContentList.CopyTo 86
Ektron.Cms.Controls.ContentList.Count 89
Ektron.Cms.Controls.ContentList.Current 89
Ektron.Cms.Controls.ContentList.DefaultContentID 90
Ektron.Cms.Controls.ContentList.Direction 90
Ektron.Cms.Controls.ContentList.DynamicParameter 90
Ektron.Cms.Controls.ContentList.EkItems 91
Ektron.Cms.Controls.ContentList.Fill 86
Ektron.Cms.Controls.ContentList.GetEnumerator 86
Ektron.Cms.Controls.ContentList.GetHtml 91
Ektron.Cms.Controls.ContentList.GetRssFeed 86
Ektron.Cms.Controls.ContentList.IncludeIcons 91
Ektron.Cms.Controls.ContentList.IsSynchronized 91
Ektron.Cms.Controls.ContentList.LinkTarget 92
Ektron.Cms.Controls.ContentList.MarkupLanguage 92
Ektron.Cms.Controls.ContentList.MetaTag 92
Ektron.Cms.Controls.ContentList.MoveNext 87
Ektron.Cms.Controls.ContentList.OrderKey 93
Ektron.Cms.Controls.ContentList.Random 93
Ektron.Cms.Controls.ContentList.Reset 87
Ektron.Cms.Controls.ContentList.SuppressWrapperTags 93
Ektron.Cms.Controls.ContentList.SyncRoot 93
Ektron.Cms.Controls.ContentList.this 87
Ektron.Cms.Controls.ContentList.ToString 87
Ektron.Cms.Controls.ContentList.Wrap 88
Ektron.Cms.Controls.ContentList.XmlDoc 94
Ektron.Cms.Controls.ContentRating 94
Ektron.Cms.Controls.ContentRating.AlreadyRatedMessage 96
Ektron.Cms.Controls.ContentRating.BadLabel 96
Ektron.Cms.Controls.ContentRating.CharsRemainingLabel 96
Ektron.Cms.Controls.ContentRating.CompleteGraphDisplay 97
Ektron.Cms.Controls.ContentRating.CookiesRequiredMessage 97
Ektron.Cms.Controls.ContentRating.DefaultContentID 98
Ektron.Cms.Controls.ContentRating.DynamicParameter 98
Ektron.Cms.Controls.ContentRating.FeedbackHeader 98
Ektron.Cms.Controls.ContentRating.Fill 94
Ektron.Cms.Controls.ContentRating.GoodLabel 99
Ektron.Cms.Controls.ContentRating.GraphBarColor 99
Ektron.Cms.Controls.ContentRating.InitialGraphDisplay 99
Ektron.Cms.Controls.ContentRating.Padding 100
Ektron.Cms.Controls.ContentRating.RatingLevelLabel 100
Ektron.Cms.Controls.ContentRating.ThankYouMessage 100
Ektron.Cms.Controls.ContentRating.ToString 95

Ektron.Cms.Controls.ContentRating.TotalRatingsLabel 101
Ektron.Cms.Controls.ContentRating.UserCommentsHeader 101
Ektron.Cms.Controls.ContentRating.ValidationMessage 102
Ektron.Cms.Controls.ContentRating.Wrap 95
Ektron.Cms.Controls.ContentReview 102
Ektron.Cms.Controls.ContentReview.DefaultContentID 104
Ektron.Cms.Controls.ContentReview.DisplayXslt 105
Ektron.Cms.Controls.ContentReview.DynamicParameter 105
Ektron.Cms.Controls.ContentReview.Fill 103
Ektron.Cms.Controls.ContentReview.GetReviews 106
Ektron.Cms.Controls.ContentReview.MaxReviews 106
Ektron.Cms.Controls.ContentReview.Moderate 106
Ektron.Cms.Controls.ContentReview.RatingsMinimum 107
Ektron.Cms.Controls.ContentReview.ToString 103
Ektron.Cms.Controls.ContentReview.UserID 107
Ektron.Cms.Controls.ContentReview.Wrap 103
Ektron.Cms.Controls.ContentReview.XmlDoc 107
Ektron.Cms.Controls.DesignTimeDiagnostic 107
Ektron.Cms.Controls.DesignTimeDiagnostic.Authenticated 109
Ektron.Cms.Controls.DesignTimeDiagnostic.Fill 108
Ektron.Cms.Controls.DesignTimeDiagnostic.Hide 109
Ektron.Cms.Controls.DesignTimeDiagnostic.Language 109
Ektron.Cms.Controls.DesignTimeDiagnostic.SuppressWrapperTags 109
Ektron.Cms.Controls.DesignTimeDiagnostic.WrapTag 109
Ektron.Cms.Controls.DhtmlMenu 110
Ektron.Cms.Controls.DhtmlMenu.AccessKey 112
Ektron.Cms.Controls.DhtmlMenu.Align 112
Ektron.Cms.Controls.DhtmlMenu.BackColor 112
Ektron.Cms.Controls.DhtmlMenu.BorderColor 112
Ektron.Cms.Controls.DhtmlMenu.BorderStyle 112
Ektron.Cms.Controls.DhtmlMenu.BorderWidth 113
Ektron.Cms.Controls.DhtmlMenu.CssClass 113
Ektron.Cms.Controls.DhtmlMenu.DhtmlMenu 110
Ektron.Cms.Controls.DhtmlMenu.DisplayXslt 113
Ektron.Cms.Controls.DhtmlMenu.Enabled 113
Ektron.Cms.Controls.DhtmlMenu.EnableIE6FormsFix 113
Ektron.Cms.Controls.DhtmlMenu.Fill 110
Ektron.Cms.Controls.DhtmlMenu.Height 114
Ektron.Cms.Controls.DhtmlMenu.HorizontalDisplay 114
Ektron.Cms.Controls.DhtmlMenu.MenuHoverOverColor 114
Ektron.Cms.Controls.DhtmlMenu.MenuStyle 115
Ektron.Cms.Controls.DhtmlMenu.Stylesheet 115
Ektron.Cms.Controls.DhtmlMenu.SubMenuBackColor 115
Ektron.Cms.Controls.DhtmlMenu.SubMenuHoverOverColor 116
Ektron.Cms.Controls.DhtmlMenu.SubMenuItemHeight 116
Ektron.Cms.Controls.DhtmlMenu.SubMenuItemWidth 116
Ektron.Cms.Controls.DhtmlMenu.SubMenuTopOffset 117
Ektron.Cms.Controls.DhtmlMenu.TabIndex 117
Ektron.Cms.Controls.DhtmlMenu.ToolTip 117
Ektron.Cms.Controls.DhtmlMenu.Width 117
Ektron.Cms.Controls.DhtmlMenu.Wrap 111
Ektron.Cms.Controls.Directory 118
Ektron.Cms.Controls.Directory.CacheInterval 122
Ektron.Cms.Controls.Directory.CopyTo 119
Ektron.Cms.Controls.Directory.Count 122
Ektron.Cms.Controls.Directory.Current 123
Ektron.Cms.Controls.Directory.DisplayXslt 123
Ektron.Cms.Controls.Directory.DynamicParameter 123
Ektron.Cms.Controls.Directory.EkItems 123
Ektron.Cms.Controls.Directory.EnableAjax 124
Ektron.Cms.Controls.Directory.EnablePaging 124
Ektron.Cms.Controls.Directory.EnableSearch 124
Ektron.Cms.Controls.Directory.Fill 119
Ektron.Cms.Controls.Directory.FolderID 124
Ektron.Cms.Controls.Directory.GetEnumerator 119
Ektron.Cms.Controls.Directory.GetView 119
Ektron.Cms.Controls.Directory.GetViewNames 119
Ektron.Cms.Controls.Directory.IsSynchronized 125
Ektron.Cms.Controls.Directory.LinkTarget 125
Ektron.Cms.Controls.Directory.MarkupLanguage 125
Ektron.Cms.Controls.Directory.MaxResults 126
Ektron.Cms.Controls.Directory.MoveNext 120
Ektron.Cms.Controls.Directory.Recursive 126
Ektron.Cms.Controls.Directory.Reset 120
Ektron.Cms.Controls.Directory.ShowAllChecked 126
Ektron.Cms.Controls.Directory.ShowCount 126
Ektron.Cms.Controls.Directory.SyncRoot 127

Ektron.Cms.Controls.Directory.TaxonomyCols 127
Ektron.Cms.Controls.Directory.TaxonomyDepth 127
Ektron.Cms.Controls.Directory.TaxonomyDescription 127
Ektron.Cms.Controls.Directory.TaxonomyId 128
Ektron.Cms.Controls.Directory.TaxonomyItemCols 128
Ektron.Cms.Controls.Directory.TaxonomyPath 128
Ektron.Cms.Controls.Directory.TaxonomyTitle 128
Ektron.Cms.Controls.Directory.TaxonomyTreeData 128
Ektron.Cms.Controls.Directory.TaxonomyXml 129
Ektron.Cms.Controls.Directory.this 120
Ektron.Cms.Controls.Directory.ToString 120
Ektron.Cms.Controls.Directory.Wrap 121
Ektron.Cms.Controls.Directory.XmlDoc 129
Ektron.Cms.Controls.EktronCatalogPart 129
Ektron.Cms.Controls.EktronCatalogPart.BlogID 130
Ektron.Cms.Controls.EktronCatalogPart.BoardID 130
Ektron.Cms.Controls.EktronCatalogPart.CalendarDynamicParameter 131
Ektron.Cms.Controls.EktronCatalogPart.CollectionDynamicParameter 131
Ektron.Cms.Controls.EktronCatalogPart.ContentBlockDynamicParameter 131
Ektron.Cms.Controls.EktronCatalogPart.DefaultCalendarID 131
Ektron.Cms.Controls.EktronCatalogPart.DefaultCollectionID 132
Ektron.Cms.Controls.EktronCatalogPart.DefaultContentID 132
Ektron.Cms.Controls.EktronCatalogPart.DefaultFormID 132
Ektron.Cms.Controls.EktronCatalogPart.EnableBlogControl 132
Ektron.Cms.Controls.EktronCatalogPart.EnableCalendarControl 133
Ektron.Cms.Controls.EktronCatalogPart.EnableCollectionControl 133
Ektron.Cms.Controls.EktronCatalogPart.EnableContentBlockControl 133
Ektron.Cms.Controls.EktronCatalogPart.EnableDiscussionBoardsControl 133
Ektron.Cms.Controls.EktronCatalogPart.EnableFormBlockControl 133
Ektron.Cms.Controls.EktronCatalogPart.EnableListSummaryControl 134
Ektron.Cms.Controls.EktronCatalogPart.EnableThreadedDiscussionsControl 134
Ektron.Cms.Controls.EktronCatalogPart.FolderID 134
Ektron.Cms.Controls.EktronCatalogPart.FormBlockDynamicParameter 134
Ektron.Cms.Controls.EktronCatalogPart.Title 135
Ektron.Cms.Controls.EktronCommunityCatalogPart 135
Ektron.Cms.Controls.EktronCommunityCatalogPart.EnableBlogs 136
Ektron.Cms.Controls.EktronCommunityCatalogPart.EnableDiscussionBoards 136
Ektron.Cms.Controls.EktronCommunityCatalogPart.EnableForums 136
Ektron.Cms.Controls.EktronCommunityCatalogPart.Title 136
Ektron.Cms.Controls.EktronWebPartZone 137
Ektron.Cms.Controls.EktronWebPartZone.CmsSitePath 137
Ektron.Cms.Controls.EkWebControl 138
Ektron.Cms.Controls.EkWebControl.Authenticated 141
Ektron.Cms.Controls.EkWebControl.CmsSitePath 141
Ektron.Cms.Controls.EkWebControl.EkWebControl 138
Ektron.Cms.Controls.EkWebControl.Error 142
Ektron.Cms.Controls.EkWebControl.Fill 138
Ektron.Cms.Controls.EkWebControl.Hide 142
Ektron.Cms.Controls.EkWebControl.IsLoggedIn 142
Ektron.Cms.Controls.EkWebControl.Language 142
Ektron.Cms.Controls.EkWebControl.LanguageID 143
Ektron.Cms.Controls.EkWebControl.LocalizeString 139
Ektron.Cms.Controls.EkWebControl.loggedInUserID 143
Ektron.Cms.Controls.EkWebControl.loggedInUserName 143
Ektron.Cms.Controls.EkWebControl.RegisterClientScriptInclude 139
Ektron.Cms.Controls.EkWebControl.RegisterStylesheet 139
Ektron.Cms.Controls.EkWebControl.SearchHeader 140
Ektron.Cms.Controls.EkWebControl.SuppressWrapperTags 143
Ektron.Cms.Controls.EkWebControl.Text 144
Ektron.Cms.Controls.EkWebControl.ToString 140
Ektron.Cms.Controls.EkWebControl.Wrap 140

Ektron.Cms.Controls.EkWebControl.WrapTag 144
Ektron.Cms.Controls.EkWebPart 144
Ektron.Cms.Controls.EkXsltBase 145
Ektron.Cms.Controls.EkXsltBase.DisplayXslt 146
Ektron.Cms.Controls.EkXsltBase.EkXsltBase 145
Ektron.Cms.Controls.EkXsltBase.SetDirty 145
Ektron.Cms.Controls.EkXsltBase.XmlDoc 146
Ektron.Cms.Controls.EkXsltBase.XsltDoc 146
Ektron.Cms.Controls.FlexMenu 147
Ektron.Cms.Controls.FlexMenu.AutoCollapseBranches 148
Ektron.Cms.Controls.FlexMenu.CacheInterval 148
Ektron.Cms.Controls.FlexMenu.DisplayXslt 149
Ektron.Cms.Controls.FlexMenu.EnableAjax 149
Ektron.Cms.Controls.FlexMenu.EnableMouseOverPopUp 149
Ektron.Cms.Controls.FlexMenu.EnableSmartOpen 150
Ektron.Cms.Controls.FlexMenu.EnableSubmenuLinkPathMatching 150
Ektron.Cms.Controls.FlexMenu.EnableSubmenuTemplateMatching 150
Ektron.Cms.Controls.FlexMenu.Fill 147
Ektron.Cms.Controls.FlexMenu.MasterControlId 151
Ektron.Cms.Controls.FlexMenu.MenuDepth 151
Ektron.Cms.Controls.FlexMenu.SetDirty 147
Ektron.Cms.Controls.FlexMenu.StartCollapsed 151
Ektron.Cms.Controls.FlexMenu.StartLevel 152
Ektron.Cms.Controls.FlexMenu.Stylesheet 152
Ektron.Cms.Controls.FlexMenu.XmlDoc 152
Ektron.Cms.Controls.FlexMenu.XsltDoc 153
Ektron.Cms.Controls.FolderBreadcrumb 153
Ektron.Cms.Controls.FolderBreadcrumb.DefaultContentID 154
Ektron.Cms.Controls.FolderBreadcrumb.DefaultFolderID 155
Ektron.Cms.Controls.FolderBreadcrumb.DisplayStyle 155
Ektron.Cms.Controls.FolderBreadcrumb.DynamicParameter 155
Ektron.Cms.Controls.FolderBreadcrumb.Fill 153
Ektron.Cms.Controls.FolderBreadcrumb.LinkTarget 156
Ektron.Cms.Controls.FolderBreadcrumb.Mode 156
Ektron.Cms.Controls.FolderBreadcrumb.separator 156
Ektron.Cms.Controls.FormBlock 156
Ektron.Cms.Controls.FormBlock.AddValidation 160
Ektron.Cms.Controls.FormBlock.CacheInterval 160
Ektron.Cms.Controls.FormBlock.CopyTo 157
Ektron.Cms.Controls.FormBlock.Count 161
Ektron.Cms.Controls.FormBlock.Current 161
Ektron.Cms.Controls.FormBlock.DefaultFormID 161
Ektron.Cms.Controls.FormBlock.DynamicParameter 162
Ektron.Cms.Controls.FormBlock.EkItem 162
Ektron.Cms.Controls.FormBlock.Fields 162
Ektron.Cms.Controls.FormBlock.Fill 158
Ektron.Cms.Controls.FormBlock.FormName 162
Ektron.Cms.Controls.FormBlock.GetEnumerator 158
Ektron.Cms.Controls.FormBlock.GetView 158
Ektron.Cms.Controls.FormBlock.GetViewNames 158
Ektron.Cms.Controls.FormBlock.IncludeTags 163
Ektron.Cms.Controls.FormBlock.IsSynchronized 163
Ektron.Cms.Controls.FormBlock.MoveNext 158
Ektron.Cms.Controls.FormBlock.Reset 159
Ektron.Cms.Controls.FormBlock.StartingFolderID 163
Ektron.Cms.Controls.FormBlock.SyncRoot 164
Ektron.Cms.Controls.FormBlock.this 159
Ektron.Cms.Controls.FormBlock.ToString 159
Ektron.Cms.Controls.FormBlock.Wrap 159
Ektron.Cms.Controls.FormFieldItem 164
Ektron.Cms.Controls.FormFieldItem.Basetype 164
Ektron.Cms.Controls.FormFieldItem.ContentType 165
Ektron.Cms.Controls.FormFieldItem.DatalistItems 165
Ektron.Cms.Controls.FormFieldItem.Datatype 165
Ektron.Cms.Controls.FormFieldItem.DisplayName 166
Ektron.Cms.Controls.FormFieldItem.FieldName 166
Ektron.Cms.Controls.FormFieldItem.Language 166
Ektron.Cms.Controls.FormFieldItem.Value 167
Ektron.Cms.Controls.Forum 167
Ektron.Cms.Controls.Forum.BoardID 169
Ektron.Cms.Controls.Forum.CacheInterval 169
Ektron.Cms.Controls.Forum.DynamicParameter 169
Ektron.Cms.Controls.Forum.ExcludeList 170
Ektron.Cms.Controls.Forum.Fill 167
Ektron.Cms.Controls.Forum.HideUserProfile 170
Ektron.Cms.Controls.Forum.JavascriptEditorHeight 170

Ektron.Cms.Controls.Forum.JavascriptEditorHTMLMode 171
Ektron.Cms.Controls.Forum.JavascriptEditorToolbar 171
Ektron.Cms.Controls.Forum.JavascriptEditorWidth 172
Ektron.Cms.Controls.Forum.Theme 172
Ektron.Cms.Controls.Forum.ToString 168
Ektron.Cms.Controls.Forum.Wrap 168
Ektron.Cms.Controls.HtmlEditDisplay 172
Ektron.Cms.Controls.HtmlEditDisplay.BaseJavaScript 173
Ektron.Cms.Controls.HtmlEditDisplay.BeginJavaScript 174
Ektron.Cms.Controls.HtmlEditDisplay.EndJavaScript 174
Ektron.Cms.Controls.HtmlEditDisplay.MakeFieldNameFromId 174
Ektron.Cms.Controls.HtmlEditDisplay.RenderCMSFields 174
Ektron.Cms.Controls.HtmlEditDisplay.SelectedJavaScript 175
Ektron.Cms.Controls.HtmlEditor 175
Ektron.Cms.Controls.HtmlEditor.Archived 179
Ektron.Cms.Controls.HtmlEditor.Authenticated 180
Ektron.Cms.Controls.HtmlEditor.CmsSitePath 180
Ektron.Cms.Controls.HtmlEditor.ContentAction 181
Ektron.Cms.Controls.HtmlEditor.ContentID 181
Ektron.Cms.Controls.HtmlEditor.ContentTitle 182
Ektron.Cms.Controls.HtmlEditor.ContentType 182
Ektron.Cms.Controls.HtmlEditor.EditorReadyCall 183
Ektron.Cms.Controls.HtmlEditor.EnableUpload 183
Ektron.Cms.Controls.HtmlEditor.ExecuteContentAction 177
Ektron.Cms.Controls.HtmlEditor.HtmlEditor 176
Ektron.Cms.Controls.HtmlEditor.IncludeContentInfo 184
Ektron.Cms.Controls.HtmlEditor.IsLoggedIn 184
Ektron.Cms.Controls.HtmlEditor.Language 185
Ektron.Cms.Controls.HtmlEditor.loggedInUserID 185
Ektron.Cms.Controls.HtmlEditor.loggedInUserName 186
Ektron.Cms.Controls.HtmlEditor.Path 186
Ektron.Cms.Controls.HtmlEditor.PreferredType 186
Ektron.Cms.Controls.HtmlEditor.SafeEncode 187
Ektron.Cms.Controls.HtmlEditor.ShowCMSToolbar 188
Ektron.Cms.Controls.HtmlEditor.ShowTitle 188
Ektron.Cms.Controls.HtmlEditor.StyleSheet 188
Ektron.Cms.Controls.HtmlEditor.Text 189
Ektron.Cms.Controls.HtmlEditor.ToolbarLevel 189
Ektron.Cms.Controls.HtmlEditor.ToolbarResetCall 190
Ektron.Cms.Controls.HtmlEditor.ToString 177
Ektron.Cms.Controls.HtmlEditor.TranslationFile 191
Ektron.Cms.Controls.HtmlEditor.WorkareaMode 178
Ektron.Cms.Controls.ImageControl 191
Ektron.Cms.Controls.ImageControl.DefaultImageID 192
Ektron.Cms.Controls.ImageControl.DynamicParameter 193
Ektron.Cms.Controls.ImageControl.Fill 192
Ektron.Cms.Controls.ImageControl.FolderID 193
Ektron.Cms.Controls.ImageControl.Wrap 192
Ektron.Cms.Controls.IndexSearch 194
Ektron.Cms.Controls.IndexSearch.AddParm 195, 196
Ektron.Cms.Controls.IndexSearch.ButtonImgSrc 200
Ektron.Cms.Controls.IndexSearch.ButtonText 200
Ektron.Cms.Controls.IndexSearch.ContentParameter 201
Ektron.Cms.Controls.IndexSearch.CopyTo 196
Ektron.Cms.Controls.IndexSearch.Count 201
Ektron.Cms.Controls.IndexSearch.Current 201
Ektron.Cms.Controls.IndexSearch.EkItems 201
Ektron.Cms.Controls.IndexSearch.EmptyResultMsg 202
Ektron.Cms.Controls.IndexSearch.EncodeContentHtml 202
Ektron.Cms.Controls.IndexSearch.Fill 196
Ektron.Cms.Controls.IndexSearch.FolderId 202
Ektron.Cms.Controls.IndexSearch.FormName 203
Ektron.Cms.Controls.IndexSearch.GetEnumerator 196
Ektron.Cms.Controls.IndexSearch.GetSearchParmDataType 196
Ektron.Cms.Controls.IndexSearch.GetSearchParmSearchType 197
Ektron.Cms.Controls.IndexSearch.GetXPathValues 197
Ektron.Cms.Controls.IndexSearch.HitsMarker 203
Ektron.Cms.Controls.IndexSearch.IndexSearch 194
Ektron.Cms.Controls.IndexSearch.IsSynchronized 203
Ektron.Cms.Controls.IndexSearch.Labelafter 203
Ektron.Cms.Controls.IndexSearch.Labelbefore 204
Ektron.Cms.Controls.IndexSearch.LabelBetween 204
Ektron.Cms.Controls.IndexSearch.LabelContains 204
Ektron.Cms.Controls.IndexSearch.LabeldateBetween 205
Ektron.Cms.Controls.IndexSearch.LabelEqual 205
Ektron.Cms.Controls.IndexSearch.LabelExactPhrase 205
Ektron.Cms.Controls.IndexSearch.LabelGreaterThan 205

Ektron.Cms.Controls.IndexSearch.Labelinputdate 206
Ektron.Cms.Controls.IndexSearch.Labelinputnumber 206
Ektron.Cms.Controls.IndexSearch.LabelLessThan 206
Ektron.Cms.Controls.IndexSearch.LabelOn 207
Ektron.Cms.Controls.IndexSearch.LabelSelect 207
Ektron.Cms.Controls.IndexSearch.LinkTarget 207
Ektron.Cms.Controls.IndexSearch.MaxResults 208
Ektron.Cms.Controls.IndexSearch.MoveNext 197
Ektron.Cms.Controls.IndexSearch.OrderBy 208
Ektron.Cms.Controls.IndexSearch.OrderByDirection 208
Ektron.Cms.Controls.IndexSearch.Recursive 209
Ektron.Cms.Controls.IndexSearch.Reset 198
Ektron.Cms.Controls.IndexSearch.Search 198
Ektron.Cms.Controls.IndexSearch.SearchParmXML 209
Ektron.Cms.Controls.IndexSearch.SearchPram 194
Ektron.Cms.Controls.IndexSearch.SearchPram structure 194
Ektron.Cms.Controls.IndexSearch.ShowSearchBoxAlways 209
Ektron.Cms.Controls.IndexSearch.SyncRoot 210
Ektron.Cms.Controls.IndexSearch.this 198
Ektron.Cms.Controls.IndexSearch.Weighted 210
Ektron.Cms.Controls.IndexSearch.Wrap 198
Ektron.Cms.Controls.IndexSearch.XmlConfigId 210
Ektron.Cms.Controls.IndexSearch.XmlDoc 210
Ektron.Cms.Controls.LanguageAPI 211
Ektron.Cms.Controls.LanguageAPI.CurrentLanguage 213
Ektron.Cms.Controls.LanguageAPI.CurrentLanguageID 213
Ektron.Cms.Controls.LanguageAPI.DefaultLanguage 213
Ektron.Cms.Controls.LanguageAPI.DefaultLanguageID 214
Ektron.Cms.Controls.LanguageAPI.Fill 211
Ektron.Cms.Controls.LanguageAPI.GetLanguage 211
Ektron.Cms.Controls.LanguageAPI.GetLanguageID 212
Ektron.Cms.Controls.LanguageAPI.IsValid 212
Ektron.Cms.Controls.LanguageAPI.LangaugeIdList 214
Ektron.Cms.Controls.LanguageAPI.LangaugeTitleList 214
Ektron.Cms.Controls.LanguageAPI.LanguageAPI 211
Ektron.Cms.Controls.LanguageAPI.MultiLanguageEnabled 214
Ektron.Cms.Controls.LanguageAPI.OverrideDefault 214
Ektron.Cms.Controls.LanguageAPI.SiteLanguage 215
Ektron.Cms.Controls.LanguageAPI.SiteLanguageID 215
Ektron.Cms.Controls.LanguageSelect 215
Ektron.Cms.Controls.LanguageSelect.Fill 216
Ektron.Cms.Controls.LanguageSelect.LabelName 217
Ektron.Cms.Controls.LanguageSelect.Text 217
Ektron.Cms.Controls.LanguageSelect.Wrap 216
Ektron.Cms.Controls.ListSummary 217
Ektron.Cms.Controls.ListSummary.AddText 222
Ektron.Cms.Controls.ListSummary.CacheInterval 222
Ektron.Cms.Controls.ListSummary.ContentParameter 223
Ektron.Cms.Controls.ListSummary.ContentType 223
Ektron.Cms.Controls.ListSummary.CopyTo 218
Ektron.Cms.Controls.ListSummary.Count 224
Ektron.Cms.Controls.ListSummary.Current 224
Ektron.Cms.Controls.ListSummary.Direction 224
Ektron.Cms.Controls.ListSummary.EkItems 224
Ektron.Cms.Controls.ListSummary.EnablePaging 225
Ektron.Cms.Controls.ListSummary.Fill 218
Ektron.Cms.Controls.ListSummary.FolderDescription 225
Ektron.Cms.Controls.ListSummary.FolderID 225
Ektron.Cms.Controls.ListSummary.FolderName 225
Ektron.Cms.Controls.ListSummary.FolderPath 226
Ektron.Cms.Controls.ListSummary.GetEnumerator 218
Ektron.Cms.Controls.ListSummary.GetHtml 226
Ektron.Cms.Controls.ListSummary.GetRssFeed 219
Ektron.Cms.Controls.ListSummary.GetView 219
Ektron.Cms.Controls.ListSummary.GetViewNames 219
Ektron.Cms.Controls.ListSummary.IncludeIcons 226
Ektron.Cms.Controls.ListSummary.IsSynchronized 227
Ektron.Cms.Controls.ListSummary.LinkTarget 227
Ektron.Cms.Controls.ListSummary.MarkupLanguage 227
Ektron.Cms.Controls.ListSummary.MaxResults 227
Ektron.Cms.Controls.ListSummary.MoveNext 219
Ektron.Cms.Controls.ListSummary.OrderKey 228
Ektron.Cms.Controls.ListSummary.Random 228
Ektron.Cms.Controls.ListSummary.Recursive 228
Ektron.Cms.Controls.ListSummary.Reset 220
Ektron.Cms.Controls.ListSummary.SuppressWrapperTags 229
Ektron.Cms.Controls.ListSummary.SyncRoot 229
Ektron.Cms.Controls.ListSummary.this 220

Ektron.Cms.Controls.ListSummary.ToString 220
Ektron.Cms.Controls.ListSummary.Wrap 220
Ektron.Cms.Controls.ListSummary.XmlDoc 229
Ektron.Cms.Controls.Login 229
Ektron.Cms.Controls.Login.AutoAddType 230
Ektron.Cms.Controls.Login.AutoLogin 231
Ektron.Cms.Controls.Login.Fill 230
Ektron.Cms.Controls.Login.OnlyAllowMemberLogin 231
Ektron.Cms.Controls.Login.Password 231
Ektron.Cms.Controls.Login.PromptLogout 231
Ektron.Cms.Controls.Login.SuppressHelpButton 231
Ektron.Cms.Controls.Login.Text 232
Ektron.Cms.Controls.Login.UserName 232
Ektron.Cms.Controls.Map 232
Ektron.Cms.Controls.Map.Address 234
Ektron.Cms.Controls.Map.ContentId 234
Ektron.Cms.Controls.Map.CustomIcon 235
Ektron.Cms.Controls.Map.DateSearch 235
Ektron.Cms.Controls.Map.DisplayTextResult 235
Ektron.Cms.Controls.Map.DistanceUnit 236
Ektron.Cms.Controls.Map.EnableSearchResult 236
Ektron.Cms.Controls.Map.Fill 233
Ektron.Cms.Controls.Map.FolderId 236
Ektron.Cms.Controls.Map.GeoControl 237
Ektron.Cms.Controls.Map.GetCallbackResult 233
Ektron.Cms.Controls.Map.Latitude 237
Ektron.Cms.Controls.Map.Longitude 237
Ektron.Cms.Controls.Map.MapCategory 237
Ektron.Cms.Controls.Map.MapProvider 238
Ektron.Cms.Controls.Map.MapStyle 238
Ektron.Cms.Controls.Map.MarkupLanguage 238
Ektron.Cms.Controls.Map.MinZoomLevel 238
Ektron.Cms.Controls.Map.PageSize 238
Ektron.Cms.Controls.Map.RaiseCallbackEvent 233
Ektron.Cms.Controls.Map.Recursive 239
Ektron.Cms.Controls.Map.SelectedCategoryLogicalType 239
Ektron.Cms.Controls.Map.StartZoomLevel 239
Ektron.Cms.Controls.Map.TypeControl 239
Ektron.Cms.Controls.Map.ZoomControl 240
Ektron.Cms.Controls.Membership 240
Ektron.Cms.Controls.Membership.BoardID 242
Ektron.Cms.Controls.Membership.DisplayMode 242
Ektron.Cms.Controls.Membership.Fill 241
Ektron.Cms.Controls.Membership.RedirectFailedURL 242
Ektron.Cms.Controls.Membership.RedirectSuccessURL 243
Ektron.Cms.Controls.Membership.RegisterButtonImg 243
Ektron.Cms.Controls.Membership.RegisterButtonText 243
Ektron.Cms.Controls.Membership.ResetButtonImg 243
Ektron.Cms.Controls.Membership.ResetButtonText 243
Ektron.Cms.Controls.Membership.SaveButtonImg 244
Ektron.Cms.Controls.Membership.SaveButtonText 244
Ektron.Cms.Controls.Membership.ShowExtended 244
Ektron.Cms.Controls.Membership.ShowTerms 244
Ektron.Cms.Controls.Membership.Text 245
Ektron.Cms.Controls.Membership.UserExistsMessage 245
Ektron.Cms.Controls.Membership.UserID 245
Ektron.Cms.Controls.Membership.UserSuccessMessage 245
Ektron.Cms.Controls.Membership.UserUpdateSuccessMessage 246
Ektron.Cms.Controls.Menu 246
Ektron.Cms.Controls.Menu.CacheInterval 247
Ektron.Cms.Controls.Menu.DefaultMenuID 248
Ektron.Cms.Controls.Menu.DisplayXslt 248
Ektron.Cms.Controls.Menu.DynamicParameter 248
Ektron.Cms.Controls.Menu.Fill 246
Ektron.Cms.Controls.Menu.Stylesheet 249
Ektron.Cms.Controls.Menu.SuppressAddEdit 249
Ektron.Cms.Controls.Menu.Wrap 247
Ektron.Cms.Controls.MetaData 249
Ektron.Cms.Controls.MetaData.DefaultContentID 250
Ektron.Cms.Controls.MetaData.DefaultItemList 251
Ektron.Cms.Controls.MetaData.DynamicParameter 251
Ektron.Cms.Controls.MetaData.Fill 250
Ektron.Cms.Controls.MetaData.GenerateDublinCore 251
Ektron.Cms.Controls.MetaDataList 252
Ektron.Cms.Controls.MetaDataList.CacheInterval 255
Ektron.Cms.Controls.MetaDataList.ContentType 256
Ektron.Cms.Controls.MetaDataList.CopyTo 253

Ektron.Cms.Controls.MetaDataList.Count 256
Ektron.Cms.Controls.MetaDataList.Current 256
Ektron.Cms.Controls.MetaDataList.EkItems 256
Ektron.Cms.Controls.MetaDataList.EnablePaging 256
Ektron.Cms.Controls.MetaDataList.ExactPhrase 257
Ektron.Cms.Controls.MetaDataList.Fill 253
Ektron.Cms.Controls.MetaDataList.FolderID 257
Ektron.Cms.Controls.MetaDataList.GetEnumerator 253
Ektron.Cms.Controls.MetaDataList.GetHtml 258
Ektron.Cms.Controls.MetaDataList.IncludeIcons 258
Ektron.Cms.Controls.MetaDataList.IsSynchronized 258
Ektron.Cms.Controls.MetaDataList.KeyWordName 258
Ektron.Cms.Controls.MetaDataList.KeyWordValue 259
Ektron.Cms.Controls.MetaDataList.KeyWordValueMatchAll 259
Ektron.Cms.Controls.MetaDataList.KeyWordValueSeparator 259
Ektron.Cms.Controls.MetaDataList.LinkTarget 260
Ektron.Cms.Controls.MetaDataList.MarkupLanguage 260
Ektron.Cms.Controls.MetaDataList.MaxNumber 260
Ektron.Cms.Controls.MetaDataList.MoveNext 253
Ektron.Cms.Controls.MetaDataList.OrderBy 261
Ektron.Cms.Controls.MetaDataList.Recursive 261
Ektron.Cms.Controls.MetaDataList.Reset 253
Ektron.Cms.Controls.MetaDataList.SortOrder 261
Ektron.Cms.Controls.MetaDataList.SuppressWrapperTags 262
Ektron.Cms.Controls.MetaDataList.SyncRoot 262
Ektron.Cms.Controls.MetaDataList.this 254
Ektron.Cms.Controls.MetaDataList.Wrap 254
Ektron.Cms.Controls.MetaDataList.XmlDoc 262
Ektron.Cms.Controls.Personalization 10
Ektron.Cms.Controls.Personalization namespace 10
Classes 10
Ektron.Cms.Controls.Personalization.EkEditorPart 10
Ektron.Cms.Controls.PersonalizationManager 262
Ektron.Cms.Controls.PersonalizationManager.AddContentExitText 264
Ektron.Cms.Controls.PersonalizationManager.AddContentText 264
Ektron.Cms.Controls.PersonalizationManager.CurrentUserButtonText 264
Ektron.Cms.Controls.PersonalizationManager.Display 265
Ektron.Cms.Controls.PersonalizationManager.Fill 263
Ektron.Cms.Controls.PersonalizationManager.HeaderBackColor 265
Ektron.Cms.Controls.PersonalizationManager.HeaderText 265
Ektron.Cms.Controls.PersonalizationManager.PersonalizeExitText 266
Ektron.Cms.Controls.PersonalizationManager.PersonalizeScopeText 266
Ektron.Cms.Controls.PersonalizationManager.PersonalizeStartText 266
Ektron.Cms.Controls.PersonalizationManager.PublicUserButtonText 267
Ektron.Cms.Controls.PersonalizationManager.ResetLinkText 267
Ektron.Cms.Controls.PersonalizationManager.ResetLinkToolTip 267
Ektron.Cms.Controls.Poll 268
Ektron.Cms.Controls.Poll.EnableAjax 269
Ektron.Cms.Controls.Poll.Fill 268
Ektron.Cms.Controls.Poll.PollID 269
Ektron.Cms.Controls.PostHistory 270
Ektron.Cms.Controls.PostHistory.BoardID 271
Ektron.Cms.Controls.PostHistory.CacheInterval 272
Ektron.Cms.Controls.PostHistory.Fill 270
Ektron.Cms.Controls.PostHistory.MaxNumber 272
Ektron.Cms.Controls.PostHistory.ToString 270
Ektron.Cms.Controls.PostHistory.URLPath 272
Ektron.Cms.Controls.PostHistory.UserID 272
Ektron.Cms.Controls.PostHistory.Wrap 271
Ektron.Cms.Controls.RssAggregator 273
Ektron.Cms.Controls.RssAggregator.CopyTo 274
Ektron.Cms.Controls.RssAggregator.Count 276
Ektron.Cms.Controls.RssAggregator.Current 276
Ektron.Cms.Controls.RssAggregator.EkItems 276
Ektron.Cms.Controls.RssAggregator.Fill 274
Ektron.Cms.Controls.RssAggregator.GetEnumerator 274
Ektron.Cms.Controls.RssAggregator.GetView 274

Ektron.Cms.Controls.RssAggregator.GetViewNames 274
Ektron.Cms.Controls.RssAggregator.IsSynchronized 276
Ektron.Cms.Controls.RssAggregator.LinkTarget 277
Ektron.Cms.Controls.RssAggregator.MaxResults 277
Ektron.Cms.Controls.RssAggregator.MoveNext 275
Ektron.Cms.Controls.RssAggregator.Reset 275
Ektron.Cms.Controls.RssAggregator.SyncRoot 277
Ektron.Cms.Controls.RssAggregator.this 275
Ektron.Cms.Controls.RssAggregator.URL 278
Ektron.Cms.Controls.RssItemBase 278
Ektron.Cms.Controls.RssItemBase.DatePosted 278
Ektron.Cms.Controls.RssItemBase.Description 279
Ektron.Cms.Controls.RssItemBase.Link 279
Ektron.Cms.Controls.RssItemBase.Title 279
Ektron.Cms.Controls.Search 279
Ektron.Cms.Controls.Search.ButtonImgSrc 284
Ektron.Cms.Controls.Search.ButtonText 285
Ektron.Cms.Controls.Search.ContentFieldsetLegend 285
Ektron.Cms.Controls.Search.CopyTo 280
Ektron.Cms.Controls.Search.Count 285
Ektron.Cms.Controls.Search.Current 286
Ektron.Cms.Controls.Search.Display 286
Ektron.Cms.Controls.Search.EkItems 286
Ektron.Cms.Controls.Search.EmptyResultMsg 286
Ektron.Cms.Controls.Search.EnableAdvancedLink 287
Ektron.Cms.Controls.Search.EnableAjax 287
Ektron.Cms.Controls.Search.EnableArchived 287
Ektron.Cms.Controls.Search.EnableArchivedBtn 288
Ektron.Cms.Controls.Search.EnableContent 288
Ektron.Cms.Controls.Search.EnableContentBtn 288
Ektron.Cms.Controls.Search.EnableContentFieldset 288
Ektron.Cms.Controls.Search.EnableDMSAssets 289
Ektron.Cms.Controls.Search.EnableDMSAssetsBtn 289
Ektron.Cms.Controls.Search.EnableFolderFieldset 289
Ektron.Cms.Controls.Search.EnableForms 290
Ektron.Cms.Controls.Search.EnableFormsBtn 290
Ektron.Cms.Controls.Search.EnableLibFiles 290
Ektron.Cms.Controls.Search.EnableLibFilesBtn 291
Ektron.Cms.Controls.Search.EnableLibHyperlinks 291
Ektron.Cms.Controls.Search.EnableLibHyperlinksBtn 291
Ektron.Cms.Controls.Search.EnableLibImages 292
Ektron.Cms.Controls.Search.EnableLibImagesBtn 292
Ektron.Cms.Controls.Search.EnableLibraryFieldset 292
Ektron.Cms.Controls.Search.Fill 280
Ektron.Cms.Controls.Search.FolderFieldsetLegend 292
Ektron.Cms.Controls.Search.FolderID 293
Ektron.Cms.Controls.Search.FolderPath 293
Ektron.Cms.Controls.Search.GetEnumerator 281
Ektron.Cms.Controls.Search.IncludeIcons 293
Ektron.Cms.Controls.Search.IsSynchronized 294
Ektron.Cms.Controls.Search.LibraryFieldsetLegend 294
Ektron.Cms.Controls.Search.LinkTarget 294
Ektron.Cms.Controls.Search.LoadSearch 281
Ektron.Cms.Controls.Search.MaxCharacters 295
Ektron.Cms.Controls.Search.MaxResults 295
Ektron.Cms.Controls.Search.MaxTeaserLength 295
Ektron.Cms.Controls.Search.MoveNext 281
Ektron.Cms.Controls.Search.OrderBy 296
Ektron.Cms.Controls.Search.OrderDirection 296
Ektron.Cms.Controls.Search.PostResultsAt 296
Ektron.Cms.Controls.Search.Recursive 297
Ektron.Cms.Controls.Search.RemoveTeaserHtml 297
Ektron.Cms.Controls.Search.Reset 281
Ektron.Cms.Controls.Search.ResultTagId 297
Ektron.Cms.Controls.Search.Search 280
Ektron.Cms.Controls.Search.SearchDisplayUI 281
Ektron.Cms.Controls.Search.SearchText 298
Ektron.Cms.Controls.Search.SearchType 298
Ektron.Cms.Controls.Search.ShowDate 298
Ektron.Cms.Controls.Search.ShowExtendedSearch 298
Ektron.Cms.Controls.Search.ShowSearchBoxAlways 299
Ektron.Cms.Controls.Search.ShowSearchOptions 299
Ektron.Cms.Controls.Search.Stylesheet 299
Ektron.Cms.Controls.Search.SyncRoot 300
Ektron.Cms.Controls.Search.TargetFolderDescriptionList 300
Ektron.Cms.Controls.Search.TextBoxSize 300
Ektron.Cms.Controls.Search.this 282
Ektron.Cms.Controls.Search.Wrap 282

Ektron.Cms.Controls.Search.XmlDoc 301
Ektron.Cms.Controls.Sitemap 301
Ektron.Cms.Controls.Sitemap.CacheInterval 302
Ektron.Cms.Controls.Sitemap.ClassName 302
Ektron.Cms.Controls.Sitemap.DisplayType 303
Ektron.Cms.Controls.Sitemap.DisplayXslt 303
Ektron.Cms.Controls.Sitemap.Fill 301
Ektron.Cms.Controls.Sitemap.FlatTopLevel 304
Ektron.Cms.Controls.Sitemap.FolderID 304
Ektron.Cms.Controls.Sitemap.MaxLevel 304
Ektron.Cms.Controls.Sitemap.StartingLevel 305
Ektron.Cms.Controls.Sitemap.Stylesheet 305
Ektron.Cms.Controls.Sitemap.XmlDoc 305
Ektron.Cms.Controls.SmartMenu 305
Ektron.Cms.Controls.SmartMenu.AutoCollapseBranches 306
Ektron.Cms.Controls.SmartMenu.DisplayXslt 307
Ektron.Cms.Controls.SmartMenu.Enable508Compliance 307
Ektron.Cms.Controls.SmartMenu.Enable508NavigationHeadings 307
Ektron.Cms.Controls.SmartMenu.EnableMouseOverPopUp 308
Ektron.Cms.Controls.SmartMenu.EnableSmartOpen 308
Ektron.Cms.Controls.SmartMenu.EnableSubmenuLinkPathMatching 308
Ektron.Cms.Controls.SmartMenu.EnableSubmenuTemplateMatching 309
Ektron.Cms.Controls.SmartMenu.LaunchLinksInNewWindow 309
Ektron.Cms.Controls.SmartMenu.ShowRootFolder 310
Ektron.Cms.Controls.SmartMenu.SingleEditButton 310
Ektron.Cms.Controls.SmartMenu.StartCollapsed 310
Ektron.Cms.Controls.SmartMenu.StartWithRootCollapsed 311
Ektron.Cms.Controls.SmartMenu.Stylesheet 311
Ektron.Cms.Controls.WebSearch 311
Ektron.Cms.Controls.WebSearch.ButtonImgSrc 315
Ektron.Cms.Controls.WebSearch.ButtonText 316
Ektron.Cms.Controls.WebSearch.CustomSearch 316
Ektron.Cms.Controls.WebSearch.DisplayXslt 316
Ektron.Cms.Controls.WebSearch.EnableAdvancedLink 317
Ektron.Cms.Controls.WebSearch.EndCount 317
Ektron.Cms.Controls.WebSearch.Fill 312
Ektron.Cms.Controls.WebSearch.FolderID 317
Ektron.Cms.Controls.WebSearch.LinkTarget 318
Ektron.Cms.Controls.WebSearch.LoadSearch 312
Ektron.Cms.Controls.WebSearch.MarkupLanguage 318
Ektron.Cms.Controls.WebSearch.MaxCharacters 318
Ektron.Cms.Controls.WebSearch.MaxTeaserLength 319
Ektron.Cms.Controls.WebSearch.OrderBy 319
Ektron.Cms.Controls.WebSearch.OrderDirection 319
Ektron.Cms.Controls.WebSearch.Recursive 320
Ektron.Cms.Controls.WebSearch.RemoveTeaserHtml 320
Ektron.Cms.Controls.WebSearch.ResultCount 320
Ektron.Cms.Controls.WebSearch.ResultDataTable 321
Ektron.Cms.Controls.WebSearch.ResultsPageSize 321
Ektron.Cms.Controls.WebSearch.ResultTagId 321
Ektron.Cms.Controls.WebSearch.ResultXml 322
Ektron.Cms.Controls.WebSearch.SearchDisplayUI 313
Ektron.Cms.Controls.WebSearch.SearchFor 322
Ektron.Cms.Controls.WebSearch.SearchText 322
Ektron.Cms.Controls.WebSearch.ShowSearchBoxAlways 323
Ektron.Cms.Controls.WebSearch.ShowSearchOptions 323
Ektron.Cms.Controls.WebSearch.StartCount 324
Ektron.Cms.Controls.WebSearch.Stylesheet 324
Ektron.Cms.Controls.WebSearch.TextBoxSize 324
Ektron.Cms.Controls.WebSearch.TimeTaken 324
Ektron.Cms.Controls.WebSearch.WebSearch 312
Ektron.Cms.Controls.WebSearch.Wrap 313
Ektron.Cms.Controls.WebSearch.XmlDoc 325
Ektron.Cms.Extensibility 325
Ektron.Cms.Extensibility namespace 325
Classes 326
Ektron.Cms.Extensibility.ContentInfo 327
Ektron.Cms.Extensibility.ContentInfo.AssetID 328
Ektron.Cms.Extensibility.ContentInfo.Comments 328
Ektron.Cms.Extensibility.ContentInfo.ContentInfo 327
Ektron.Cms.Extensibility.ContentInfo.ErrorMessage 328
Ektron.Cms.Extensibility.ContentInfo.FileAndPath 328
Ektron.Cms.Extensibility.ContentInfo.FileExtension 328
Ektron.Cms.Extensibility.ContentInfo.FileName 329

Ektron.Cms.Extensibility.ContentInfo.FolderID 329
Ektron.Cms.Extensibility.ContentInfo.Html 329
Ektron.Cms.Extensibility.ContentInfo.ID 329
Ektron.Cms.Extensibility.ContentInfo.Language 329
Ektron.Cms.Extensibility.ContentInfo.MimeType 329
Ektron.Cms.Extensibility.ContentInfo.Status 330
Ektron.Cms.Extensibility.ContentInfo.Title 330
Ektron.Cms.Extensibility.ContentInfo.Type 330
Ektron.Cms.Extensibility.ContentInfo.UserID 330
Ektron.Cms.Extensibility.ExtensionAuthor 330
Ektron.Cms.Extensibility.ExtensionAuthor.Author 331
Ektron.Cms.Extensibility.ExtensionAuthor.ExtensionAuthor 331
Ektron.Cms.Extensibility.ExtensionAuthor.ToString 331
Ektron.Cms.Extensibility.ExtensionDescription 332
Ektron.Cms.Extensibility.ExtensionDescription.Desc 333
Ektron.Cms.Extensibility.ExtensionDescription.ExtensionDescription 332
Ektron.Cms.Extensibility.ExtensionDescription.ToString 332
Ektron.Cms.Extensibility.ExtensionEvent 333
Ektron.Cms.Extensibility.ExtensionEvent.Content 347
Ektron.Cms.Extensibility.ExtensionEvent.Folder 347
Ektron.Cms.Extensibility.ExtensionEvent.Form 348
Ektron.Cms.Extensibility.ExtensionEvent.GetActiveDirectoryUserProxy 335
Ektron.Cms.Extensibility.ExtensionEvent.GetAssetProxy 335
Ektron.Cms.Extensibility.ExtensionEvent.GetBlogProxy 336
Ektron.Cms.Extensibility.ExtensionEvent.GetBusinessRulesProxy 336
Ektron.Cms.Extensibility.ExtensionEvent.GetCalendarEventTypeProxy 336
Ektron.Cms.Extensibility.ExtensionEvent.GetCalendarProxy 336
Ektron.Cms.Extensibility.ExtensionEvent.GetContentProxy 337
Ektron.Cms.Extensibility.ExtensionEvent.GetContentRatingProxy 337
Ektron.Cms.Extensibility.ExtensionEvent.GetCustomFieldsProxy 337
Ektron.Cms.Extensibility.ExtensionEvent.GetFolderProxy 337
Ektron.Cms.Extensibility.ExtensionEvent.GetFontProxy 337
Ektron.Cms.Extensibility.ExtensionEvent.GetFormProxy 338
Ektron.Cms.Extensibility.ExtensionEvent.GetLibraryProxy 338
Ektron.Cms.Extensibility.ExtensionEvent.GetMetadataProxy 338
Ektron.Cms.Extensibility.ExtensionEvent.GetObjectStateAsXml 338
Ektron.Cms.Extensibility.ExtensionEvent.GetPermissionsProxy 339
Ektron.Cms.Extensibility.ExtensionEvent.GetSearchManagerProxy 339
Ektron.Cms.Extensibility.ExtensionEvent.GetSiteMapProxy 339
Ektron.Cms.Extensibility.ExtensionEvent.GetSiteProxy 339
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskCategoryProxy 340
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskCategoryTypeProxy 340
Ektron.Cms.Extensibility.ExtensionEvent.GetTaskProxy 340
Ektron.Cms.Extensibility.ExtensionEvent.GetThreadedDiscussionProxy 340
Ektron.Cms.Extensibility.ExtensionEvent.GetUserProxy 340
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentAdd 341
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentDelete 341
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterContentEdit 341
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderAdd 341
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderDelete 342
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFolderEdit 342
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterFormSubmit 342
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterPublish 342
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterReviewRejected 343
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserAdd 343
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserDelete 343
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserEdit 343
Ektron.Cms.Extensibility.ExtensionEvent.OnAfterUserLogin 344
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentAdd 344

Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentDelete 344
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeContentEdit 344
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderAdd 344
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderDelete 345
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFolderEdit 345
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeFormSubmit 345
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforePublish 345
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeReviewRejected 346
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserAdd 346
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserDelete 346
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserEdit 346
Ektron.Cms.Extensibility.ExtensionEvent.OnBeforeUserLogin 347
Ektron.Cms.Extensibility.ExtensionEvent.Site 348
Ektron.Cms.Extensibility.ExtensionEvent.User 348
Ektron.Cms.Extensibility.FolderInfo 348
Ektron.Cms.Extensibility.FolderInfo.Description 349
Ektron.Cms.Extensibility.FolderInfo.ErrorMessage 349
Ektron.Cms.Extensibility.FolderInfo.ID 349
Ektron.Cms.Extensibility.FolderInfo.Name 349
Ektron.Cms.Extensibility.FolderInfo.ParentID 350
Ektron.Cms.Extensibility.FolderInfo.ReplicationState 350
Ektron.Cms.Extensibility.FolderInfo.StyleSheet 350
Ektron.Cms.Extensibility.FolderInfo.Type 350
Ektron.Cms.Extensibility.FormField 350
Ektron.Cms.Extensibility.FormField.FieldName 351
Ektron.Cms.Extensibility.FormField.FieldValue 351
Ektron.Cms.Extensibility.FormField.FormField 351
Ektron.Cms.Extensibility.FormInfo 351
Ektron.Cms.Extensibility.FormInfo.ErrorMessage 352
Ektron.Cms.Extensibility.FormInfo.FormDescription 353
Ektron.Cms.Extensibility.FormInfo.FormFields 353
Ektron.Cms.Extensibility.FormInfo.FormTitle 353
Ektron.Cms.Extensibility.FormInfo.this 352
Ektron.Cms.Extensibility.RequestInfo 353
Ektron.Cms.Extensibility.ReviewInfo 353
Ektron.Cms.Extensibility.ReviewInfo.AverageRating 354
Ektron.Cms.Extensibility.ReviewInfo.ContentHtml 355
Ektron.Cms.Extensibility.ReviewInfo.ContentID 355
Ektron.Cms.Extensibility.ReviewInfo.ContentLanguage 355
Ektron.Cms.Extensibility.ReviewInfo.ContentTitle 355
Ektron.Cms.Extensibility.ReviewInfo.ErrorMessage 355
Ektron.Cms.Extensibility.ReviewInfo.ID 355
Ektron.Cms.Extensibility.ReviewInfo.NumberofRatings 356
Ektron.Cms.Extensibility.ReviewInfo.Rating 356
Ektron.Cms.Extensibility.ReviewInfo.RatingDate 356
Ektron.Cms.Extensibility.ReviewInfo.State 356
Ektron.Cms.Extensibility.ReviewInfo.UserAvatar 356
Ektron.Cms.Extensibility.ReviewInfo.UserComments 356
Ektron.Cms.Extensibility.ReviewInfo.UserDisplayName 357
Ektron.Cms.Extensibility.ReviewInfo.UserFirstName 357
Ektron.Cms.Extensibility.ReviewInfo.UserID 357
Ektron.Cms.Extensibility.ReviewInfo.UserLastName 357
Ektron.Cms.Extensibility.ReviewInfo.UserName 357
Ektron.Cms.Extensibility.ReviewInfo.VisitorID 357
Ektron.Cms.Extensibility.SiteInfo 358
Ektron.Cms.Extensibility.SiteInfo.AppPath 358
Ektron.Cms.Extensibility.SiteInfo.IsSSL 358
Ektron.Cms.Extensibility.SiteInfo.Name 359
Ektron.Cms.Extensibility.SiteInfo.PhysicalSitePath 359
Ektron.Cms.Extensibility.SiteInfo.Port 359
Ektron.Cms.Extensibility.SiteInfo.Url 359
Ektron.Cms.Extensibility.UserInfo 359
Ektron.Cms.Extensibility.UserInfo.DisplayName 360
Ektron.Cms.Extensibility.UserInfo.Editor 360
Ektron.Cms.Extensibility.UserInfo.EmailAddress 360
Ektron.Cms.Extensibility.UserInfo.ErrorMessage 361
Ektron.Cms.Extensibility.UserInfo.FirstName 361
Ektron.Cms.Extensibility.UserInfo.ID 361
Ektron.Cms.Extensibility.UserInfo.Language 361
Ektron.Cms.Extensibility.UserInfo.LastName 361
Ektron.Cms.Extensibility.UserInfo.Notifications 361

Ektron.Cms.Extensibility.UserInfo.Type 362
Ektron.Cms.Extensibility.UserInfo.UserName 362
Ektron.Services 575
Ektron.Services namespace 575
Classes 592
Ektron.Services.ActiveDirectoryUser 593
Ektron.Services.ActiveDirectoryUser.AddADGroupToCMS 594
Ektron.Services.ActiveDirectoryUser.DeSynchUserGroups 595
Ektron.Services.ActiveDirectoryUser.DeSynchUsers 595
Ektron.Services.ActiveDirectoryUser.GetADMapping 596
Ektron.Services.ActiveDirectoryUser.GetADStatus 596
Ektron.Services.ActiveDirectoryUser.GetAvailableADGroups 596
Ektron.Services.ActiveDirectoryUser.GetAvailableADUsers 597
Ektron.Services.ActiveDirectoryUser.GetCMSGroupsToSync 597
Ektron.Services.ActiveDirectoryUser.GetCMSRelationshipsToSync 598
Ektron.Services.ActiveDirectoryUser.GetCMSUsersToSync 598
Ektron.Services.ActiveDirectoryUser.GetDomains 599
Ektron.Services.ActiveDirectoryUser.MapCMSUserGroupToAD 599
Ektron.Services.ActiveDirectoryUser.MapCMSUserToAD 600
Ektron.Services.ActiveDirectoryUser.UpdateADMapping 600
Ektron.Services.ActiveDirectoryUser.UpdateADUsersGroups 601
Ektron.Services.ActiveDirectoryUser.UpdateUsersInfoFromAD 601
Ektron.Services.Asset 601
Ektron.Services.Asset.GetAssetMgtConfigInfo 602
Ektron.Services.Asset.GetAssetMgtInfo 603
Ektron.Services.Asset.GetAssetSupertypes 603
Ektron.Services.Asset.GetCreateDragDropSnippet 603
Ektron.Services.Asset.GetCreateEktronExplorerSnippet 604
Ektron.Services.Asset.GetCreateSnippet 604
Ektron.Services.Asset.GetDragDropWOSnippet 604
Ektron.Services.Asset.GetEditSnippet 605
Ektron.Services.Asset.GetMyAssetMgtID 605
Ektron.Services.Asset.MakeCopyDMSAsset 606
Ektron.Services.Asset.SetAssetMgtConfigInfo 606
Ektron.Services.Blog 607
Ektron.Services.Blog.AddBlog 608
Ektron.Services.Blog.AddCommentForPost 608
Ektron.Services.Blog.AddPost 609
Ektron.Services.Blog.DeleteBlog 609
Ektron.Services.Blog.DeletePost 610
Ektron.Services.Blog.GetBlankBlogPostData 610
Ektron.Services.Blog.GetBlog 611, 612
Ektron.Services.Blog.GetBlogData 613, 614, 615
Ektron.Services.Blog.GetBlogDataForPost 615
Ektron.Services.Blog.GetBlogPostData 616
Ektron.Services.Blog.GetBlogRoll 617
Ektron.Services.Blog.GetPost 618
Ektron.Services.Blog.GetPostbyID 618
Ektron.Services.BusinessRules 619
Ektron.Services.BusinessRules.GetBusinessRules 619, 620
Ektron.Services.Calendar 621
Ektron.Services.CalendarEvent 621
Ektron.Services.CalendarEventType 621
Ektron.Services.Content 622
Ektron.Services.Content.AddContent 623, 624, 625
Ektron.Services.Content.CheckOutContent 625
Ektron.Services.Content.CopyContentToFolder 626
Ektron.Services.Content.DeleteContentItem 626
Ektron.Services.Content.GetAddViewLanguage 627
Ektron.Services.Content.GetAllComments 627
Ektron.Services.Content.GetAllTemplates 628
Ektron.Services.Content.GetAllUnassignedItemApprovals 628
Ektron.Services.Content.GetChildContent 628
Ektron.Services.Content.GetContent 629
Ektron.Services.Content.GetContentByHistory 629
Ektron.Services.Content.GetContentForEditing 629
Ektron.Services.Content.GetContentState 630
Ektron.Services.Content.GetContentStatus 630
Ektron.Services.Content.GetDomain 631
Ektron.Services.Content.MoveContentToFolder 631
Ektron.Services.Content.PublishContent 631
Ektron.Services.Content.RenameContent 632
Ektron.Services.Content.SaveContent 632
Ektron.Services.Content.SubmitForDelete 633

Ektron.Services.Content.UndoCheckout 633
Ektron.Services.Content.UpdateContentMetaData 633
Ektron.Services.ContentRating 634
Ektron.Services.ContentRating.AddContentRating 634
Ektron.Services.ContentRating.GetContentRatingResults 635
Ektron.Services.ContentRating.PurgeContentRatings 635
Ektron.Services.CustomFields 636
Ektron.Services.CustomFields.AddNewEditableCustomFieldAssignments 637
Ektron.Services.CustomFields.BoolToYesNo 638
Ektron.Services.CustomFields.GetEditableCustomFieldAssignments 638
Ektron.Services.CustomFields.GetEnhancedMetadataArea 639
Ektron.Services.CustomFields.GetEnhancedMetadataScript 639
Ektron.Services.CustomFields.GetFieldsByFolder 640
Ektron.Services.CustomFields.GetSearchProperties 640
Ektron.Services.CustomFields.ProcessCustomFields 640
Ektron.Services.CustomFields.SearchAllAssets 641
Ektron.Services.CustomFields.WriteFilteredMetadataForEdit 641
Ektron.Services.CustomFields.WriteFilteredMetadataForView 642
Ektron.Services.CustomFields.WriteLibrarySearchExtended 642
Ektron.Services.CustomFields.WriteMetadataDefaultForEdit 643
Ektron.Services.CustomFields.WriteMetadataTypeForView 643
Ektron.Services.CustomFields.WritePublicSearch 643
Ektron.Services.Folder 644
Ektron.Services.Folder.AddFolder 645
Ektron.Services.Folder.CreateFolder 645
Ektron.Services.Folder.DeleteFolder 645
Ektron.Services.Folder.GetChildFolders 646
Ektron.Services.Folder.GetFolder 646
Ektron.Services.Folder.GetFolderID 647
Ektron.Services.Folder.GetPath 647
Ektron.Services.Folder.RenameFolder 648
Ektron.Services.Folder.UpdateFolder 648
Ektron.Services.Font 648
Ektron.Services.Font.AddFont 649
Ektron.Services.Font.DeleteFont 649
Ektron.Services.Font.GetAllFonts 650
Ektron.Services.Font.GetFont 650
Ektron.Services.Font.GetFontConfigList 651
Ektron.Services.Font.UpdateFont 651
Ektron.Services.Form 651
Ektron.Services.Form.AssignFormTask 652
Ektron.Services.Form.EmailFormFieldData 653
Ektron.Services.Form.GetAllForms 653, 654
Ektron.Services.Form.GetForm 654
Ektron.Services.Form.GetFormDataHistogram 654
Ektron.Services.Form.GetFormFieldData 655, 656
Ektron.Services.Form.GetFormFieldList 656
Ektron.Services.Form.GetFormFieldQuestions 657
Ektron.Services.Form.GetFormTitle 657
Ektron.Services.Form.SubmitFormFieldData 657
Ektron.Services.Form.UpdateFormFieldData 658
Ektron.Services.Library 658
Ektron.Services.Library.AddLoadBalanceItem 659
Ektron.Services.Library.DeleteLibraryItem 660
Ektron.Services.Library.DeleteLibraryItemById 660
Ektron.Services.Library.DeleteLoadBalanceItem 660
Ektron.Services.Library.GetAllChildLibItems 661
Ektron.Services.Library.GetAllLoadBalancePaths 662
Ektron.Services.Library.GetLibraryItem 663
Ektron.Services.Library.GetLibrarySettings 664
Ektron.Services.Library.GetLibraryTypes 664
Ektron.Services.Library.GetLoadBalancePath 665
Ektron.Services.Library.InternalLibrarySearch 665
Ektron.Services.Library.UpdateLibrary 665
Ektron.Services.Library.UpdateLibrarySettings 666
Ektron.Services.Library.UpdateLoadBalanceSettings 666
Ektron.Services.Library.UpdateQlinkTemplates 666
Ektron.Services.Metadata 667
Ektron.Services.Metadata.AddMetaDataType 667
Ektron.Services.Metadata.DeleteMetadataType 668
Ektron.Services.Metadata.GetContentMetadataList 668, 669
Ektron.Services.Metadata.GetMetadataType 669
Ektron.Services.Metadata.GetMetaDataTypes 670
Ektron.Services.Metadata.UpdateMetadataType 670

Ektron.Services.Permissions 671
Ektron.Services.Permissions.AddItemPermission 671
Ektron.Services.Permissions.DeleteItemPermission 672
Ektron.Services.Permissions.DisableContentInheritance 672
Ektron.Services.Permissions.DisableFolderInheritance 673
Ektron.Services.Permissions.EnableContentInheritance 673
Ektron.Services.Permissions.EnableFolderInheritance 674
Ektron.Services.Permissions.GetUserPermissions 674, 675
Ektron.Services.Permissions.LoadPermissions 675
Ektron.Services.Permissions.UpdateItemPermission 676
Ektron.Services.SearchManager 676
Ektron.Services.SearchManager.ExecuteSearch 677, 678
Ektron.Services.SearchManager.Search 678, 679
Ektron.Services.SearchManager.SearchForCategories 680
Ektron.Services.SearchManager.SearchTable 680
Ektron.Services.Site 681
Ektron.Services.Site.BreakTaskPermissionInherit 682
Ektron.Services.Site.GetAllActiveLanguages 682
Ektron.Services.Site.GetAllLanguages 683
Ektron.Services.Site.GetClientDayNames 684
Ektron.Services.Site.GetClientMonthNames 684
Ektron.Services.Site.GetCookieValue 684
Ektron.Services.Site.GetLanguage 685
Ektron.Services.Site.GetLanguages 685
Ektron.Services.Site.GetSiteVariables 685
Ektron.Services.Site.GetTaskPermission 686
Ektron.Services.Site.IsTaskPermissionInherited 686
Ektron.Services.Site.SetCookieValue 687
Ektron.Services.Site.SetTaskPermission 687
Ektron.Services.Site.UpdateSiteEnabledLanguages 688
Ektron.Services.SiteMap 688
Ektron.Services.SiteMap.GetFolderBreadcrumbPath 689
Ektron.Services.SiteMap.GetWebsiteMap 689
Ektron.Services.SiteMap.RenderFolderBreadcrumbHtml 690
Ektron.Services.SiteMap.RenderWebsiteMap 690
Ektron.Services.Task 576
Ektron.Services.Task namespace 576
Classes 576
Ektron.Services.Task.Task 576
Ektron.Services.Task.Task.AddTask 577
Ektron.Services.Task.Task.AssignTaskToUser 578
Ektron.Services.Task.Task.AssignTaskToUsergroup 579
Ektron.Services.Task.Task.DeleteTask 580
Ektron.Services.Task.Task.DeleteTasks 581, 582
Ektron.Services.Task.Task.GetTask 582
Ektron.Services.Task.Task.GetTasks 582
Ektron.Services.Task.Task.GetTasksAssignedByUser 583
Ektron.Services.Task.Task.GetTasksAssignedToUser 584
Ektron.Services.Task.Task.GetTasksAssignedToUsergroup 585, 586
Ektron.Services.Task.Task.GetTaskState 586
Ektron.Services.Task.Task.SetStateForTask 586
Ektron.Services.Task.Task.UpdateTask 587
Ektron.Services.Task.TaskCategory 587
Ektron.Services.Task.TaskCategory.AddTaskCategory 588
Ektron.Services.Task.TaskCategory.DeleteTaskCategory 588
Ektron.Services.Task.TaskCategory.GetAllCategories 589
Ektron.Services.Task.TaskCategory.GetTaskCategory 589
Ektron.Services.Task.TaskCategory.UpdateTaskCategory 590
Ektron.Services.Task.TaskCategoryType 590
Ektron.Services.Task.TaskCategoryType.AddTaskType 591
Ektron.Services.Task.TaskCategoryType.DeleteTaskType 591
Ektron.Services.Task.TaskCategoryType.GetTaskType 592
Ektron.Services.Task.TaskCategoryType.UpdateTaskType 592
Ektron.Services.Taxonomy 691
Ektron.Services.Taxonomy.AddTaxonomyItem 692
Ektron.Services.Taxonomy.AddTaxonomySyncFolder 692
Ektron.Services.Taxonomy.CreateTaxonomy 693
Ektron.Services.Taxonomy.DeleteTaxonomy 693
Ektron.Services.Taxonomy.GetTaxonomyIdByPath 694
Ektron.Services.Taxonomy.LoadTaxonomy 694
Ektron.Services.Taxonomy.ReadAllAssignedCategory 694
Ektron.Services.Taxonomy.ReadTaxonomy 695
Ektron.Services.Taxonomy.RemoveTaxonomyFolder 695
Ektron.Services.Taxonomy.RemoveTaxonomyItem 696
Ektron.Services.Taxonomy.ReOrderTaxonomyItems 696
Ektron.Services.Taxonomy.UpdateTaxonomy 696
Ektron.Services.ThreadedDiscussion 697
Ektron.Services.ThreadedDiscussion.AddBoard 698

Ektron.Services.ThreadedDiscussion.AddForum 698
Ektron.Services.ThreadedDiscussion.AddTopic 699, 700
Ektron.Services.ThreadedDiscussion.DeleteBoard 700
Ektron.Services.ThreadedDiscussion.DeleteForum 701
Ektron.Services.ThreadedDiscussion.DeleteTopic 701
Ektron.Services.ThreadedDiscussion.GetBoardCategories 701
Ektron.Services.ThreadedDiscussion.GetBoardForums 702
Ektron.Services.ThreadedDiscussion.GetDiscussionBoard 702, 703
Ektron.Services.ThreadedDiscussion.GetDiscussionBoardString 703
Ektron.Services.ThreadedDiscussion.GetForum 704
Ektron.Services.ThreadedDiscussion.GetRepliesForTopics 704
Ektron.Services.ThreadedDiscussion.GetTopic 705
Ektron.Services.ThreadedDiscussion.SearchReplies 705
Ektron.Services.User 706
Ektron.Services.User.ActivateUserAccount 708
Ektron.Services.User.AddMembershipUser 709
Ektron.Services.User.AddNewUser 709
Ektron.Services.User.AddRolePermission 710
Ektron.Services.User.AddUser 710
Ektron.Services.User.AddUserGroup 711
Ektron.Services.User.AddUserToGroup 711
Ektron.Services.User.AutoLogInUser 712
Ektron.Services.User.CreateBlankUserObject 712, 713
Ektron.Services.User.DeleteUserByID 713
Ektron.Services.User.DeleteUserByIds 713
Ektron.Services.User.DeleteUserFromGroup 714
Ektron.Services.User.DeleteUserGroup 714
Ektron.Services.User.EditUserCustomProperties 715
Ektron.Services.User.GetActiveUser 715
Ektron.Services.User.GetADMapping 715
Ektron.Services.User.GetAllCmsUsers 716
Ektron.Services.User.GetAllCustomProperties 717
Ektron.Services.User.GetAllUserByCustomProperty 717
Ektron.Services.User.GetAllUserGroups 718
Ektron.Services.User.GetAllUsers 719
Ektron.Services.User.GetGroupsUserIsIn 720
Ektron.Services.User.GetMembershipUser 720
Ektron.Services.User.GetUser 721
Ektron.Services.User.GetUserbyUsername 721
Ektron.Services.User.GetUserCustomProperties 722
Ektron.Services.User.GetUserEmailInfo 722
Ektron.Services.User.GetUserForFolderAdmin 723
Ektron.Services.User.GetUserGroup 723
Ektron.Services.User.GetUserGroupForFolderAdmin 724
Ektron.Services.User.GetUserPermissions 724
Ektron.Services.User.GetUserPreference 725
Ektron.Services.User.GetUsers 725, 726
Ektron.Services.User.GetUsersForTask 726
Ektron.Services.User.GetUsersNotInGroup 727
Ektron.Services.User.IsAdmin 728
Ektron.Services.User.IsARoleMember 729, 730
Ektron.Services.User.IsARoleMemberForContent 731
Ektron.Services.User.IsARoleMemberForFolder 731
Ektron.Services.User.LogInUser 732
Ektron.Services.User.ResetMembershipUserPassword 733
Ektron.Services.User.UnsubscribeUser 733
Ektron.Services.User.UpdateCustomPropertiesItemOrder 734
Ektron.Services.User.UpdateUser 735
Ektron.Services.User.UpDateUserGroup 735
Ektron.Services.User.UpdateUserPreferences 736
Ektron.Services.User.UserObject 736, 737
EktronCatalogPart class 129
- about EktronCatalogPart class 129
- BlogID 130
- BoardID 130
- CalendarDynamicParameter 131
- CollectionDynamicParameter 131
- ContentBlockDynamicParameter 131
- DefaultCalendarID 131
- DefaultCollectionID 132
- DefaultContentID 132
- DefaultFormID 132
- EktronCatalogPart properties 129
- EnableBlogControl 132
- EnableCalendarControl 133
- EnableCollectionControl 133

EnableContentBlockControl 133
EnableDiscussionBoardsControl 133
EnableFormBlockControl 133
EnableListSummaryControl 134
EnableThreadedDiscussionsControl 134
FolderID 134
FormBlockDynamicParameter 134
Title 135
EktronCommunityCatalogPart class 135
about EktronCommunityCatalogPart class 135
EktronCommunityCatalogPart properties 135
EnableBlogs 136
EnableDiscussionBoards 136
EnableForums 136
Title 136
EktronWebPartZone class 137
about EktronWebPartZone class 137
CmsSitePath 137
EktronWebPartZone properties 137
EkWebControl class 138
about EkWebControl class 138
Authenticated 141
CmsSitePath 141
EkWebControl 138
EkWebControl methods 138
EkWebControl properties 140
Error 142
Fill 138
Hide 142
IsLoggedIn 142
Language 142
LanguageID 143
LocalizeString 139
loggedInUserID 143
loggedInUserName 143
RegisterClientScriptInclude 139
RegisterStylesheet 139
SearchHeader 140
SuppressWrapperTags 143
Text 144
ToString 140
Wrap 140
WrapTag 144
EkWebPart class 144
about EkWebPart class 144
EkXsltBase class 145
about EkXsltBase class 145
DisplayXslt 146
EkXsltBase 145
EkXsltBase methods 145
EkXsltBase properties 146
SetDirty 145
XmlDoc 146
XsltDoc 146
ExtensionAuthor class 330
about ExtensionAuthor class 330
Author 331
ExtensionAuthor 331
ExtensionAuthor methods 331
ExtensionAuthor properties 331
ToString 331
ExtensionDescription class 332
about ExtensionDescription class 332
Desc 333
ExtensionDescription 332
ExtensionDescription methods 332
ExtensionDescription properties 333
ToString 332
ExtensionEvent class 333
about ExtensionEvent class 333
Content 347
ExtensionEvent methods 333
ExtensionEvent properties 347
Folder 347
Form 348
GetActiveDirectoryUserProxy 335
GetAssetProxy 335
GetBlogProxy 336

GetBusinessRulesProxy 336
GetCalendarEventTypeProxy 336
GetCalendarProxy 336
GetContentProxy 337
GetContentRatingProxy 337
GetCustomFieldsProxy 337
GetFolderProxy 337
GetFontProxy 337
GetFormProxy 338
GetLibraryProxy 338
GetMetadataProxy 338
GetObjectStateAsXml 338
GetPermissionsProxy 339
GetSearchManagerProxy 339
GetSiteMapProxy 339
GetSiteProxy 339
GetTaskCategoryProxy 340
GetTaskCategoryTypeProxy 340
GetTaskProxy 340
GetThreadedDiscussionProxy 340
GetUserProxy 340
OnAfterContentAdd 341
OnAfterContentDelete 341
OnAfterContentEdit 341
OnAfterFolderAdd 341
OnAfterFolderDelete 342
OnAfterFolderEdit 342
OnAfterFormSubmit 342
OnAfterPublish 342
OnAfterReviewRejected 343
OnAfterUserAdd 343
OnAfterUserDelete 343
OnAfterUserEdit 343
OnAfterUserLogin 344
OnBeforeContentAdd 344
OnBeforeContentDelete 344
OnBeforeContentEdit 344
OnBeforeFolderAdd 344
OnBeforeFolderDelete 345
OnBeforeFolderEdit 345
OnBeforeFormSubmit 345
OnBeforePublish 345
OnBeforeReviewRejected 346
OnBeforeUserAdd 346
OnBeforeUserDelete 346
OnBeforeUserEdit 346
OnBeforeUserLogin 347
Site 348
User 348

F

FlexMenu class 147
about FlexMenu class 147
AutoCollapseBranches 148
CacheInterval 148
DisplayXslt 149
EnableAjax 149
EnableMouseOverPopUp 149
EnableSmartOpen 150
EnableSubmenuLinkPathMatching 150
EnableSubmenuTemplateMatching 150
Fill 147
FlexMenu methods 147
FlexMenu properties 147
MasterControlId 151
MenuDepth 151
SetDirty 147
StartCollapsed 151
StartLevel 152
Stylesheet 152
XmlDoc 152
XsltDoc 153
Folder class 383, 644
about Folder class 383, 644
AddCommunityFolder 384
AddFolder 385, 645
CreateFolder 386, 645
DeleteFolder 645

DeleteFolderById 386
Folder methods 384, 644
GetChildFolders 387, 646
GetFolder 389, 390, 646
GetFolder method 389
GetFolderDataWithPermission 390
GetFolderId 390, 391
GetFolderID 647
GetFolderId method 390
GetFolderID method 646
GetFolderInfo 391
GetPath 391, 647
RenameFolder 392, 648
UpdateFolder 392, 648
FolderBreadcrumb class 153
about FolderBreadcrumb class 153
DefaultContentID 154
DefaultFolderID 155
DisplayStyle 155
DynamicParameter 155
Fill 153
FolderBreadcrumb methods 153
FolderBreadcrumb properties 154
LinkTarget 156
Mode 156
separator 156
FolderData 755
FolderData class 755
about FolderData class 755
ApprovalMethod 757
CategoryRequired 758
ChildFolders 758
Description 758
DomainProduction 759
DomainStaging 759
FolderData properties 755
FolderTaxonomy 759
FolderType 760
HasChildren 760
Id 760
Inherited 761
InheritedFrom 761
IsCommunityFolder 761
IsDomainFolder 762
MetaInherited 762
MetaInheritedFrom 762
Name 762
NameWithPath 763
ParentId 763
Permissions 763
PrivateContent 764
PublishHtmlActive 764
PublishPdfActive 764
PublishPdfEnabled 765
ReplicationMethod 765
SitemapInherited 765
SitemapInheritedFrom 766
SitemapPath 766
StyleSheet 766
StyleSheetInherited 767
TaxonomyInherited 767
TaxonomyInheritedFrom 767
TemplateFileName 768
TemplateId 768
TemplateInherited 768
TotalContent 768
UpdatedContent 769
XmlConfiguration 769
XmlInherited 769
XmlInheritedFrom 770
FolderData.ApprovalMethod 757
FolderData.CategoryRequired 758
FolderData.ChildFolders 758
FolderData.Description 758
FolderData.DomainProduction 759
FolderData.DomainStaging 759
FolderData.FolderTaxonomy 759
FolderData.FolderType 760

FolderData.HasChildren 760
FolderData.Id 760
FolderData.Inherited 761
FolderData.InheritedFrom 761
FolderData.IsCommunityFolder 761
FolderData.IsDomainFolder 762
FolderData.MetaInherited 762
FolderData.MetaInheritedFrom 762
FolderData.Name 762
FolderData.NameWithPath 763
FolderData.ParentId 763
FolderData.Permissions 763
FolderData.PrivateContent 764
FolderData.PublishHtmlActive 764
FolderData.PublishPdfActive 764
FolderData.PublishPdfEnabled 765
FolderData.ReplicationMethod 765
FolderData.SitemapInherited 765
FolderData.SitemapInheritedFrom 766
FolderData.SitemapPath 766
FolderData.StyleSheet 766
FolderData.StyleSheetInherited 767
FolderData.TaxonomyInherited 767
FolderData.TaxonomyInheritedFrom 767
FolderData.TemplateFileName 768
FolderData.TemplateId 768
FolderData.TemplateInherited 768
FolderData.TotalContent 768
FolderData.UpdatedContent 769
FolderData.XmlConfiguration 769
FolderData.XmlInherited 769
FolderData.XmlInheritedFrom 770
FolderInfo class 348
- about FolderInfo class 348
- Description 349
- ErrorMessage 349
- FolderInfo properties 349
- ID 349
- Name 349
- ParentID 350
- ReplicationState 350
- StyleSheet 350
- Type 350

FolderRequest 770
FolderRequest class 770
- about FolderRequest class 770
- BreakInheritButton 772
- CategoryRequired 772
- ContentSubAssignments 773
- DomainProduction 773
- DomainStaging 773
- EnableReplication 773
- FileDirectory 774
- FolderCfldAssignments 774
- FolderDescription 774
- FolderId 775
- FolderName 775
- FolderRequest properties 770
- FolderType 775
- ImageDirectory 776
- IsDomainFolder 776
- MetaInherited 776
- MetaInheritedFrom 777
- ParentId 777
- PublishActive 777
- SiteMapPath 777
- SiteMapPathInherit 778
- StyleSheet 778
- SubscriptionProperties 778
- suppressNotification 779
- TaxonomyIdList 779
- TaxonomyInherited 779
- TaxonomyInheritedFrom 780
- TemplateFileName 780
- XmlConfiguration 780
- XmlInherited 781

FolderRequest.BreakInheritButton 772
FolderRequest.CategoryRequired 772

FolderRequest.ContentSubAssignments 773
FolderRequest.DomainProduction 773
FolderRequest.DomainStaging 773
FolderRequest.EnableReplication 773
FolderRequest.FileDirectory 774
FolderRequest.FolderCfldAssignments 774
FolderRequest.FolderDescription 774
FolderRequest.FolderId 775
FolderRequest.FolderName 775
FolderRequest.FolderType 775
FolderRequest.ImageDirectory 776
FolderRequest.IsDomainFolder 776
FolderRequest.MetaInherited 776
FolderRequest.MetaInheritedFrom 777
FolderRequest.ParentId 777
FolderRequest.PublishActive 777
FolderRequest.SiteMapPath 777
FolderRequest.SiteMapPathInherit 778
FolderRequest.StyleSheet 778
FolderRequest.SubscriptionProperties 778
FolderRequest.suppressNotification 779
FolderRequest.TaxonomyIdList 779
FolderRequest.TaxonomyInherited 779
FolderRequest.TaxonomyInheritedFrom 780
FolderRequest.TemplateFileName 780
FolderRequest.XmlConfiguration 780
FolderRequest.XmlInherited 781
Font class 392, 648
about Font class 392, 648
AddFont 393, 649
DeleteFont 393, 649
Font methods 393, 649
GetAllFonts 393, 650
GetFont 394, 650
GetFontConfigList 394, 651
UpdateFont 394, 651
Form class 493, 651
about Form class 493, 651
AssignFormTask 494, 652
EmailFormFieldData 494, 653
Form methods 493, 652
GetAllForms 495, 653, 654
GetAllForms method 494, 653
GetForm 495, 654
GetFormDataHistogram 654
GetFormFieldData 496, 655, 656
GetFormFieldData method 496, 655
GetFormFieldList 497, 656
GetFormFieldQuestions 497, 657
GetFormTitle 497, 657
SubmitFormFieldData 498, 657
UpdateFormFieldData 498, 658
FormBlock class 156
about FormBlock class 156
AddValidation 160
CacheInterval 160
CopyTo 157
Count 161
Current 161
DefaultFormID 161
DynamicParameter 162
EkItem 162
Fields 162
Fill 158
FormBlock methods 157
FormBlock properties 160
FormName 162
GetEnumerator 158
GetView 158
GetViewNames 158
IncludeTags 163
IsSynchronized 163
MoveNext 158
Reset 159
StartingFolderID 163
SyncRoot 164
this 159
ToString 159

Wrap 159
FormData 781
FormData class 781
about FormData class 781
Autofill 782
Description 782
FormData properties 781
MailCc 783
MailFrom 783
MailPreamble 783
MailSubject 783
MailTo 784
PollRevision 784
SendXmlPacket 784
StoreDataTo 785
Validation 785
FormData.Autofill 782
FormData.Description 782
FormData.MailCc 783
FormData.MailFrom 783
FormData.MailPreamble 783
FormData.MailSubject 783
FormData.MailTo 784
FormData.PollRevision 784
FormData.SendXmlPacket 784
FormData.StoreDataTo 785
FormData.Validation 785
FormField class 350
about FormField class 350
FieldName 351
FieldValue 351
FormField 351
FormField properties 351
FormFieldDataItem 785
FormFieldDataItem class 785
about FormFieldDataItem class 785
DataValue 786
FieldName 786
FormFieldDataItem properties 786
FormFieldDataItem.DataValue 786
FormFieldDataItem.FieldName 786
FormFieldDatalist 786
FormFieldDatalist class 786
about FormFieldDatalist class 786
CaptionXPath 787
DatalistItems 788
DataNamespaces 788
DataSelect 788
DataSrc 788
FormFieldDatalist properties 787
Language 788
Name 789
ValueXPath 789
FormFieldDatalist.CaptionXPath 787
FormFieldDatalist.DatalistItems 788
FormFieldDatalist.DataNamespaces 788
FormFieldDatalist.DataSelect 788
FormFieldDatalist.DataSrc 788
FormFieldDatalist.Language 788
FormFieldDatalist.Name 789
FormFieldDatalist.ValueXPath 789
FormFieldDatalistItem 789
FormFieldDatalistItem class 789
about FormFieldDatalistItem class 789
DataValue 790
DisplayValue 790
FormFieldDatalistItem properties 790
IsDefault 791
IsEnabled 791
FormFieldDatalistItem.DataValue 790
FormFieldDatalistItem.DisplayValue 790
FormFieldDatalistItem.IsDefault 791
FormFieldDatalistItem.IsEnabled 791
FormFieldDefinition 791
FormFieldDefinition class 791
about FormFieldDefinition class 791
Basetype 792
ContentType 792

DatalistName 793
Datatype 793
DisplayName 793
FieldName 794
FormFieldDefinition properties 792
Language 794
XPath 794
FormFieldDefinition.Basetype 792
FormFieldDefinition.ContentType 792
FormFieldDefinition.DatalistName 793
FormFieldDefinition.Datatype 793
FormFieldDefinition.DisplayName 793
FormFieldDefinition.FieldName 794
FormFieldDefinition.Language 794
FormFieldDefinition.XPath 794
FormFieldItem class 164
about FormFieldItem class 164
Basetype 164
ContentType 165
DatalistItems 165
Datatype 165
DisplayName 166
FieldName 166
FormFieldItem properties 164
Language 166
Value 167
FormFieldList 794
FormFieldList class 794
about FormFieldList class 794
Datalists 795
Fields 795
FormFieldList properties 795
FormFieldList.Datalists 795
FormFieldList.Fields 795
FormInfo class 351
about FormInfo class 351
ErrorMessage 352
FormDescription 353
FormFields 353
FormInfo methods 352
FormInfo properties 352
FormTitle 353
this 352
FormRequestData 796
FormRequestData class 796
about FormRequestData class 796
AutoFill 797
FieldList 797
FormRequestData properties 796
Id 797
IsPoll 797
Language 798
PostBackData 798
PreFillFielddata 798
TagRequired 799
FormRequestData.AutoFill 797
FormRequestData.FieldList 797
FormRequestData.Id 797
FormRequestData.IsPoll 797
FormRequestData.Language 798
FormRequestData.PostBackData 798
FormRequestData.PreFillFielddata 798
FormRequestData.TagRequired 799
FormSubmittedData 799
FormSubmittedData class 799
about FormSubmittedData class 799
DataItems 800
DateSubmitted 800
FormDataID 800
FormSubmittedData properties 799
UserId 800
UserInfo 801
FormSubmittedData.DataItems 800
FormSubmittedData.DateSubmitted 800
FormSubmittedData.FormDataID 800
FormSubmittedData.UserId 800
FormSubmittedData.UserInfo 801
Forum class 167

about Forum class 167
BoardID 169
CacheInterval 169
DynamicParameter 169
ExcludeList 170
Fill 167
Forum methods 167
Forum properties 168
HideUserProfile 170
JavascriptEditorHeight 170
JavascriptEditorHTMLMode 171
JavascriptEditorToolbar 171
JavascriptEditorWidth 172
Theme 172
ToString 168
Wrap 168

H

How Do I... 2
HtmlEditDisplay class 172
about HtmlEditDisplay class 172
BaseJavaScript 173
BeginJavaScript 174
EndJavaScript 174
HtmlEditDisplay methods 173
MakeFieldNameFromId 174
RenderCMSFields 174
SelectedJavaScript 175
HtmlEditor class 175
about HtmlEditor class 175
Archived 179
Authenticated 180
CmsSitePath 180
ContentAction 181
ContentID 181
ContentTitle 182
ContentType 182
EditorReadyCall 183
EnableUpload 183
ExecuteContentAction 177
HtmlEditor 176
HtmlEditor methods 176
HtmlEditor properties 178
IncludeContentInfo 184
IsLoggedIn 184
Language 185
loggedInUserID 185
loggedInUserName 186
Path 186
PreferredType 186
SafeEncode 187
ShowCMSToolbar 188
ShowTitle 188
StyleSheet 188
Text 189
ToolbarLevel 189
ToolbarResetCall 190
ToString 177
TranslationFile 191
WorkareaMode 178

I

ImageControl class 191
about ImageControl class 191
DefaultImageID 192
DynamicParameter 193
Fill 192
FolderID 193
ImageControl methods 192
ImageControl properties 192
Wrap 192
IndexSearch class 194
about IndexSearch class 194
AddParm 195, 196
AddParm method 195
ButtonImgSrc 200
ButtonText 200
ContentParameter 201

Copyright (c) Ektron Inc. 2007

CopyTo 196
Count 201
Current 201
EkItems 201
EmptyResultMsg 202
EncodeContentHtml 202
Fill 196
FolderId 202
FormName 203
GetEnumerator 196
GetSearchParmDataType 196
GetSearchParmSearchType 197
GetXPathValues 197
HitsMarker 203
IndexSearch 194
IndexSearch methods 195
IndexSearch properties 199
IndexSearch structures 194
IsSynchronized 203
Labelafter 203
Labelbefore 204
LabelBetween 204
LabelContains 204
LabeldateBetween 205
LabelEqual 205
LabelExactPhrase 205
LabelGreaterThan 205
Labelinputdate 206
Labelinputnumber 206
LabelLessThan 206
Labelon 207
LabelSelect 207
LinkTarget 207
MaxResults 208
MoveNext 197
OrderBy 208
OrderByDirection 208
Recursive 209
Reset 198
Search 198
Search method 198
SearchParmXML 209
ShowSearchBoxAlways 209
SyncRoot 210
this 198
Weighted 210
Wrap 198
XmlConfigId 210
XmlDoc 210

L

LanguageAPI class 211
about LanguageAPI class 211
CurrentLanguage 213
CurrentLanguageID 213
DefaultLanguage 213
DefaultLanguageID 214
Fill 211
GetLanguage 211
GetLanguageID 212
IsValid 212
IsValid method 212
LangaugeIdList 214
LangaugeTitleList 214
LanguageAPI 211
LanguageAPI methods 211
LanguageAPI properties 212
MultiLanguageEnabled 214
OverrideDefault 214
SiteLanguage 215
SiteLanguageID 215
LanguageSelect class 215
about LanguageSelect class 215
Fill 216
LabelName 217
LanguageSelect methods 216
LanguageSelect properties 216
Text 217

Wrap 216
Library class 395, 658
about Library class 395, 658
AddLibraryItem 396
AddLoadBalanceItem 396, 659
DeleteLibraryItem 396, 660
DeleteLibraryItemById 396, 660
DeleteLoadBalanceItem 397, 660
GetAllChildLibItems 397, 661
GetAllLoadBalancePaths 399, 400, 662
GetAllLoadBalancePaths method 399, 661
GetLibraryItem 400, 663
GetLibraryItem method 400, 662
GetLibrarySettings 401, 664
GetLibraryTypes 401, 664
GetLoadBalancePath 401, 665
InternalLibrarySearch 402, 665
Library methods 395, 659
UpdateLibrary 402, 665
UpdateLibrarySettings 402, 666
UpdateLoadBalanceSettings 403, 666
UpdateQlinkTemplates 403, 666
List of Developer Resources 4
ListSummary class 217
about ListSummary class 217
AddText 222
CacheInterval 222
ContentParameter 223
ContentType 223
CopyTo 218
Count 224
Current 224
Direction 224
EkItems 224
EnablePaging 225
Fill 218
FolderDescription 225
FolderID 225
FolderName 225
FolderPath 226
GetEnumerator 218
GetHtml 226
GetRssFeed 219
GetView 219
GetViewNames 219
IncludeIcons 226
IsSynchronized 227
LinkTarget 227
ListSummary methods 218
ListSummary properties 221
MarkupLanguage 227
MaxResults 227
MoveNext 219
OrderKey 228
Random 228
Recursive 228
Reset 220
SuppressWrapperTags 229
SyncRoot 229
this 220
ToString 220
Wrap 220
XmlDoc 229
Localization class 403
about Localization class 403
GetAddViewLanguage 404
GetAddViewLanguageForAllContent 405
GetAddViewLanguageForMenus 405
GetContentItems 405
GetFile 406
GetFiles 406
GetFilesInZipFile 406
GetFlagUrl 407, 408
GetFlagUrl method 407
GetInterchangeFile 408
GetInterchangeFiles 408
GetInterchangeFilesInZipFile 409
GetJobs 410

GetJobs method 409
GetLocalizationUrl 410
GetTranslationUploadDirectory 410
Localization methods 403
StartExportContentForTranslation 411
StartExportFolderForTranslation 411
StartExportMenusForTranslation 412
StartImportTranslation 412
Login class 229
about Login class 229
AutoAddType 230
AutoLogin 231
Fill 230
Login methods 230
Login properties 230
OnlyAllowMemberLogin 231
Password 231
PromptLogout 231
SuppressHelpButton 231
Text 232
UserName 232

M

Map class 232
about Map class 232
Address 234
ContentId 234
CustomIcon 235
DateSearch 235
DisplayTextResult 235
DistanceUnit 236
EnableSearchResult 236
Fill 233
FolderId 236
GeoControl 237
GetCallbackResult 233
Latitude 237
Longitude 237
Map methods 232
Map properties 233
MapCategory 237
MapProvider 238
MapStyle 238
MarkupLanguage 238
MinZoomLevel 238
PageSize 238
RaiseCallbackEvent 233
Recursive 239
SelectedCategoryLogicalType 239
StartZoomLevel 239
TypeControl 239
ZoomControl 240
Membership class 240
about Membership class 240
BoardID 242
DisplayMode 242
Fill 241
Membership methods 240
Membership properties 241
RedirectFailedURL 242
RedirectSuccessURL 243
RegisterButtonImg 243
RegisterButtonText 243
ResetButtonImg 243
ResetButtonText 243
SaveButtonImg 244
SaveButtonText 244
ShowExtended 244
ShowTerms 244
Text 245
UserExistsMessage 245
UserID 245
UserSuccessMessage 245
UserUpdateSuccessMessage 246
Menu class 246
about Menu class 246
CacheInterval 247
DefaultMenuID 248

DisplayXslt 248
DynamicParameter 248
Fill 246
Menu methods 246
Menu properties 247
Stylesheet 249
SuppressAddEdit 249
Wrap 247
Metadata class 412, 667
about Metadata class 412, 667
AddMetaDataType 413, 414, 667
AddMetaDataType method 413
DeleteMetadataType 415, 668
GetContentMetadataList 416, 417, 668, 669
GetContentMetadataList method 416, 668
GetIntegratedSearchFolder 418
GetIntegratedSearchFolders 418
GetMetadataType 418, 669
GetMetaDataTypes 419, 670
Metadata methods 413, 667
UpdateIntegratedSearchFolder 419
UpdateMetadataType 419, 670
MetaData class 249
about MetaData class 249
DefaultContentID 250
DefaultItemList 251
DynamicParameter 251
Fill 250
GenerateDublinCore 251
MetaData methods 249
MetaData properties 250
MetaDataList class 252
about MetaDataList class 252
CacheInterval 255
ContentType 256
CopyTo 253
Count 256
Current 256
EkItems 256
EnablePaging 256
ExactPhrase 257
Fill 253
FolderID 257
GetEnumerator 253
GetHtml 258
IncludeIcons 258
IsSynchronized 258
KeyWordName 258
KeyWordValue 259
KeyWordValueMatchAll 259
KeyWordValueSeparator 259
LinkTarget 260
MarkupLanguage 260
MaxNumber 260
MetaDataList methods 252
MetaDataList properties 254
MoveNext 253
OrderBy 261
Recursive 261
Reset 253
SortOrder 261
SuppressWrapperTags 262
SyncRoot 262
this 254
Wrap 254
XmlDoc 262

P

Permissions class 419, 671
about Permissions class 419, 671
AddItemPermission 420, 671
DeleteItemPermission 420, 672
DisableContentInheritance 421, 672
DisableFolderInheritance 421, 673
EnableContentInheritance 421, 673
EnableFolderInheritance 422, 674
GetUserPermissions 422, 423, 674, 675
GetUserPermissions method 422, 674

LoadPermissions 423, 675
Permissions methods 420, 671
UpdateItemPermission 424, 676
PersonalizationManager class 262
about PersonalizationManager class 262
AddContentExitText 264
AddContentText 264
CurrentUserButtonText 264
Display 265
Fill 263
HeaderBackColor 265
HeaderText 265
PersonalizationManager methods 263
PersonalizationManager properties 263
PersonalizeExitText 266
PersonalizeScopeText 266
PersonalizeStartText 266
PublicUserButtonText 267
ResetLinkText 267
ResetLinkToolTip 267
Plug-in Extension API 325
Poll class 268
about Poll class 268
EnableAjax 269
Fill 268
Poll methods 268
Poll properties 269
PollID 269
PostHistory class 270
about PostHistory class 270
BoardID 271
CacheInterval 272
Fill 270
MaxNumber 272
PostHistory methods 270
PostHistory properties 271
ToString 270
URLPath 272
UserID 272
Wrap 271

R

RequestInfo class 353
about RequestInfo class 353
ReviewInfo class 353
about ReviewInfo class 353
AverageRating 354
ContentHtml 355
ContentID 355
ContentLanguage 355
ContentTitle 355
ErrorMessage 355
ID 355
NumberofRatings 356
Rating 356
RatingDate 356
ReviewInfo properties 354
State 356
UserAvatar 356
UserComments 356
UserDisplayName 357
UserFirstName 357
UserID 357
UserLastName 357
UserName 357
VisitorID 357
RssAggregator class 273
about RssAggregator class 273
CopyTo 274
Count 276
Current 276
EkItems 276
Fill 274
GetEnumerator 274
GetView 274
GetViewNames 274
IsSynchronized 276
LinkTarget 277

MaxResults 277
MoveNext 275
Reset 275
RssAggregator methods 273
RssAggregator properties 275
SyncRoot 277
this 275
URL 278
RssItemBase class 278
about RssItemBase class 278
DatePosted 278
Description 279
Link 279
RssItemBase fields 278
Title 279

S

Search class 279
about Search class 279
ButtonImgSrc 284
ButtonText 285
ContentFieldsetLegend 285
CopyTo 280
Count 285
Current 286
Display 286
EkItems 286
EmptyResultMsg 286
EnableAdvancedLink 287
EnableAjax 287
EnableArchived 287
EnableArchivedBtn 288
EnableContent 288
EnableContentBtn 288
EnableContentFieldset 288
EnableDMSAssets 289
EnableDMSAssetsBtn 289
EnableFolderFieldset 289
EnableForms 290
EnableFormsBtn 290
EnableLibFiles 290
EnableLibFilesBtn 291
EnableLibHyperlinks 291
EnableLibHyperlinksBtn 291
EnableLibImages 292
EnableLibImagesBtn 292
EnableLibraryFieldset 292
Fill 280
FolderFieldsetLegend 292
FolderID 293
FolderPath 293
GetEnumerator 281
IncludeIcons 293
IsSynchronized 294
LibraryFieldsetLegend 294
LinkTarget 294
LoadSearch 281
MaxCharacters 295
MaxResults 295
MaxTeaserLength 295
MoveNext 281
OrderBy 296
OrderDirection 296
PostResultsAt 296
Recursive 297
RemoveTeaserHtml 297
Reset 281
ResultTagId 297
Search 280
Search methods 280
Search properties 282
SearchDisplayUI 281
SearchText 298
SearchType 298
ShowDate 298
ShowExtendedSearch 298
ShowSearchBoxAlways 299
ShowSearchOptions 299

Stylesheet 299
SyncRoot 300
TargetFolderDescriptionList 300
TextBoxSize 300
this 282
Wrap 282
XmlDoc 301
SearchManager class 515, 676
about SearchManager class 515, 676
Execute 516
Execute method 515
ExecuteSearch 677, 678
ExecuteSearch method 677
Search 517, 518, 678, 679
Search method 517, 678
SearchForCategories 519, 680
SearchManager methods 515, 677
SearchTable 519, 680
Server Controls API 7
Site class 424, 681
about Site class 424, 681
BreakTaskPermissionInherit 425, 682
DeleteCookie 425
GetAllActiveLanguages 425, 682
GetAllLanguages 426, 683
GetAllLanguages method 426, 683
GetClientDayNames 426, 684
GetClientMonthNames 427, 684
GetCookieValue 427, 684
GetLanguage 427, 685
GetLanguages 428, 685
GetSiteVariables 428, 685
GetTaskPermission 428, 686
IsTaskPermissionInherited 429, 686
SetCookieValue 429, 687
SetTaskPermission 429, 687
Site methods 424, 681
UpdateSiteEnabledLanguages 429, 688
UpdateSiteVariables 430
SiteInfo class 358
about SiteInfo class 358
AppPath 358
IsSSL 358
Name 359
PhysicalSitePath 359
Port 359
SiteInfo properties 358
Url 359
Sitemap class 301
about Sitemap class 301
CacheInterval 302
ClassName 302
DisplayType 303
DisplayXslt 303
Fill 301
FlatTopLevel 304
FolderID 304
MaxLevel 304
Sitemap methods 301
Sitemap properties 302
StartingLevel 305
Stylesheet 305
XmlDoc 305
SiteMap class 430, 688
about SiteMap class 430, 688
GetFolderBreadcrumbPath 431, 689
GetWebsiteMap 432, 689
RenderFolderBreadcrumbHtml 434, 690
RenderWebsiteMap 434, 690
SiteMap methods 430, 688
SmartMenu class 305
about SmartMenu class 305
AutoCollapseBranches 306
DisplayXslt 307
Enable508Compliance 307
Enable508NavigationHeadings 307
EnableMouseOverPopUp 308
EnableSmartOpen 308

EnableSubmenuLinkPathMatching 308
EnableSubmenuTemplateMatching 309
LaunchLinksInNewWindow 309
ShowRootFolder 310
SingleEditButton 310
SmartMenu properties 306
StartCollapsed 310
StartWithRootCollapsed 311
Stylesheet 311
Subscription class 435
about Subscription class 435
AddEmailFrom 436
AddSubscription 436
AddSubscriptionMessage 436
DeleteEmailFrom 437
DeleteSubscription 437
DeleteSubscriptionMessage 437
GetAllActiveSubscriptions 437
GetAllEmailFrom 438
GetAllSubscriptionMessages 438
GetAllSubscriptions 438
GetFolderInheritedFrom 438
GetSubscription 439
GetSubscriptionMessage 439
GetSubscriptionMessagesForType 439
GetSubscriptionPropertiesForContent 439
GetSubscriptionPropertiesForFolder 440
GetSubscriptionsForContent 440
GetSubscriptionsForFolder 440
Subscription methods 435
UpdateEmailFrom 441
UpdateSubscription 441
UpdateSubscriptionMessage 441

T

Task class 520, 576
about Task class 520, 576
AddTask 521, 577
AssignTaskToUser 521, 522, 578
AssignTaskToUser method 521, 578
AssignTaskToUsergroup 522, 523, 579
AssignTaskToUsergroup method 522, 579
DeleteTask 523, 524, 580
DeleteTask method 523, 580
DeleteTasks 524, 525, 581, 582
DeleteTasks method 524, 581
GetTask 525, 582
GetTasks 525, 526, 582
GetTasks method 525
GetTasksAssignedByUser 526, 583
GetTasksAssignedByUser method 526, 583
GetTasksAssignedToUser 527, 584
GetTasksAssignedToUser method 527, 584
GetTasksAssignedToUsergroup 528, 585, 586
GetTasksAssignedToUsergroup method 528, 585
GetTaskState 529, 586
SetStateForTask 529, 586
Task methods 520, 577
UpdateTask 529, 587
TaskCategory class 530, 587
about TaskCategory class 530, 587
AddTaskCategory 530, 588
DeleteTaskCategory 531, 588
GetAllCategories 531, 589
GetTaskCategory 531, 589
TaskCategory methods 530, 588
UpdateTaskCategory 531, 590
TaskCategoryType class 532, 590
about TaskCategoryType class 532, 590
AddTaskType 532, 591
DeleteTaskType 533, 591
GetTaskType 533, 592
TaskCategoryType methods 532, 590
UpdateTaskType 533, 592
Taxonomy class 498, 691
about Taxonomy class 498, 691
AddTaxonomyItem 499, 500, 692
AddTaxonomyItem method 499, 692

AddTaxonomySyncFolder 500, 692
CreateTaxonomy 500, 693
DeleteTaxonomy 500, 693
GetTaxonomyIdByPath 501, 694
ImportTaxonomy 501
LoadTaxonomy 501, 694
ReadAllAssignedCategory 502, 694
ReadTaxonomy 502, 695
RemoveTaxonomyFolder 502, 695
RemoveTaxonomyItem 502, 696
ReOrderTaxonomyItems 503, 696
Taxonomy methods 499, 691
UpdateTaxonomy 503, 696
ThreadedDiscussion class 503, 697
about ThreadedDiscussion class 503, 697
AddBoard 504, 698
AddForum 505, 698
AddTopic 507, 508, 699, 700
AddTopic method 507, 699
DeleteBoard 508, 700
DeleteForum 508, 701
DeleteTopic 508, 701
GetBoardCategories 509, 701
GetBoardForums 509, 702
GetDiscussionBoard 510, 702, 703
GetDiscussionBoard method 509, 702
GetDiscussionBoardString 510, 703
GetForum 511, 704
GetRepliesForTopics 512, 704
GetTopic 513, 705
SearchReplies 513, 705
ThreadedDiscussion methods 504, 697

U

User class 542, 706
about User class 542, 706
ActivateUserAccount 544, 708
ActivateUserAccount method 544, 708
AddMembershipUser 545, 709
AddNewUser 547, 709
AddRolePermission 547, 710
AddUser 548, 710
AddUser method 548
AddUserGroup 548, 711
AddUserToGroup 549, 711
AutoLogInUser 549, 712
CreateBlankUserObject 550, 712, 713
CreateBlankUserObject method 549, 712
DeleteUserByID 550, 713
DeleteUserByIds 550, 713
DeleteUserFromGroup 551, 714
DeleteUserGroup 551, 714
EditUserCustomProperties 551, 715
GetActiveUser 552, 715
GetADMapping 552, 715
GetAllCmsUsers 552, 716
GetAllCustomProperties 553, 717
GetAllUserByCustomProperty 553, 717
GetAllUserGroups 554, 718
GetAllUserGroups method 553, 717
GetAllUsers 555, 719
GetAllUsers method 554, 719
GetGroupsUserIsIn 555, 720
GetMembershipUser 556, 720
GetUser 556, 721
GetUserbyUsername 721
GetUserByUsername 556
GetUserCustomProperties 558, 722
GetUserEmailInfo 558, 722
GetUserForFolderAdmin 559, 723
GetUserGroup 559, 723
GetUserGroupByName 559
GetUserGroupForFolderAdmin 559, 724
GetUserPermissions 560, 724
GetUserPreference 560, 725
GetUsers 561, 725, 726
GetUsers method 561, 725
GetUsersForTask 562, 726

GetUsersNotInGroup 562, 727
IsAdmin 563, 728
IsARoleMember 564, 565, 729, 730
IsARoleMember method 563, 728
IsARoleMemberForContent 566, 731
IsARoleMemberForContent_FolderUserAdmin 566
IsARoleMemberForFolder 566, 731
IsARoleMemberForFolder_FolderUserAdmin 567
IsARoleMemberForFolder_MetadataAdmin 567
LogInUser 568, 732
ReadCustomProperties 570
ResetMembershipUserPassword 570, 733
ResetMembershipUserPassword method 570, 732
SetAuthenticationCookie 571
UnsubscribeUser 571, 733
UpdateCustomPropertiesItemOrder 571, 734
UpdateUser 572, 735
UpdateUser method 572, 734
UpDateUserGroup 573, 735
UpDateUserGroup method 572
UpdateUserPreferences 573, 736
User methods 542, 706
User properties 574
UserIdAuthenticated 575
UserObject 574, 736, 737
UserObject method 573, 736
UserData 801
UserData class 801
about UserData class 801
AcceptedTerms 802
AdUserName 802
Avatar 803
DateCreated 803
DisplayName 803
DisplayUserName 803
Email 804
FirstName 804
GroupId 804
GroupName 805
Id 805
IsMemberShip 805
LanguageId 805
LanguageName 806
LastLoginDate 806
LastName 806
LoginAttempts 807
Password 807
Rank 807
Signature 807
UserData properties 801
Username 808
UserPreference 808
UserData.AcceptedTerms 802
UserData.AdUserName 802
UserData.Avatar 803
UserData.DateCreated 803
UserData.DisplayName 803
UserData.DisplayUserName 803
UserData.Email 804
UserData.FirstName 804
UserData.GroupId 804
UserData.GroupName 805
UserData.Id 805
UserData.IsMemberShip 805
UserData.LanguageId 805
UserData.LanguageName 806
UserData.LastLoginDate 806
UserData.LastName 806
UserData.LoginAttempts 807
UserData.Password 807
UserData.Rank 807
UserData.Signature 807
UserData.Username 808
UserData.UserPreference 808
UserInfo class 359
about UserInfo class 359
DisplayName 360
Editor 360

EmailAddress 360
ErrorMessage 361
FirstName 361
ID 361
Language 361
LastName 361
Notifications 361
Type 362
UserInfo properties 360
UserName 362
Utilities class 441
about Utilities class 441
LoadPermissions 442
Utilities methods 442

W

Web Services API 575
WebSearch class 311
about WebSearch class 311
ButtonImgSrc 315
ButtonText 316
CustomSearch 316
DisplayXslt 316
EnableAdvancedLink 317
EndCount 317
Fill 312
FolderID 317
LinkTarget 318
LoadSearch 312
MarkupLanguage 318
MaxCharacters 318
MaxTeaserLength 319
OrderBy 319
OrderDirection 319
Recursive 320
RemoveTeaserHtml 320
ResultCount 320
ResultDataTable 321
ResultsPageSize 321
ResultTagId 321
ResultXml 322
SearchDisplayUI 313
SearchFor 322
SearchText 322
ShowSearchBoxAlways 323
ShowSearchOptions 323
StartCount 324
Stylesheet 324
TextBoxSize 324
TimeTaken 324
WebSearch 312
WebSearch methods 312
WebSearch properties 313
Wrap 313
XmlDoc 325
Welcome 1

Copyright (c) Ektron Inc. 2007