



Ektron CMS400.NET Developer Reference Manual

Version 4.8, Revision 1

Ektron® Inc.
5 Northern Blvd., Suite 6
Amherst, NH 03031
Tel: +1 603-594-0249
Fax: +1 603-594-0258
Email: sales@ektron.com
<http://www.ektron.com>

Copyright 2005 Ektron®, Inc. All rights reserved.

EKTRON is a registered trademark of Ektron, Inc.

Version 4.8, Revision 1, April 8, 2005

For updated versions of this manual, please visit:

<http://www.ektron.com/cms400-web-cms.aspx?id=1701#documentation>

EKTRON, INC. SOFTWARE LICENSE AGREEMENT

**** READ THIS BEFORE LOADING SOFTWARE ****

YOUR RIGHT TO USE THE PRODUCT DELIVERED IS SUBJECT TO THE TERMS AND CONDITIONS SET OUT IN THIS LICENSE AGREEMENT. USING THIS PRODUCT SIGNIFIES YOUR AGREEMENT TO THESE TERMS. IF YOU DO NOT AGREE TO THIS SOFTWARE LICENSE AGREEMENT, DO NOT DOWNLOAD.

CUSTOMER should carefully read the following terms and conditions before using the software program(s) contained herein (the "Software"). Downloading and/or using the Software or copying the Software onto CUSTOMER'S computer hard drive indicates CUSTOMER'S acceptance of these terms and conditions. If CUSTOMER does not agree with the terms of this agreement, CUSTOMER should not download.

Ektron, Inc. ("Ektron") grants, and the CUSTOMER accepts, a nontransferable and nonexclusive License to use the Software on the following terms and conditions:

1. Right to use: The Software is licensed for use only in delivered code form. Each copy of the Software is licensed for use only on a single URL. Each license is valid for the number of seats listed below (the "Basic Package"). Any use of the Software beyond the number of authorized seats contained in the Basic Package without paying additional license fees as provided herein shall cause this license to terminate. Should CUSTOMER wish to add seats beyond the seats licensed in the Basic Package, the CUSTOMER may add seats on a block basis at the then current price for additional seats (see product pages for current price). The Basic Packages are as follows:

Ektron eMPower for ColdFusion -- Licensed for ten seats (10 named users) per URL.

Ektron CMS100 -- Licensed for five seats (5 named users) per URL.

Ektron CMS200 -- Licensed for ten seats (10 named users) per URL.

Ektron CMS300 -- Licensed for ten seats (10 named users) per URL.

Ektron CMS400 -- Licensed for ten seats (10 named users) per URL.

Ektron eWebEditPro -- Licensed for ten seats (10 named users) per URL.

Ektron eWebEditPro+XML — Licensed for ten seats (10 named users) per URL.

Ektron eWebWP — Licensed for unlimited seats per URL.

For purposes of this section, the term "seat" shall mean an individual user provided access to the capabilities of the Software.

The CUSTOMER may not modify, alter, reverse engineer, disassemble, or decompile the Software. This software product is licensed, not sold.

2. Duration: This License shall continue so long as CUSTOMER uses the Software in compliance with this License. Should CUSTOMER breach any of its obligations hereunder, CUSTOMER agrees to return all copies of the Software and this License upon notification and demand by Ektron.

3. Copyright: The Software (including any images, "applets," photographs, animations, video, audio, music and text incorporated into the Software) as well as any accompanying written materials (the "Documentation") is owned by Ektron or its suppliers, is protected by United States copyright laws and international treaties, and contains confidential information and trade secrets. CUSTOMER agrees to protect the confidentiality of the Software and Documentation. CUSTOMER agrees that it will not provide a copy of this Software or Documentation nor divulge any proprietary information of Ektron to any person, other than its employees, without the prior consent of Ektron; CUSTOMER shall use its best efforts to see that any user of the Software licensed hereunder complies with this license.

4. Limited Warranty: Ektron warrants solely that the medium upon which the Software is delivered will be free from defects in material and workmanship under normal, proper and intended usage for a period of three (3) months from the date of receipt. Ektron does not warrant the use of the Software will be uninterrupted or error free, nor that program errors will be corrected. This limited warranty shall not apply to any error or failure resulting from (i) machine error, (ii) Customer's failure to follow operating instructions, (iii) negligence or accident, or (iv) modifications to the Software by any person or entity other than Company. In the event of a breach of warranty, Customer's sole and exclusive remedy, is repair of all or any portion of the Software. If such remedy fails of its essential purpose, Customer's sole remedy and Ektron's maximum liability shall be a refund of the paid purchase price for the defective Products only. This limited warranty is only valid if Ektron receives written notice of breach of warranty within thirty days after the warranty period expires.

5. Limitation of Warranties and Liability: THE SOFTWARE AND DOCUMENTATION ARE SOLD "AS IS" AND WITHOUT ANY WARRANTIES AS TO THE PERFORMANCE, MERCHANTABILITY, DESIGN, OR OPERATION OF THE SOFTWARE. NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED. EXCEPT AS DESCRIBED IN SECTION 4, ALL WARRANTIES EXPRESS AND IMPLIED ARE HEREBY DISCLAIMED.

NEITHER COMPANY NOR ITS SUPPLIERS SHALL BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS OR GOODWILL, LOSS OF DATA OR USE OF DATA, INTERRUPTION OF BUSINESS NOR FOR ANY OTHER INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND UNDER OR ARISING OUT OF, OR IN ANY RELATED TO THIS AGREEMENT, HOWEVER,

CAUSED, WHETHER FOR BREACH OF WARRANTY, BREACH OR REPUDIATION OF CONTRACT, TORT, NEGLIGENCE, OR OTHERWISE, EVEN IF COMPANY OR ITS REPRESENTATIVES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH LOSS.

6. Miscellaneous: This License Agreement, the License granted hereunder, and the Software may not be assigned or in any way transferred without the prior written consent of Ektron. This Agreement and its performance and all claims arising from the relationship between the parties contemplated herein shall be governed by, construed and enforced in accordance with the laws of the State of New Hampshire without regard to conflict of laws principles thereof. The parties agree that any action brought in connection with this Agreement shall be maintained only in a court of competent subject matter jurisdiction located in the State of New Hampshire or in any court to which appeal therefrom may be taken. The parties hereby consent to the exclusive personal jurisdiction of such courts in the State of New Hampshire for all such purposes. The United Nations Convention on Contracts for the International Sale of Goods is specifically excluded from governing this License. If any provision of this License is to be held unenforceable, such holding will not affect the validity of the other provisions hereof. Failure of a party to enforce any provision of this Agreement shall not constitute or be construed as a waiver of such provision or of the right to enforce such provision. If you fail to comply with any term of this License, YOUR LICENSE IS AUTOMATICALLY TERMINATED. This License represents the entire understanding between the parties with respect to its subject matter.

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, THAT YOU UNDERSTAND THIS AGREEMENT, AND UNDERSTAND THAT BY CONTINUING THE INSTALLATION OF THE SOFTWARE, BY LOADING OR RUNNING THE SOFTWARE, OR BY PLACING OR COPYING THE SOFTWARE ONTO YOUR COMPUTER HARD DRIVE, YOU AGREE TO BE BOUND BY THIS AGREEMENT'S TERMS AND CONDITIONS. YOU FURTHER AGREE THAT, EXCEPT FOR WRITTEN SEPARATE AGREEMENTS BETWEEN EKTRON AND YOU, THIS AGREEMENT IS A COMPLETE AND EXCLUSIVE STATEMENT OF THE RIGHTS AND LIABILITIES OF THE PARTIES.

Copyright 1999 - 2003 Ektron, Inc. All rights reserved.

LA10031, Revision 1.4

What's New in the 4.8 Release

- **Updated Calendar Server Control** - The Calendar Server Control has been updated. The following are new properties:
 - **DisplayType** - Lets you choose the view of the calendar. You can choose from Month, Day, Event, Month with Event, or Inline.
 - **StartDate** and **EndDate** - Allows you to Select the start and end dates for the calendar. For example, a calendar that has the date range of May 5th, 2005 through July 5th, 2005.
 - **ShowXML** - Outputs the raw XML data for use with an XSL. This allows you to create a completely custom calendar. There is a new style sheet for the Calendar Server Control, `calendarStyles.css`.
For more information on the updated Calendar Server Control. See ["Calendar Server Control" on page 92](#).
- **Updated DhtmlMenu Server Control** - The DhtmlMenu Server Control has been updated. The following are new properties:
 - **Align** - Sets the text alignment of the root menu.
 - **CssClass** - Assign a .css class to be used with the root menu.
 - **HorizontalDisplay** - Have the first submenu appear to the right of the root menu, instead of appearing below it.
 - **MenuStyle** - Specify an inline style to be used with the root menu.
For more information on the updated DhtmlMenu Server Control. See ["DhtmlMenu Server Control" on page 106](#).
- **Caching with Server Controls** - An explanation of caching and how to use Server Controls to help define caching parameters. For more information, see ["Caching with Server Controls" on page 127](#).

Summary Table of Contents

What's New in the 4.8 Release	i
About This Manual	1
Ektron CMS400.NET Server Controls.....	2
Standard ASP.NET Methods	130
Web Services	140
Syndicating Web Content as RSS	163
Learning About Visual Studio .NET	168
Additional Information	170



Detailed Table of Contents

What's New in the 4.8 Release	i
About This Manual	1
Ektron CMS400.NET Server Controls.....	2
Opening the Sample Project.....	2
Setting Up a Template	4
Making the Server Controls Available	5
Removing the Server Controls	7
Updating the Server Controls	7
Inserting Server Controls Using Drag and Drop.....	7
Adding a Server Control to a Template.....	7
Build the Project	11
Inserting Server Controls Programmatically.....	11
Steps to Insert a Server Control Programmatically.....	12
Server Control Properties in Code Behind	16
Customizing Server Controls.....	26
Using Drag & Drop and Programmatically Together	29
List of Server Controls	32
Login Server Control.....	33
LanguageSelect Server Control	35
LanguageAPI Server Control.....	37
ContentBlock Server Control	39
FormBlock Server Control	48
Search Server Control	51
IndexSearch Server Control	64
ListSummary Server Control	80
Collection Server Control.....	86
Calendar Server Control.....	92
Menu Server Control	99
DhtmlMenu Server Control.....	106
MetaData Server Control.....	114
MetaDataList Server Control	117
Data Binding with Server Controls.....	122
Steps to Data Binding Using Drag and Drop Server Controls	123
Customizing the DataGrid Table View	125
Caching with Server Controls	127
Standard ASP.NET Methods	130

Replacement of Standard Methods	130
Referencing Methods in Visual Basic	132
Adding a New Web Form.....	132
Requirements for Standard Methods	133
Standard Methods.....	134
Add Content	135
Add Content Example.....	137
Web Services	140
Introduction to Web Services	140
CMS400.NET's Implementation of Web Services	141
Benefits of Using Web Services	141
WS Components and CMS400.NET Architecture	141
Software and Hardware Requirements	141
Installation of Files	142
Setup Instructions	142
Providing Web Service Information in Your CMS	144
CMS400.NET Methods.....	144
The CMS400.NET Content Sample Page	148
Sample Output from One Method.....	152
The News.aspx Sample (non-XML).....	153
The News.aspx Sample (XML).....	156
Creating a Page that Provides a Web Service.....	161
Syndicating Web Content as RSS	163
What is RSS Syndication?	163
The RSS Specification.....	163
RSS Syndication in Ektron CMS400.NET	164
Syndicating via .NET.....	165
Architecture of RSS Syndication	165
aspx pages	165
Creating Your Own aspx Collection Page	165
Example.....	166
Example.....	166
Creating Your Own aspx List Summary Page	166
Example.....	167
Example.....	167
Learning About Visual Studio .NET	168
Grid Layout vs. Flow Layout.....	168
Additional Information	170
Supporting Documents	170

Support	170
Sales	170
Information	170



About This Manual

To utilize the full potential of an Ektron CMS400.NET driven Web site, learning and using the server controls supplied by Ektron is essential. These server controls help make your site more robust and easier to maintain.

This manual can be used as a reference for a developer or the person setting up your Ektron CMS400.NET Web site. This manual explains all server controls supported by Ektron CMS400.NET.

Ektron CMS400.NET Server Controls

Ektron CMS400.NET's server controls let you insert many standard methods and properties within the Visual Studio .NET 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.

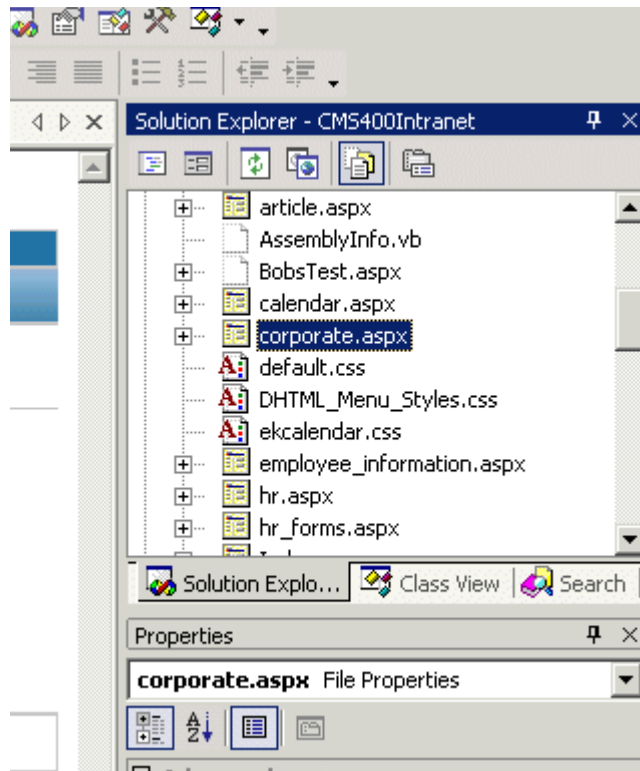
You can insert server controls using drag and drop or programmatically. You can also use databinding to retrieve and display data from **Ektron CMS400.NET**. This chapter explains the use of the Ektron CMS400 Server Controls through the following topics.

- ["Opening the Sample Project" on page 2](#)
- ["Setting Up a Template" on page 4](#)
- ["Making the Server Controls Available" on page 5](#)
- ["Inserting Server Controls Using Drag and Drop" on page 7](#)
- ["Inserting Server Controls Programmatically" on page 11](#)
- ["Using Drag & Drop and Programmatically Together" on page 29](#)
- ["List of Server Controls" on page 32](#)
- ["Data Binding with Server Controls" on page 122](#)
- ["Caching with Server Controls" on page 127](#)

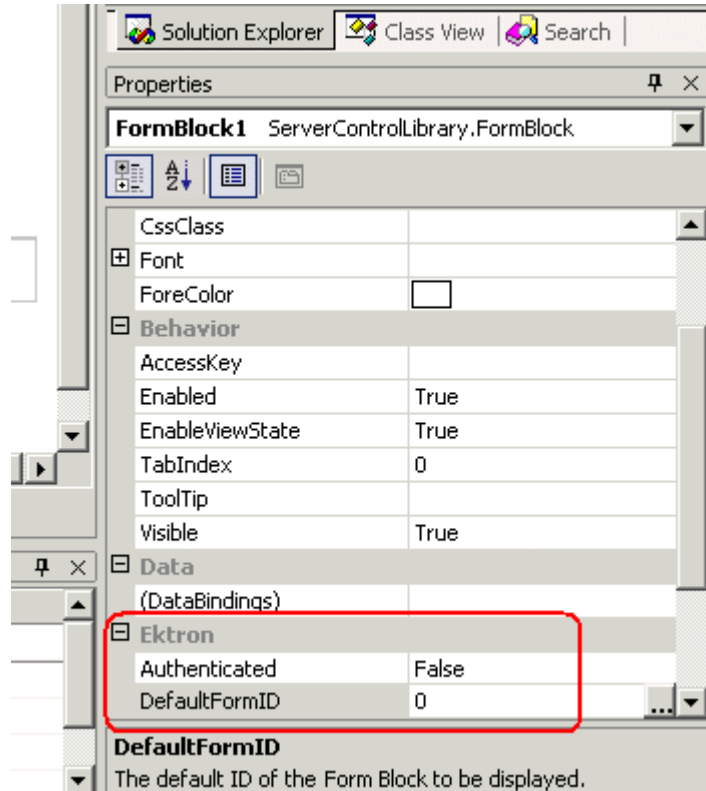
Opening the Sample Project

To help get you started, these directions explain how to open the sample site delivered with **Ektron CMS400.NET**.

1. Browse to and double click Ektron CMS400.NET's solution file, `localhost/CMS400Example/CMS400Example.sln`.
2. The sample site project opens.
3. To work on a template page, click it from the Solution Explorer.



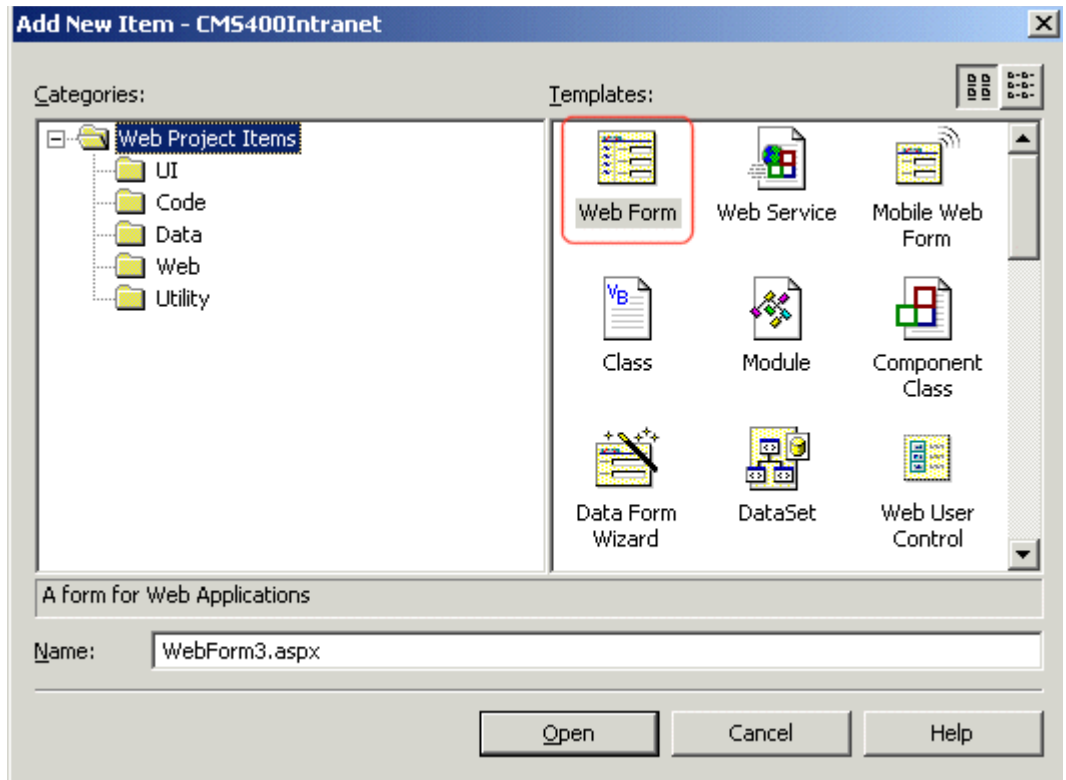
When you select a page, its properties appear in the Properties area (below Solution Explorer), and the page appears in the center of the screen. A control's properties include several standard .NET properties along with Ektron-specific ones. The Ektron properties are labeled as illustrated below.



Setting Up a Template

To create a new template (.aspx) page, follow these steps.

1. Click **Project > Add Web Form**.
2. On the Add New Item screen, click Web Form and assign a name.



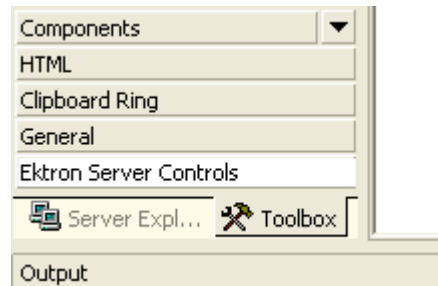
3. Add controls to determine the page content.

Making the Server Controls Available

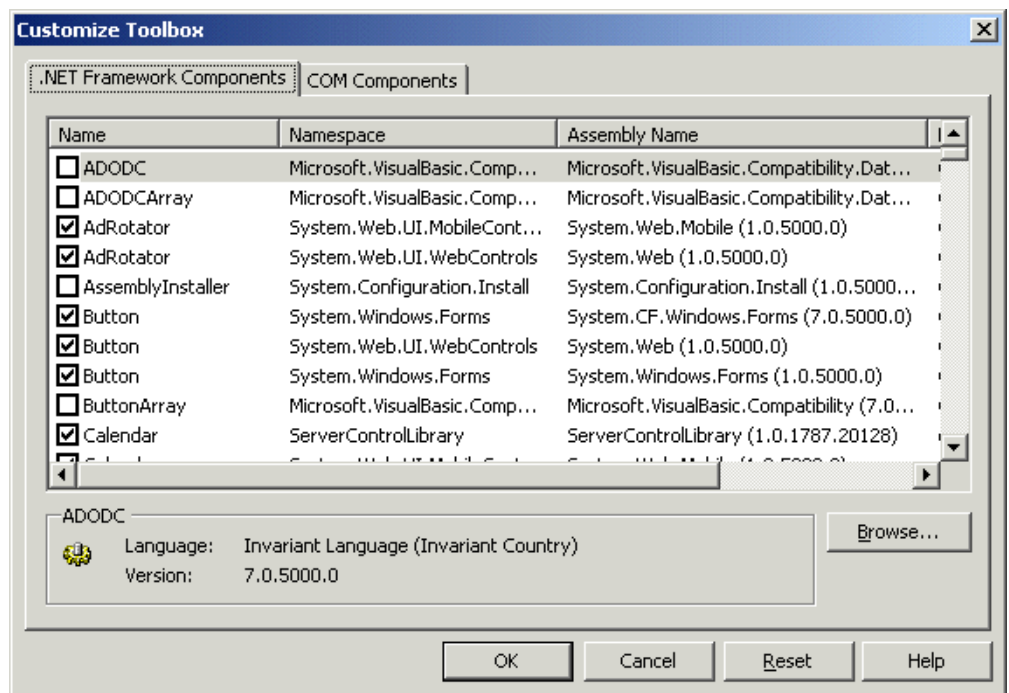
NOTE You must copy the dlls to a local drive before installing them. You cannot copy them from a network drive.

1. Display the Visual Studio .NET toolbox (**View > Toolbox**).
2. Right click the mouse within the Toolbox.
3. Click **Add Tab**.

4. Type **Ektron Server Controls** then press enter.



5. Click the Ektron Server Controls Tab.
6. Right click the mouse in the empty area.
7. Click **Add/Remove Items**.
8. The Customize Toolbox dialog appears.



9. Select the **.NET Framework Components** tab.
10. Browse to the directory that stores Ektron CMS400.NET's dll files, `localhost/CMS400Example/bin`.

11. Add the Ektron.Cms.Controls.dll file and press **OK**. This file provides access to **Ektron CMS400.NET**'s server controls.

Removing the Server Controls

1. Display the Visual Studio .NET toolbox (**View > Toolbox**).
2. Right click the mouse within the Toolbox.
3. Click **Add/Remove Items**.
4. Click **Namespace** or **Assembly Name** to sort the Server Controls by manufacturer.
5. Uncheck all boxes that are Ektron.Cms.Controls.
6. Click **OK**
7. Right click the Ektron Server Control Tab.
8. Click **Delete Tab** to remove it.

Updating the Server Controls

To update the server controls you must first remove the existing server controls in Visual Basic, then add the new server controls.

For information on removing server controls, see ["Removing the Server Controls" on page 7](#)

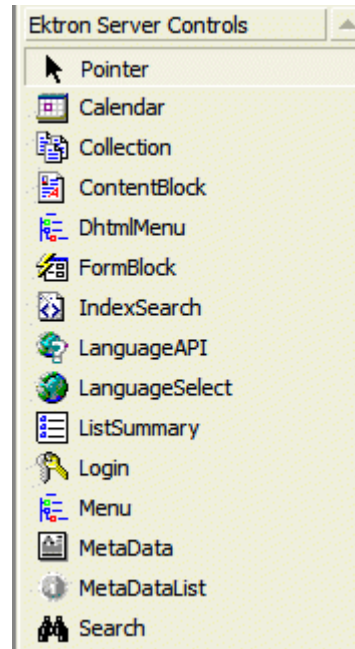
For information on adding server controls, see ["Making the Server Controls Available" on page 5](#)

Inserting Server Controls Using Drag and Drop

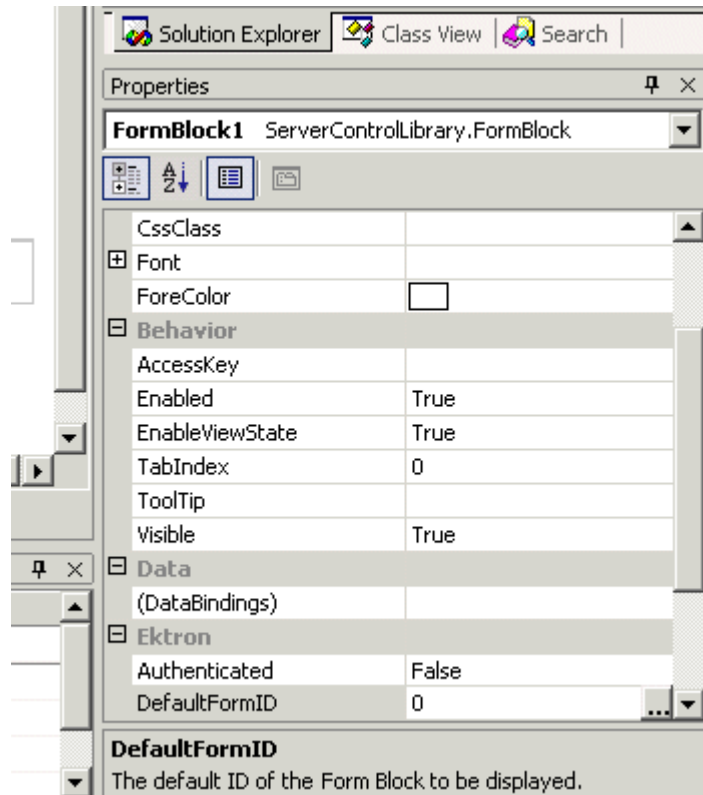
Because Visual Studio .NET is a visual environment, you can watch the page layout change as you add or move a control and adjust its properties. And, whenever you need to identify an **Ektron CMS400.NET** object (such as content block or collection), you can use the CMS Explorer to visually browse to and select it.

Adding a Server Control to a Template

1. Display the Visual Studio .NET toolbox (**View > Toolbox**).
2. Click the **Ektron Server Control** tab. **Ektron CMS400.NET**'s server controls appear.



3. Drag a server control and drop it into the desired location on the page.
4. As desired, modify the control's properties using the Properties area of Visual Studio .NET.



For a description of server control properties, see ["List of Server Controls" on page 32](#).

ServerControlIWS.asmx

Best Practice

For improved security, Ektron recommends renaming or removing the Web services file when you move it to your production server. After installation, this file is named `ServerControlIWS.asmx` and resides in the `/CMS400Example/CMS400Scripts/` folder in your Web root.

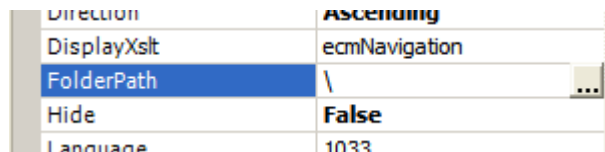
`ServerControlIWS.asmx` is the Web service that lets the server controls talk to **Ektron CMS400.NET**. The path is coded in the `web.config` file. This is how it appears:

```
<!-- Web Service URL for Server Controls design time --> <add key="WSPath" value="http://localhost/cms400example/cms400scripts/ServerControlIWS.asmx" />
```

You will need to edit this line if you change the location or name of the ServerControlWS.asmx file.

Using CMS Explorer to Browse Your Ektron CMS400.NET Site

Ektron CMS400.NET provides a window (called the CMS Explorer) that lets you browse through your Web site to identify CMS objects such as folders, calendars and content blocks. For example, if you insert a List Summary Server Control, its **FolderPath** property identifies the folder whose contents are displayed. If you are not sure of the folder path, click the gray button.



When you do, a login screen appears (illustrated below).

NOTE A user cannot be logged into the CMS Explorer and the CMS400.NET Workarea at the same time. If you log into the CMS Explorer while logged into the Workarea, you will receive the following error when you try to use the Workarea: **Error :** The current user does not have permission to carry out this request [CBR R31]. If you log into the Workarea while logged into the CMS Explorer, you will need to re-login to the CMS Explorer when you return to using it.

CMS Login

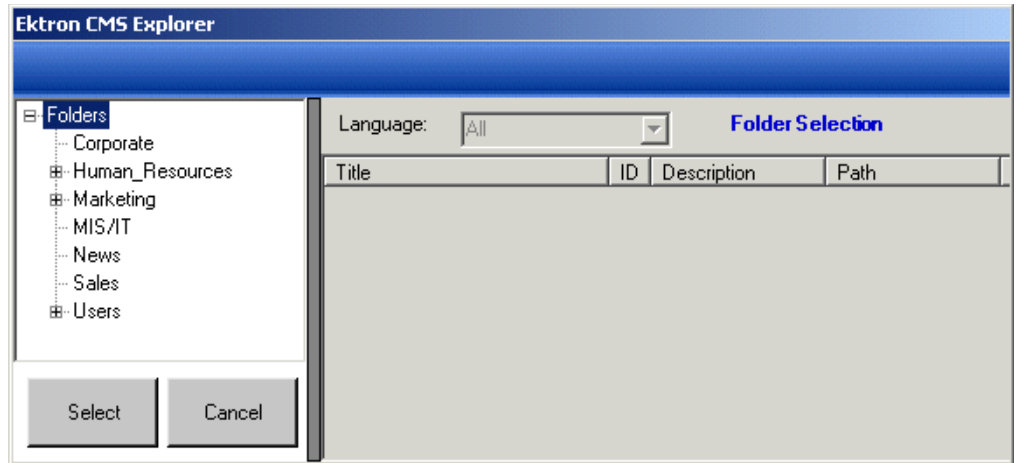
Status: Logged Out

Username:

Password:

Domain:

Enter your **Ektron CMS400.NET** username and password. When you do, the CMS Explorer window appears, providing a visual way to browse your **Ektron CMS400.NET** Web site.



Navigate through the folders, select an object, then click the **Select** button. The selected object is pasted to the server control property.

NOTE Although you see the object in the selected language in Visual Studio .NET, the language is not stored. For example, if you select a German content block whose ID=2, **Ektron CMS400.NET** only stores content block ID=2. When a visitor to your site browses to that page, the content block does not appear in the selected language. Instead, it is determined by a cookie or the user's language selection.

Build the Project

After you open the solution file in Visual Studio .NET and add the required files, build the project. This may take a while the first time you do so. After the project is built, a browser opens and it is displayed as a Web page.

You can also view a Web page while working on it by right clicking on the Web form and clicking **View in Browser**.

Inserting Server Controls Programmatically

Besides using the drag-and-drop method to insert a server control (explained in "Inserting Server Controls Using Drag and Drop" on

page 7), you can insert one programmatically. You might prefer to insert server controls programmatically for several reasons, such as

- you want the control to be loaded into memory only under certain conditions. In this case, insert the logic that only displays the control if the condition exists.
- you want to display only certain properties of an object, such as the title of last edited date of a content block.

Note that you can drag and drop controls on your page then customize them using code behind. For more information, see: ["Using Drag & Drop and Programmatically Together" on page 29.](#)

The following sections explain how to use the server controls programmatically.

- ["Steps to Insert a Server Control Programmatically" on page 12](#)
- ["Server Control Properties in Code Behind" on page 16](#)
- ["Customizing Server Controls" on page 26](#)

Steps to Insert a Server Control Programmatically

To insert an Ektron server control programmatically, follow these steps.

["Step 1: Declare the Namespace" on page 13](#)

["Step 2: Create an Instance of the New Control" on page 13](#)

["Step 3: Set the Properties" on page 14](#)

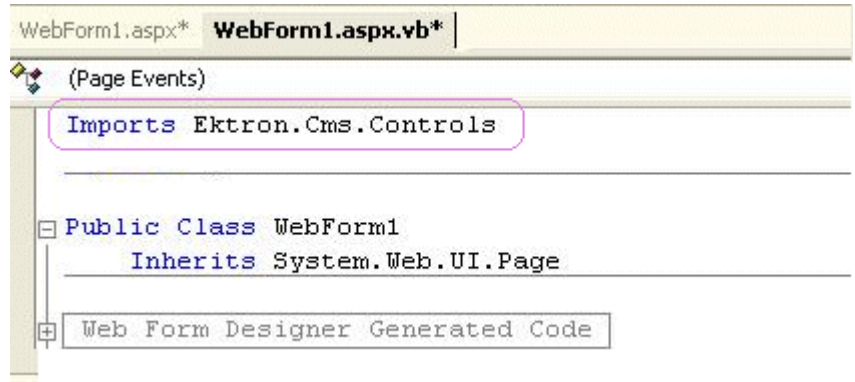
["Step 4: Call the Fill Method" on page 15](#)

["Step 5: Use .text Property to Display the Control" on page 15](#)

Each step is described below. These steps show an example of using the Collection Server Control.

Step 1: Declare the Namespace

Before inserting the server control, you can declare the namespace at the top of the code behind Visual Basic file, as shown below.



```
WebForm1.aspx* | WebForm1.aspx.vb* |
(Page Events)
Imports Ektron.Cms.Controls

Public Class WebForm1
    Inherits System.Web.UI.Page

    Web Form Designer Generated Code
```

NOTE You do not need to declare a namespace. However if you do not, you must fully qualify objects that you create.

Best Practice

For any customization of the CMS, classes or controls that inherit from Ektron classes, it is recommend that you create your own namespace within 'Ektron.Cms.Custom'. For example, if your company is 'AcmeExampleTech, Inc.' you should create all of your custom classes within the namespace 'Ektron.Cms.Custom.AcmeExampleTech'.

Step 2: Create an Instance of the New Control

Next, declare a control as an object in the code behind. In this example, we create a collection named `MyColl`.

```
dim MyColl as New Collection
```

or

```
dim MyColl as new Ektron.Cms.Controls.Collection
```

With C#, use this syntax.

```
Ektron.Cms.Controls.Collection MyColl = new  
Ektron.Cms.Controls.Collection();
```

You can declare any server control as an object by using the server control name in the ["List of Server Controls" on page 32](#). Another example would be: `Dim MyMdl as New MetaDataList`.

Step 3: Set the Properties

NOTE For descriptions of the properties and how to use them, see ["Server Control Properties in Code Behind" on page 16](#).

After inserting the control, you can set the properties that you want to display on the page. For example:

```
Dim MyColl as New Collection  
MyColl.DefaultCollectionID = 1
```

or

```
dim MyColl as new Ektron.Cms.Controls.Collection  
MyColl.DefaultCollectionID = 1
```

With C#, use this syntax.

```
Ektron.Cms.Controls.Collection MyColl = new  
Ektron.Cms.Controls.Collection();  
MyColl.DefaultCollectionID = 1;
```

This line tells the page to display CollectionID 1 unless otherwise specified.

NOTE If you do not know an object's ID number, you can switch to Design mode, drag and drop the object, then use the CMS Explorer to find the ID number. (See ["Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10](#).) If you do this, remember to delete the dropped object when you are done.

You can also obtain the ID number via the Workarea.

```
Dim MyColl as New Collection  
MyColl.DefaultCollectionID = 1  
MyColl.Random = True
```

This line sets the Random property to **true**. See Also: "Random" on page 83

Step 4: Call the Fill Method

Because there is no render event when using objects as components not as controls, use the Fill method to fill an object's properties on the page. An example is below.

```
Dim MyColl as New Collection

MyColl.DefaultCollectionID = 1

MyColl.Random = True

MyColl.Fill()
```

Step 5: Use .text Property to Display the Control

Finally, use the .text property to determine what appears on the Web page. For example, to display the first item in a collection, use this syntax.

NOTE

Before adding this line you need to drag and drop a label on your Web form.

```
Dim MyColl as New Collection

MyColl.DefaultCollectionID = 1

MyColl.Random = True

MyColl.Fill()

Label1.Text = myColl.EkItems(0).Title
```

To display *all* items in a collection, use this syntax.

```
Dim myColl As New Collection

Dim ekitem As New Ektron.Cms.Common.ContentBase

myColl.DefaultCollectionID = 2

myColl.Fill()

Label1.Text = "<ul>"

For Each ekitem In myColl.EkItems

    Label1.Text &= "<li><a href="" & ekitem.QuickLink & "">" & ekitem.Title & "</a>"

Next
```

```
Label1.Text &= "</ul>"
```

This example displays the quick link for every content block in the collection, formatted as a bulleted list. You can use similar code to display a List Summary or search results. The following explains the new (red) code above.

- `label1.Text = ""` displays the opening tag for the bulleted list
- `For Each ekitem In myColl.EkItems` creates a loop for all content blocks in the collection
- `label1.Text &= "" & ekitem.Title & ""` for each content block in the collection, displays its quicklink and title
- `Next` loops through all content blocks in the collection
- `label1.Text &= ""` closes the bulleted list

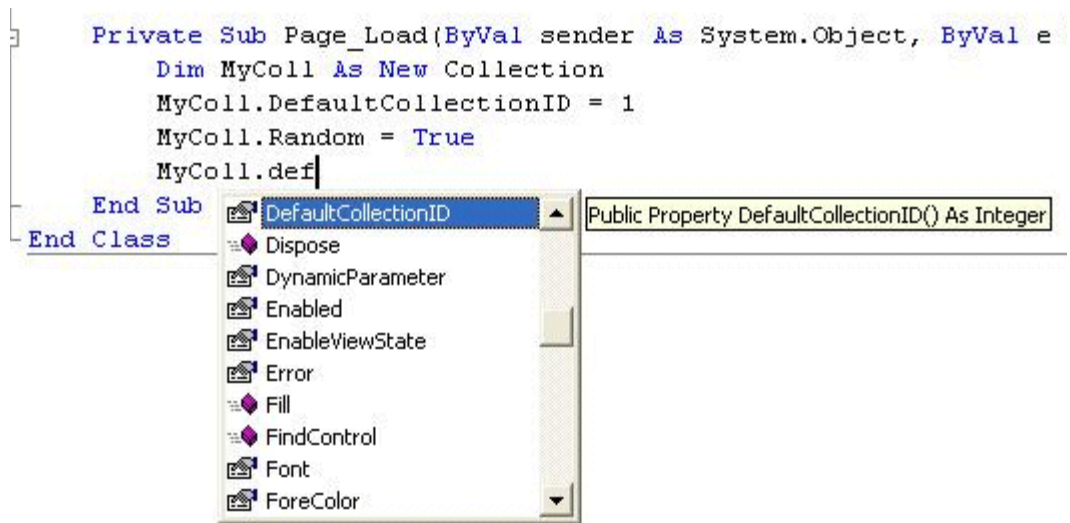
Server Control Properties in Code Behind

Every server control has properties associated with it that you can only access programmatically. This subsection explains, what they are, how to access them, and how to use them.

- ["Displaying Properties for a Control" on page 16](#)
- ["Mutual Server Control Properties in Code Behind" on page 17](#)
- ["Accessing Additional Properties" on page 19](#)
- ["Accessing Items in an Array" on page 25](#)
- ["Referencing the Page Property" on page 25](#)

Displaying Properties for a Control

You can use Visual Studio .NET's intellisense feature to display a control's properties. The intellisense box appears as soon as you insert the period (.) after the object, as illustrated below.



The intellisense box displays *all* properties that can be applied. For a description of Ektron’s standard server control properties, see ["List of Server Controls" on page 32](#). To learn about native VS .NET properties, see its documentation.

The property’s tooltip text indicates its type. In the above example, you can see that the DefaultCollectionID’s type is integer.

For more information about accessing **Ektron CMS400.NET** object properties, see ["Customizing the Server Control in the Code Behind" on page 28](#).

Mutual Server Control Properties in Code Behind

There are several read-only properties that can be called in the code behind and used by every server control. The list below explains these properties.

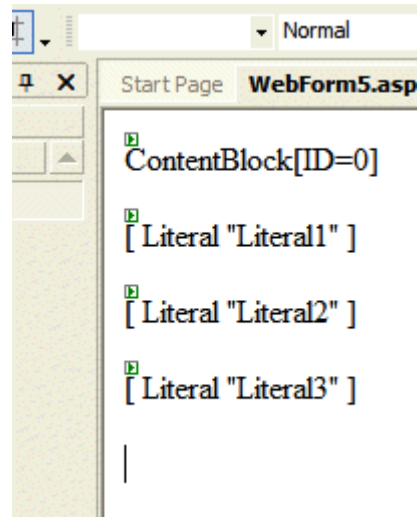
NOTE The properties do not display values within Visual Studio.NET during design time. Instead, they only display values at run time, which are dependent on the user’s login status.

Property	Value	Data Type
IsLoggedIn	Tells if a user is logged in to Ektron CMS400.NET . True = User is logged in False = User is not logged in	Boolean
LoggedInUserName	Gets the Ektron CMS400.NET user name to display.	String
LoggedInUserID	Gets the Ektron CMS400.NET ID of the user to display.	Integer

These properties will allow you to personalize any page with your users names and IDs, and show if they are logged in. Here is an example of using these properties in code behind.

NOTE You must be logged in to **Ektron CMS400.NET** for this example to show your name and ID.

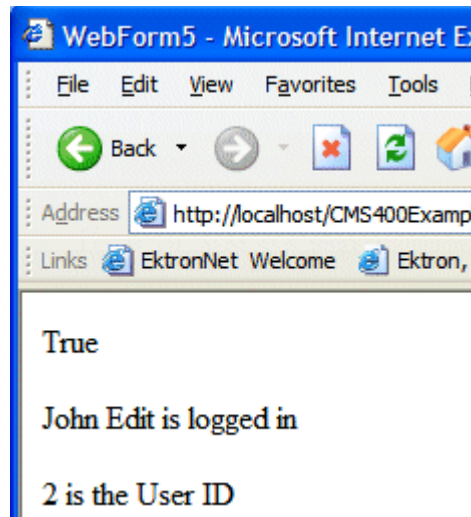
1. Drag an **Ektron CMS400.NET** server control onto a Web form.
2. Drag three Literals onto the Web form.



3. Open the code behind for the Web form.
4. Add the following code to the Page_Load event.

```
Literal1.Text = ContentBlock1.IsLoggedIn  
  
If ContentBlock1.IsLoggedIn Then  
  
    Literal2.Text = ContentBlock1.loggedInUserName & " is logged in "  
  
    Literal3.Text = ContentBlock1.loggedInUserID & " is the User ID "  
  
End If
```

5. Build and run the solution.
6. Browse to the login page and log in.
7. Browse to the new Web form you added.
8. The login information is displayed.



Accessing Additional Properties

In addition to the standard properties, **Ektron CMS400.NET** provides access to additional properties for the following objects.

- List Summary
- Collection
- Search
- ContentBlock
- FormContentBlock

To access additional properties, use the same syntax you use for standard properties but add `.ekitem` or `.ekitems` after the object. Here is an example.

```
dim MyCB as New ContentBlock

label1.text = MyCB.EkItem.dateCreated
```

or

```
dim MyCB as new Ektron.Cms.Controls.ContentBlock

label1.text = MyCB.EkItem.dateCreated
```

IMPORTANT!

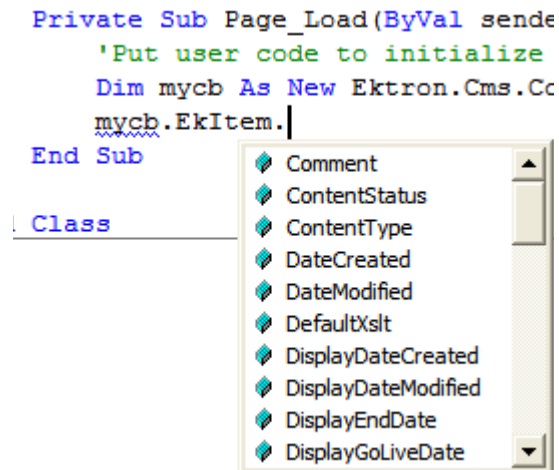
To access additional properties for the Collection, ListSummary, and Search objects, use `ekitems`, not `ekitem`. For example: `MyColl.ekitems(0).dateCreated`, where (0) is the index of the array. For more information on using `ekitems`, see "Accessing Items in an Array" on page 25.

With C#, use this syntax.

```
Ektron.Cms.Controls.ContentBlock MyCB = new
Ektron.Cms.Controls.ContentBlock();

label1.text = MyCB.EkItem.dateCreated
```

You can use intellisense to select from a list of additional object properties, as shown below.



The additional properties are listed below.

NOTE The following properties are read-only. For example, you can get a content block's ID and pass it through to another part of the code, however you cannot set a content blocks ID to be shown. `mycb.Ekitem.id = 8` will not set a content block's ID. The correct way to set a content blocks ID is `DefaultContentID = 8`.

Property	Description	For more information, see
Comment	The content block's comment	
ContentStatus	The status of the content block. <ul style="list-style-type: none"> • approved • checked out • checked in • expired • pending deletion • pending expiration • pending start date • submitted 	Ektron CMS400.NET User Manual, "Appendix A: Content Statuses."
ContentType	One of the following <ul style="list-style-type: none"> • all types • archived content • archived forms • content • forms 	To learn about archived content, see the Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting Archive Options."
DateCreated	The date when the content block was created, formatted as a .NET date type	
DateModified	The date when the content block was modified, formatted as a .NET date type	
DefaultXslt	The default Xslt	
DisplayDateCreated	The date when the content block was created. It is formatted as a string that represents Ektron CMS400's display of the date.	

Property	Description	For more information, see
DisplayDateModified	The date when the content block was edited. It is formatted as a string that represents Ektron CMS400's display of the date.	
DisplayEndDate	The content block's end date. It is formatted as a string that represents Ektron CMS400's display of the date.	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting an End Date on a Content Block"
DisplayGoLiveDate	The content block's start date. It is formatted as a string that represents Ektron CMS400's display of the date.	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting a Start Date"
DisplayStartDate	The content block's start date. It is formatted as a string that represents Ektron CMS400's display of the date.	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting a Start Date"
EndDate	The content block's end date, formatted as a .NET date type	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting an End Date on a Content Block"
EndDateAction	<ul style="list-style-type: none"> • archive display • archive expire • refresh report 	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting Archive Options."
FolderID	The ID of the folder that contains each content block	
GoLiveDate	The content block's start date formatted as a .NET date type	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting a Start Date"
Html	The content that makes up the content block. If content block is in XML it will return it as raw XML content	
Hyperlink	Content block title wrapped by <code><a href></code> tags	

Property	Description	For more information, see
Id	The content block 's ID number	
InheritedFrom	If folder permissions are inherited, the folder from which they are inherited	Ektron CMS400.NET Administrator manual "Setting Permissions" > "Inheritance"
IsInherited	Whether a content block's permissions are inherited	Ektron CMS400.NET Administrator manual "Setting Permissions" > "Inheritance"
IsPrivate	Whether or not a content block is private	Ektron CMS400.NET Administrator manual "Managing Users & User Groups" > "Private Content & Memberships"
Language	The content block's language	Ektron CMS400.NET Administrator manual "Multi-Language Support" > "Working with Multi-Language Content"
LastEditorFname	The first name of the last person to edit the content block	
LastEditorLname	The last name of the last person to edit the content block	
PackageDisplayXSLT	If the content block is XML, the name of its XSLT	
QuickLink	The content block's quicklink	Ektron CMS400.NET User manual "Library Folder" > "Quicklinks"
StartDate	The content block's start date formatted as a .NET date type	Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting a Start Date"

Property	Description	For more information, see
Status	The status of the content block <ul style="list-style-type: none"> • approved • checked out • checked in • expired • pending deletion • pending expiration • pending start date • submitted 	Ektron CMS400.NET User manual “Appendix A: Content Statuses”
Teaser	The content block summary	“Adding a Content Block Summary” in the Ektron CMS400.NET User manual
TemplateLink	Currently empty and not being used with the ContentBlock server control	
Title	The content block title	
UserID	Last user who edited the content	
Xslt1	The content block’s first Xslt, as defined in Ektron CMS400.NET	
Xslt2	The content block’s second Xslt, as defined in Ektron CMS400.NET	
Xslt3	The content block’s third Xslt, as defined in Ektron CMS400.NET	
Xslt4	The developer can use this property programmatically. Ektron CMS400.NET only uses Xslt1, 2 and 3 in the workarea.	
Xslt5	The developer can use this property programmatically. Ektron CMS400.NET only uses Xslt1, 2 and 3 in the workarea.	

Accessing Items in an Array

To access and manipulate content blocks returned by an object, use the common class `Ektron.Cms.Common.ContentBase`. `Ekltems` is an array of `Ektron.Cms.Common.ContentBase`. `Ekltem` is a single `Ektron.Cms.Common.ContentBase`.

Search, Collection and ListSummary have `Ekltems` (an array of `ContentBase`), while `ContentBlock` has a single `Ekltem`. Here is an example of how to use this feature.

```
dim MyC as new Ektron.Cms.Controls.Collection
MyC.DefaultCollectionID = 1

MyC.Fill()

dim item as Ektron.Cms.Common.ContentBase
MyC.Text = "<ul>"

for each item in MyC.EkItems
    MyC.Text &= "<li>" & item.Title & "</li>"
next

MyC.Text &= "</ul>"

Response.Write(MyC.Text())
```

This example formats every item in the collection in a bulleted list.

Referencing the Page Property

If a control requires a reference to its parent page (for example, utilizing the `DynamicParameter` property on a content block to check for a query string), you must provide access to the page object if you declared your control in the code behind. To do this, set the control's page property to the Web page you're working on. For an example, see the code in red and italics below.

```
dim search as new Ektron.Cms.Controls.Search()

MySearch.Page = me

MySearch.Fill()
```

With C#, use this syntax.

```
Ektron.Cms.Controls.Search MySearch = new
Ektron.Cms.Controls.Search();

MySearch.Page = this;

MySearch.Fill();
```

This relationship is only required when inserting a control in the code behind. When dragging and dropping, even if you make changes in code behind, the relationship is automatically generated.

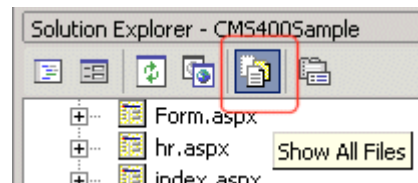
Best Practice

Ektron recommends including the page property reference when using the server controls as components in the code behind.

Customizing Server Controls

One of Visual Studio .NET's strengths is its separation of coding and logic from presentation. Web page formatting is handled by a page's HTML, while the logic is handled by the code behind, which is stored in the corresponding .vb file. For example, if the ASP.NET page is mypage.aspx, the code behind file is mypage.aspx.vb.

NOTE If you do not see the code behind files, click the Show All Files button on the VS



.NET Solution Explorer toolbar.

Within the vb file, you can use Visual Basic to insert code to manipulate the events that occur on the page.

This subsection contains the following topics:

- "Recognizing the Server Control Within the HTML" on page 27
- "Recognizing the Server Control Within the Code Behind" on page 27

- "Customizing the Server Control in the Code Behind" on page 28
- "Troubleshooting Error Creating Control Message" on page 28

Recognizing the Server Control Within the HTML

Within a Web page's HTML, a `<cms>` tag wraps the VS. NET object, as shown below.

```
<cms:Search id="Search1" runat="server" ButtonText="Search" Display="Vertical"></cms:Search>
```

Above is an example of a Search Server Control. Below is an example of a ContentBlock Server Control.

```
<cms:ContentBlock id="ctrlMainContentBlock" runat="server" DefaultContentID="1"
DynamicParameter="id" OverrideXslt="Default"></cms:ContentBlock>
```

Recognizing the Server Control Within the Code Behind

Within the VS .NET code behind file, the Ektron server controls appear (along with the VS .NET controls) in the Web Form Designer Generated Code section. When you click the plus sign (+) to display this section, you see something like the following. The content block listed in HTML above is circled below to help you see their relationship.

```
#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()...
    Protected WithEvents MetaDataArea As System.Web.UI.WebControls.Literal
    Protected WithEvents DhtmlJavaScript As System.Web.UI.WebControls.Literal
    Protected WithEvents DropDownMenu As System.Web.UI.WebControls.Literal
    Protected WithEvents ctrlMainContentBlock As Ektron.Cms.Controls.ContentBlock
    Protected WithEvents ctrlTopContentBlock As Ektron.Cms.Controls.ContentBlock
    Protected WithEvents ctrlBottomLeftContentBlock As Ektron.Cms.Controls.ContentBl
    Protected WithEvents ctrlBottomRightContentBlock As Ektron.Cms.Controls.ContentB
    Protected WithEvents ctrlLanguageSelect As Ektron.Cms.Controls.LanguageSelect
    'NOTE: The following placeholder declaration is required by the Web Form Designe
    Private designerPlaceholderDeclaration As System.Object
```

The next section of the code behind page loads the page into the browser.

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

We want our events to occur while the page is loading, so we'll add custom code following this line.

Customizing the Server Control in the Code Behind

To customize an Ektron server control in the code behind, insert code similar to the following *after* the Page_load command.

```
Dim MyObj As New Ektron.Cms.Controls.ContentBlock
```

This code declares a variable named MyObj and assigns to it the value of a content block. The content block is part of the Ektron.CMS.Controls content base, so it has access to the **Ektron CMS400.NET** database.

After defining MyObj as a content block, you can access its properties. For example, to assign a defaultID of 24, insert the following.

```
Dim MyObj As New Ektron.Cms.Controls.ContentBlock
```

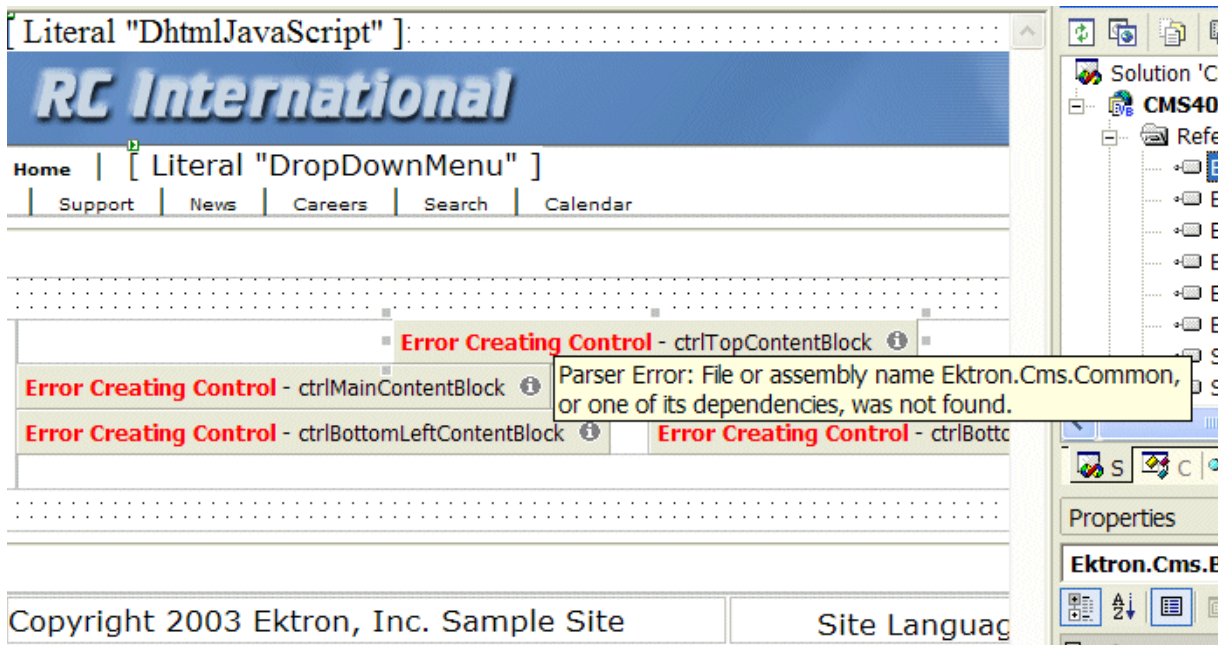
```
MyObj.DefaultContentID = 24
```

Now, the content block can be specified dynamically in the URL of the hyperlink that calls it. If not, content block 24 displays.

This is just an example of programmatically applying property values to content blocks. For a complete list of properties available to **Ektron CMS400.NET** objects, see "[List of Server Controls](#)" on [page 32](#).

Troubleshooting Error Creating Control Message

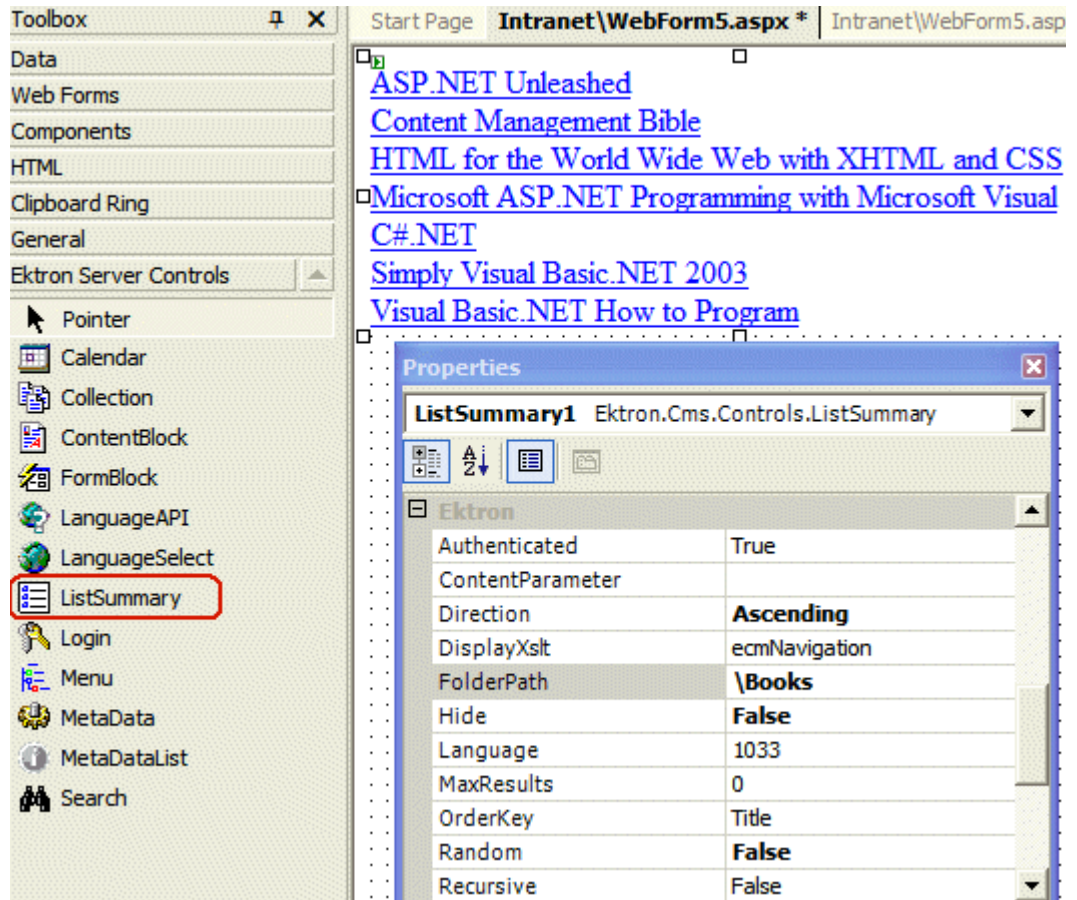
If you get an Error Creating Control message while trying to use a server control, you can view the text of the error message by hovering the mouse over the control. See illustration below.



Using Drag & Drop and Programmatically Together

All server controls can be further manipulated after dragging and dropping them on a web form by using the code behind. Here is an example of using a drag and drop ListSummary Server Control then modifying it programmatically.

First, drag and drop a List Summary on your Web form and set your properties.



Then add the following code to the code behind.

```
Dim myString As String

Dim i

For i = LBound(ListSummary1.EkItems) To UBound(ListSummary1.EkItems)

myString &= "<a href="" & ListSummary1.EkItems(i).QuickLink & "">" &
ListSummary1.EkItems(i).DateCreated & "</a><br>"

ListSummary1.Text = myString

Next
```

An Explanation of the code:

ListSummary1 is the ID of the object. It is used to get access to its properties.

Create a string that contains the output (myString).

```
Dim myString As String
```

Set the object's text property to that string.

```
myString &= "<a href="" & ListSummary1.EkItems(i).QuickLink & "">"  
& ListSummary1.EkItems(i).DateCreated & "</a><br>"
```

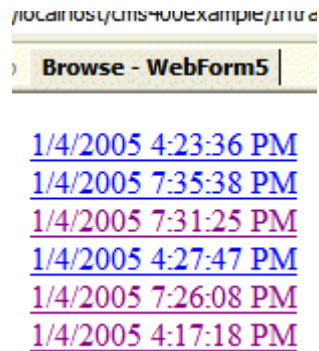
Wrap in a loop so it loops through each of the items.

```
Dim i
```

```
For i = LBound(ListSummary1.EkItems) To UBound(ListSummary1.EkItems)
```

```
Next
```

This example outputs the date created for each content block in a List Summary.



List of Server Controls

To customize your Ektron CMS400.NET application, you can use these server controls.



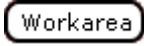
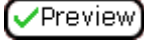

Server Control Names	Description	More Information
Calendar	Displays an event calendar.	"Calendar Server Control" on page 92
Collection	Displays links generated by collections.	"Collection Server Control" on page 86
ContentBlock	Displays and manages a content block.	"ContentBlock Server Control" on page 39
DhtmlMenu	Displays a drop down menu with submenus	"DhtmlMenu Server Control" on page 106
FormBlock	Displays a content block that is associated with a form.	"FormBlock Server Control" on page 48
IndexSearch	Perform XML index searches on XML content.	"IndexSearch Server Control" on page 64 For information on using XHTML Search, see "Search Server Control" on page 51
LanguageAPI	Allows developers to force a language for a Web site.	"LanguageAPI Server Control" on page 37
LanguageSelect	Lets user select language of site.	"LanguageSelect Server Control" on page 35
ListSummary	Displays a list of content block summaries.	"ListSummary Server Control" on page 80
Login	Displays the login or logout button, which lets the user log in and use Ektron CMS400.NET.	"Login Server Control" on page 33
Menu	Calls a defined menu to be used with an Xslt to display the menu.	"Menu Server Control" on page 99

Server Control Names	Description	More Information
MetaData	Adds metadata contained in content blocks to a Web page.	"MetaData Server Control" on page 114
MetaDataList	Creates a list of content blocks based on the metadata contained in the content blocks.	"MetaDataList Server Control" on page 117
Search	Inserts a text box with a search button in the template. The site visitor enters criteria and launches a search of the XHTML content on your Web site.	"Search Server Control" on page 51 For information on using XML Indexing search, see "IndexSearch Server Control" on page 64

Login Server Control

The Login Server Control paints a login button on the template when displayed in a browser.

When the Login Server Control is inserted and the project is built, the control displays the following buttons on a Web page.

Button	Description
 Login	When user is not logged in, this button appears. Clicking the button opens the login window, where a user can enter a username and password. Upon authentication, the user is logged in to the Ektron CMS400 Web site.
 Logout	After a user logs in, this button replaces the login button to let the user log out.
 Workarea	When logged in, this button appears under the logout button, allowing the user to access the Workarea.
 Preview	Lets the user preview the entire Web site as if all checked-in content were published. For more information, see the Ektron CMS400.NET User Manual section "Logging In and Out" > "Site Preview."
 X Preview	Turns off site preview mode.

This subsection contains the following topics:

- ["Placing a Login Button" on page 34](#)
- ["Login Server Control Properties" on page 34](#)

Placing a Login Button

You can add any number of login buttons to a template. You can insert a login button on each template, or set up a special Web page, called login.aspx, from which users can log into the CMS400 site without the public being able to access the page.

Login Server Control Properties

The following table explains the properties of the Login Server Control.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

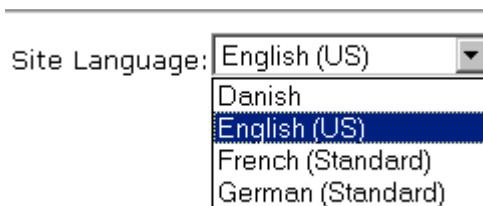
Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
Hide	Used to hide login button in design time and run time. True = Hide login button False = Show login button	Boolean
Language	Design time only: View content blocks, collections, etc. in different languages Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.	Integer

Property	Value	Data Type
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

LanguageSelect Server Control

The LanguageSelect Server Control displays a language selection dropdown list on an Ektron CMS400.NET Web page.

This control lets a visitor to your Web site select a language in which to view the site. Here is what the control looks like when published on a Web page.



It lists all languages selected on the **Settings > Configuration > Language settings** screen. (For more information, see the “Multi-Language Support” chapter of the **Ektron CMS400.NET Administrator manual**.)

You can place this control in any location of any page on your site. On the sample site, it appears in the lower right corner of the home page.

This subsection contains the following topics:

- ["LanguageSelect Server Control vs. LanguageAPI Server Control" on page 36](#)
- ["LanguageSelect Server Control Properties" on page 36](#)

LanguageSelect Server Control vs. LanguageAPI Server Control

The LanguageSelect Server Control lets users view your site in the language they choose. The LanguageAPI Server Control, on the other hand, forces users to view a site in a particular language. For information on LanguageAPI, see "[LanguageAPI Server Control](#)" on page 37.

LanguageSelect Server Control Properties

The following table explains the properties of the LanguageSelect control.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
Hide	Used to hide language select dropdown box and label in design time and run time. True = Hide language select dropdown box and label False = Show language select dropdown box and label	Boolean
LabelName	Lets you define the label next to the language select dropdown box. The default label is Language	String
Language	Design time only: View content blocks, collections, etc. in different languages <u>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</u>	Integer

Property	Value	Data Type
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

LanguageAPI Server Control

The LanguageAPI Server Control lets a developer force a particular language for a Web site. You can do this simply by dropping the server control on the page and then choosing a language in the SiteLanguage property box. You can also override site language logic by programmatically using the LanguageAPI Server Control to detect the browsers language, and display the site in that language.

This subsection contains the following topics:

- ["LanguageSelect Server Control vs. LanguageAPI Server Control" on page 37](#)
- ["LanguageAPI Server Control Properties" on page 37](#)
- ["Code behind Only Properties" on page 38](#)

LanguageSelect Server Control vs. LanguageAPI Server Control

The LanguageSelect Server Control lets users view your site in the language they choose. The LanguageAPI Server Control, on the other hand, forces users to view a site in a particular language. For information on LanguageSelect, see ["LanguageSelect Server Control" on page 35](#).

LanguageAPI Server Control Properties

The LanguageAPI Server Control properties are described in this table.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Value	Data Type
SiteLanguage	Sets the site language. Runs through the IsValid function to verify that the language is an active language in the system.	String
SiteLanguageID	Numeric value of the site language. This is the property you will use if you are using code behind to set the sites language ID. Example: 1036 = French For a list of supported languages, see The Administrator Manual > "Appendix A: List of Supported Languages"	Integer

Code behind Only Properties

The following is a list of LanguageAPI specific properties that can only be used programmatically.

Property	Value	Data Type
DefaultLanguage	Read only. What the default language of the site is set to. For example, the demo site is "English (Standard)".	String
DefaultLanguageID	Read only. Returns the value that is the default language id of the site. For example, the demo site is "1033" for English.	Integer
GetLanguage	This method returns a string. Pass in a valid language ID and it will return the language name.	Integer Argument
GetLanguageID	This method returns an integer. Pass in a valid language name and it will return the language ID.	String Argument
IsValid	This method returns a boolean. You can pass in a language ID or a string and it will tell you if the system is supporting it.	Argument

Property	Value	Data Type
LanguageIdList	Read only. Lists all the language IDs that are activated in Ektron CMS400.NET . For information on how to enable languages, see The Administrator Manual > “Multi-Language Support” > “Working with Multi-Language Content” > “Determining Which Languages are Available”.	Array of Integers
LanguageTitleList	Read only. Lists all the languages that are activated in Ektron CMS400.NET .	Array of Strings
MultiLanguageEnabled	Read only. Tells if the site supports multi-language mode. True = Multi-language enabled False = Multi-language not enabled Multi-language is enable by default in Ektron CMS400.NET . To disable Multi-language, see Administrator Manual > “Multi-Language Support” > “Working with Multi-Language Content” > “Disabling Support for Multiple Language Content”.	Boolean

ContentBlock Server Control

The ContentBlock Server Control displays a content block on an Ektron CMS400.NET Web page.

Ektron CMS400.NET has two types of content blocks:

- Static - displays one specified content block
- Dynamic - displays the content block of the ID passed through a URL parameter

In addition, a content block can contain XHTML or XML content.

For an example of using the ContentBlock Server Control programmatically, see ["Using the ContentBlock Server Control Programmatically Example" on page 47](#)

This subsection contains the following topics:

- ["Static Content Block" on page 40](#)
- ["ContentBlock Server Control Properties" on page 40](#)
- ["Dynamic Content Block" on page 42](#)

- "XML Content Block" on page 44
- "The OverrideXslt and DisplayXslt Properties" on page 45
- "Using the ContentBlock Server Control Programmatically Example" on page 47

Static Content Block

A static content block displays one identified content block on a Web page. The following table explains how to complete the server control properties to insert a static content block on a Web page.

ContentBlock Server Control Properties

The following table explains the properties of the ContentBlock Server Control.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also</i> : "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
DefaultContentID	The ID of a content block that appears where you insert this server control. If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See <i>Also</i> : "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	Integer
DynamicParameter	Select None - Use Default . This parameter is used for dynamic content blocks.	String

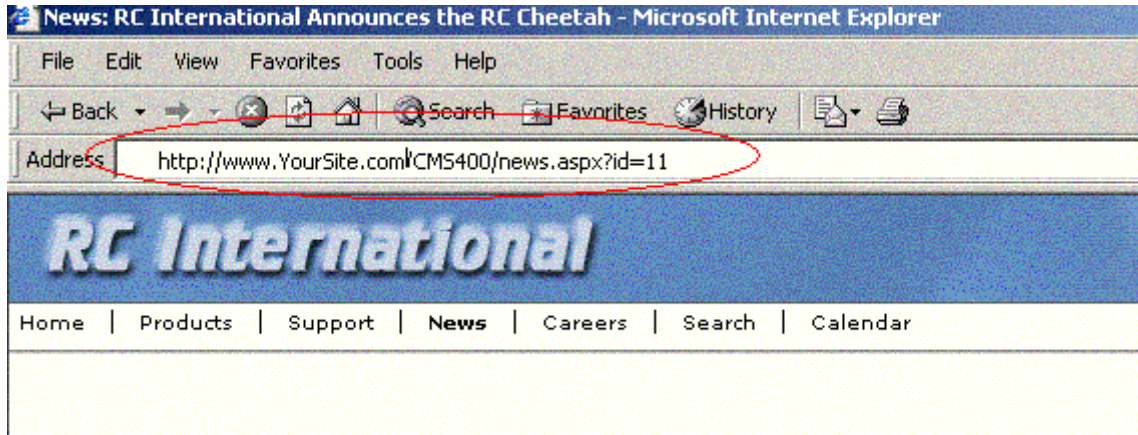
Property	Value	Data Type
Hide	Used to hide a content block in design time and run time. True = Hide content block False = Show content block	Boolean
Language	Design time only: View content blocks, collections, etc. in different languages <u>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</u>	Integer
OverrideXslt	Ignore for a non-XML content block.	must be one of these values: default, one, two, three
DisplayXslt	Ignore for a non-XML content block.	String
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

Ektron	
Authenticated	True
DefaultContentID	28
DisplayXslt	
DynamicParameter	
Hide	False
Language	1033
OverrideXslt	Default

In the above example, the server control retrieves content block id=28 and displays it in the browser.

Dynamic Content Block

Use a dynamic content block to display a content block whose ID is passed through a URL parameter. You would use this server control with a dynamic template.



The following table explains how to complete the ContentBlock Server Control's properties to insert a dynamic content block on a template.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see [VisualStudio.Net help](#).

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String

Property	Value	Data Type
DefaultContentID	The ID of a content block that appears where you inserted this server control if no other content block is identified, or is not available. If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See Also: " Using CMS Explorer to Browse Your Ektron CMS400.NET Site " on page 10	Integer
DynamicParameter	Select id . When you do, this server control uses the content block passed as a URL parameter.	String
Hide	Used to hide a content block in design time and run time. True = Hide content block False = Show content block	Boolean
Language	Design time only: View content blocks, collections, etc. in different languages <hr/> Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list. <hr/>	Integer
OverrideXslt	Ignore for a non-XML content block.	must be one of these values: default, one, two, three
DisplayXslt	Ignore for a non-XML content block.	String
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

The following example shows how the ContentBlock Server Control can be used to pass the id as a URL parameter.

Ektron	
Authenticated	False
DefaultContentID	1
DisplayXslt	
DynamicParameter	id
Hide	False
Language	1033
OverrideXslt	Default

When a user clicks a link that passes the content block ID as a URL parameter, that content block appears. If that content block is not available, content block 1 appears.

XML Content Block

Use an XML content block to display an XML content block on an Ektron CMS400.NET Web page.

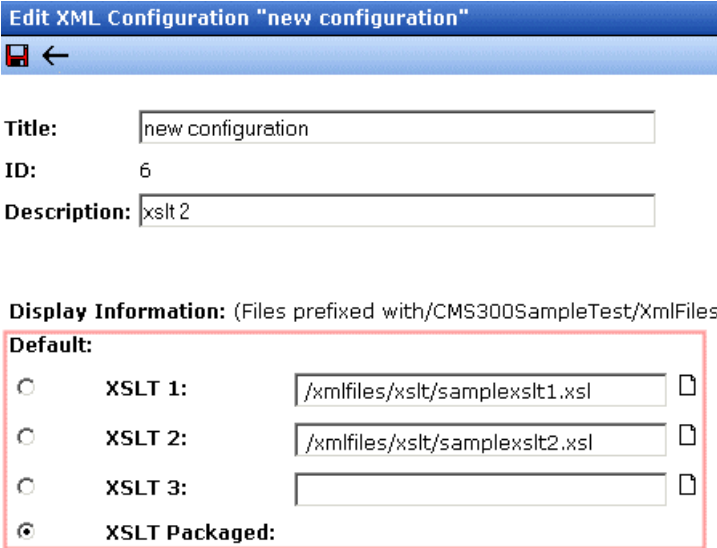
NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also</i> : "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
DefaultContentID	The ID of a content block that appears where you inserted this server control if no other content block is identified, or is not available. If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See <i>Also</i> : "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	Integer

Property	Value	Data Type
DynamicParameter	To make this content block dynamic, select id . When you do, this server control uses the content block passed as a URL parameter.	String
Hide	Used to hide a content block in design time and run time. True = Hide content block False = Show content block	Boolean
Language	Design time only: View content blocks, collections, etc. in different languages <u>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</u>	Integer
OverrideXslt	An internal Xslt file used to display XML content if one is not identified in the Xslt property. See Also: " The OverrideXslt and DisplayXslt Properties " on page 45	must be one of these values: default, one, two, three
DisplayXslt	Specify an external XSLT file. See Also: " The OverrideXslt and DisplayXslt Properties " on page 45	String
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

The OverrideXslt and DisplayXslt Properties

The DisplayXslt property is optional. If used, it specifies an external XSLT file. If the DisplayXslt property is not defined, the OverrideXslt property specifies an XSLT identified in the Edit XML Configuration screen. The following table provides more information about these properties.

To use this display XSLT	Enter this value for XSLT	Enter this value for Override XSLT
<p>The default XSLT specified in the Edit XML Configuration screen (illustrated below). Note that, in this example, the XSLT Packaged option is the default XSLT, since it is selected.</p> <p>XSLT Packaged is the XSLT from the Edit XML configuration screen (that is, the XSLT created in the Data Designer).</p>  <p>Edit XML Configuration "new configuration"</p> <p>Title: <input type="text" value="new configuration"/></p> <p>ID: <input type="text" value="6"/></p> <p>Description: <input type="text" value="xslt 2"/></p> <p>Display Information: (Files prefixed with /CMS300SampleTest/XmlFiles)</p> <p>Default:</p> <p><input type="radio"/> XSLT 1: <input type="text" value="/xmlfiles/xslt/samplexslt1.xsl"/></p> <p><input type="radio"/> XSLT 2: <input type="text" value="/xmlfiles/xslt/samplexslt2.xsl"/></p> <p><input type="radio"/> XSLT 3: <input type="text"/></p> <p><input checked="" type="radio"/> XSLT Packaged:</p>		Default
XSLT 1 from the Edit XML configuration screen		1
XSLT 2 from the Edit XML configuration screen		2
XSLT 3 from the Edit XML configuration screen		3

To use this display XSLT	Enter this value for XSLT	Enter this value for Override XSLT
An absolute or relative path to an XSLT file.	An external XSLT file not specified in the Edit XML configuration screen. For example sample.xslt	does not matter - if XSLT value exists, OverrideXSLT is ignored

See *Also*: The “Managing XML” chapter of the **Ektron CMS400.NET** Administrator manual

Using the ContentBlock Server Control Programmatically Example

The following code will display a content block:

NOTE Before adding these lines of code, you need to drag and drop a literal box on your Web form.

```
Dim MyContentBlock As New ContentBlock

MyContentBlock.DefaultContentID = 8

MyContentBlock.Fill()

Literal1.Text = MyContentBlock.EkItem.Html
```

To display a content block with the content block title, do the following:

NOTE Before adding these lines of code, you need to drag and drop a two literal boxes on your Web form.

```
Dim MyContentBlock As New ContentBlock

MyContentBlock.DefaultContentID = 8

MyContentBlock.Fill()

Literal1.Text = MyContentBlock.EkItem.Title

Literal2.Text = MyContentBlock.EkItem.Html
```

FormBlock Server Control

The FormBlock Server Control displays a content block associated with a form.

IMPORTANT!

If you create a template for an existing form content block, you must manually change its quicklink to point to the new template. This change does not occur automatically. This procedure is described in the Ektron CMS400.NET Administrator manual "Managing Library Assets" > "Updating Default Template for Multiple Quicklinks."

When added to a template and visited, the form content block might look like this. You can change the display to suit your needs by modifying its properties.

Absence Request Form

Name:

E-mail Address:

Position:

Department:

Reason For Absence?

Floating Holiday

Vacation

Sick Leave

FMLA

Other:

Dates of Absence: to

Total Days Absent:

With Pay

Without Pay

Comments:

This subsection contains the following topics:

- ["FormBlock Server Control Properties" on page 49](#)
- ["Automatic versus Manual Generation of Form Tags" on page 50](#)

FormBlock Server Control Properties

The following table explains the properties of the FormBlock Server Control.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see [VisualStudio.Net help](#).

Property	Value	Data Type
AddValidation	Set to True to add Validation to your main runat=server form.	Boolean
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
DefaultFormID	The ID of a FormBlock that appears where you inserted this server control if no other form block is identified. If you don't know the ID number of the form block, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	Integer
DynamicParameter	To make this form block dynamic, select id . When you do, this server control uses the form block passed as a URL parameter.	String
FormName	Set this value to the name of your main runat=server form.	String
Hide	Used to hide a form block in design time and run time. True = Hide form block False = Show form block	Boolean

Property	Value	Data Type
Include Tags	Determines if tags are generated automatically or manually. See Also: "Automatic versus Manual Generation of Form Tags" on page 50	Boolean
Language	Design time only: View content blocks, collections, etc. in different languages. <u>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</u>	Integer
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an in-line portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

Automatic versus Manual Generation of Form Tags

When using the FormBlock Server control's Include tags property, you have two options.

Option	Recommended if you are developing	How to Enable
Let Ektron CMS400.NET generate form tags automatically	A pure script .aspx page	Set Include Tags property to True .

Option	Recommended if you are developing	How to Enable
Modify HTML form tags in the .aspx file	An .aspx page and associated code-behind web form	Set Include Tags property to False . Here is the default .NET generated form tag: <pre><form id="Form1" method="post" runat="server"></pre> Modify the form tag as indicated in red: <pre><form id="Form1" method="post" runat="server" OnSubmit="EkFmValidate(this);"></pre>

Search Server Control

The Search Server Control displays a text box with a search button on the template. To learn about which content is considered by this search function, see the **Ektron CMS400.NET** Administrator manual section “Searching Content on the Web Site.”

When added to a template and visited, the Search Server Control looks like this. Users can enter text and click the search button to search for XHTML content on the Web site.

This subsection contains the following topics:

- ["Search Server Control vs. IndexSearch Server Control" on page 52](#)
- ["Search Server Control Properties" on page 52](#)
- ["Search Server Control Property Example" on page 55](#)
- ["Text Box Examples" on page 57](#)

- ["Button Image Source Examples"](#) on page 58
- ["Search Button Text Examples"](#) on page 59
- ["Display Examples"](#) on page 60
- ["ShowDate Examples"](#) on page 60
- ["Specify Search Control To Display Search Results on a Separate Page"](#) on page 61

Search Server Control vs. IndexSearch Server Control

The Search Server Control lets users search your Web site for XHTML Content. The IndexSearch Server Control lets users search XML content on your Web site. For more information on the IndexSearch Server Control, see ["IndexSearch Server Control" on page 64](#)

Search Server Control Properties

The following table explains the properties of the Search Server Control.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Description	Data Type	For more information, see
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to the folder needed for the FolderPath property.	String	"Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10
ButtonImgSrc	If you want to display an image on the submit button, enter the server path to it.	String	"Button Image Source Examples" on page 58
ButtonText	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.	String	"Search Button Text Examples" on page 59

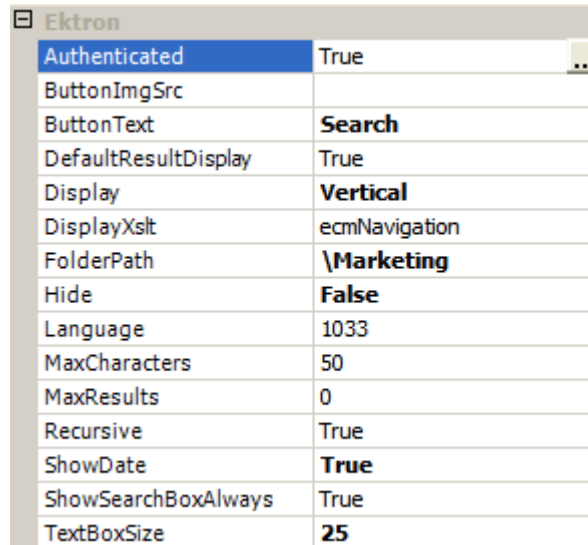
Property	Description	Data Type	For more information, see
DefaultResultDisplay	<p>Select true to use the default Ektron display for search results. Select false to pass the results through as an object to be manipulated in code behind for a customized display. Some examples are to change formatting, or filter the displayed results by using code.</p> <hr/> <p>Note: If you set DefaultResultDisplay to True, the ShowDate and DisplayXslt properties will not work. This is because the server control is using a default display.</p> <hr/>	Boolean	
Display	Determines if the search options are in a row (Horizontal) or a column (Vertical).	either vertical or horizontal	"Display Examples" on page 60
DisplayXslt	<p>Determines how the information on the page is displayed</p> <p>None-databind only</p> <p>ecmNavigation - lists the title of every content block in the folder</p> <p>ecmTeaser - lists the title of every content block in the folder plus the content summary</p> <p>Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page</p> <hr/> <p>Note: If you set DefaultResultDisplay to True, the ShowDate and DisplayXslt properties will not work. This is because the server control is using a default display.</p> <hr/>	String	<p>ecmNavigation - "Example of ecmNavigation Display" on page 89</p> <p>ecmTeaser - "Example of ecmTeaser Display" on page 89</p>

Property	Description	Data Type	For more information, see
EmptyResultMsg	The return message displayed, if the search returns no hits. The default is: Search returned zero results	String	
FolderPath	The folder at which the search begins. The starting folder need not be the root folder. The Recursive property determines if the search examines this folder's subfolders.	String	"Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10
Hide	Used to hide Search Server Control in design time and runtime. True = Hide control False = Show control	Boolean	
Language	Design time only: View content blocks, collections, etc. in different languages. <u>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</u>	Integer	
MaxCharacters	The maximum number of characters the input text box accepts when doing a search.	Integer	
MaxResults	The Maximum number of content blocks returned (0=unlimited).	Integer	
Recursive	Whether to search sub-folders of the identified root folder. The root folder is identified in the Folderpath property.	Boolean	

Property	Description	Data Type	For more information, see
ShowDate	<p>If True, displays the content block's last modified date. This is useful for determining the most recent content block.</p> <hr/> <p>Note: If you set DefaultResultDisplay to True, the ShowDate and DisplayXslt properties will not work. This is because the server control is using a default display.</p> <hr/>	Boolean	"ShowDate Examples" on page 60
ShowSearchBoxAlways	If set to false , the search box does not appear on PostBack	Boolean	
Text Box Size	The size of the display text box for user input. The width is based on the number of characters.	Integer	"Text Box Examples" on page 57
WrapTag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String	

Search Server Control Property Example

The following example places a text field and a search button on a template.

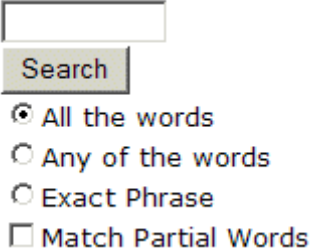
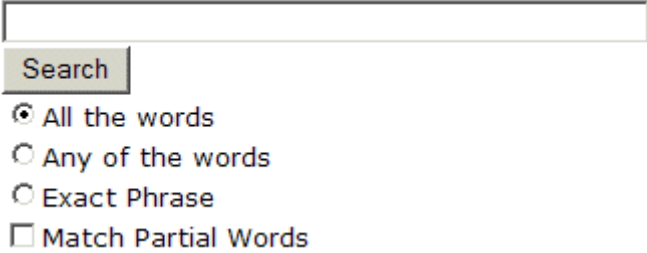


Ektron	
Authenticated	True
ButtonImgSrc	
ButtonText	Search
DefaultResultDisplay	True
Display	Vertical
DisplayXslt	ecmNavigation
FolderPath	\Marketing
Hide	False
Language	1033
MaxCharacters	50
MaxResults	0
Recursive	True
ShowDate	True
ShowSearchBoxAlways	True
TextBoxSize	25




The above example has the following parameters:

- *No image* is used for the button
- The text on the submit button is **Search**
- The results will display on the page
- The search options are arranged *vertically*
- Searches content in the **content\marketing** folder
- The search allows a maximum of **50** characters
- Returns *unlimited* results when the search is performed
- Search is **recursive**
- Shows the *last modified date* of the content
- The search box will appear on PostBack
- The size of the text box is **25** characters

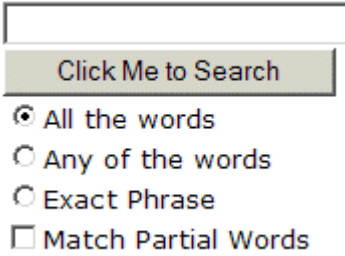
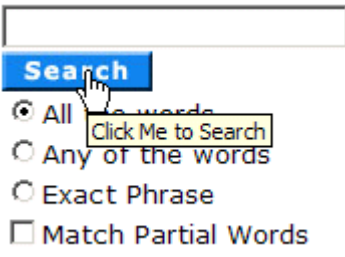
Text Box Examples

Value	Web Page View
10	
50	

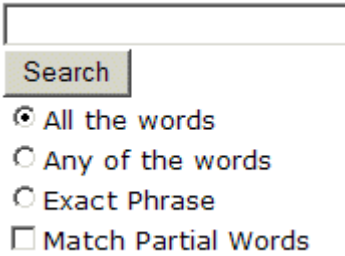
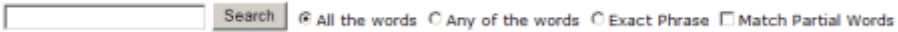
Button Image Source Examples

Value	Web Page View
(blank)	 <p>A search form with a text input field, a button labeled "Search" with a blank image, and four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words" (checkbox).</p>
search.jpg	 <p>A search form with a text input field, a button labeled "Search" with a blue background and white text, and four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words" (checkbox).</p>
http://www.ektron.com/images/search.jpg	 <p>A search form with a text input field, a button labeled "Search" with a blue background and white text, and four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words" (checkbox).</p>


Search Button Text Examples

Value	Web Page View
"Click Me to Search"	 <p>A screenshot of a search interface. At the top is a search input field. Below it is a button with the text "Click Me to Search". Underneath the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words" (checkbox).</p>
"Click Me to Search"	 <p>A screenshot of a search interface. At the top is a search input field. Below it is a button with the text "Search". A mouse cursor is pointing at the button, and a tooltip box containing the text "Click Me to Search" is visible over the button. Underneath the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words" (checkbox).</p>

Display Examples

Value	Web Page View
Vertical	 <p>A vertical search control with a text input field, a 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.</p>
Horizontal	 <p>A horizontal search control with a text input field, a 'Search' button, and four radio button options: 'All the words', 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.</p>

ShowDate Examples

Value	Web Page View
False	 <p>Search Results</p> <p>RC Sportster RC Redstar RC Lilly RC Cheetah Products Page RC International Ships RC Sportster RC International Hosts RC Conference RC International Announces the RC Cheetah</p>

Value	Web Page View
True	<p>Search Results</p> <p>RC Sportster (8/27/2002 7:52:40 PM) RC Redstar (8/27/2002 7:52:01 PM) RC Lilly (8/27/2002 7:51:26 PM) RC Cheetah (8/27/2002 7:50:53 PM) Products Page (8/27/2002 7:50:09 PM) RC International Ships RC Sportster (8/27/2002 2:46:12 PM) RC International Hosts RC Conference (8/27/2002 2:44:44 PM) RC International Announces the RC Cheetah (8/27/2002 2:43:01 PM)</p>

Specify Search Control To Display Search Results on a Separate Page

The following example shows a search dialog box on a Web form and the results displayed on separate Web form. This is done when you have the search dialog box in a navigation pane, and the results displayed on a separate Web form.

1. Drag and drop a Search Server Control on a Web form.

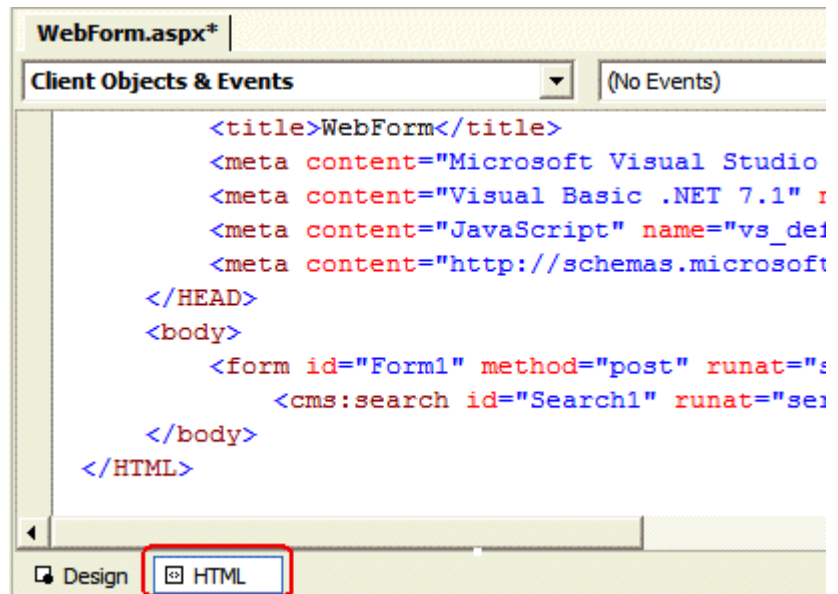
The screenshot shows a search control on a web form. At the top is a search input field with a magnifying glass icon. Below it is a 'Search' button. Underneath the button are four radio button options: 'All the words' (which is selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.

2. Set the following Ektron properties for the Search Server Control.

DefaultResultsDisplay = False

DisplayXslt = None - Databind Only

- Click on **html**. (Circled below)



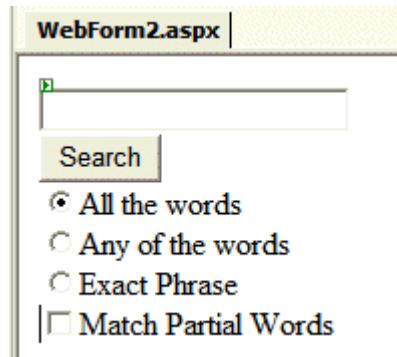
- In the body, after Method="post", replace runat="server" with action="Name of Web form you want the results displayed on". See example below.

NOTE webform2.aspx represents the name of the second webform.

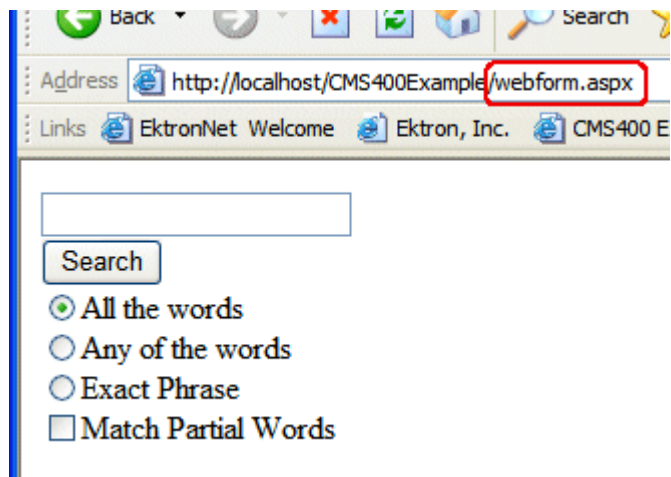
<p>Before</p> <pre><body> <form id="Form1" method="post" runat="server"> <cms:Search id="Search1" runat="server"></cms:Search </body></pre>
<p>After</p> <pre><body> <form id="Form1" method="post" action="webform2.aspx"> <cms:search id="Search1" runat="server" DefaultResu </body></pre>

- Add a second web form.

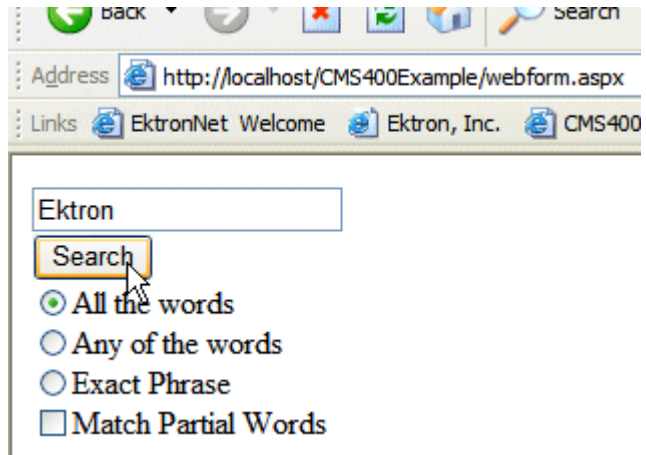
6. Drag and drop a Search Server Control on it.



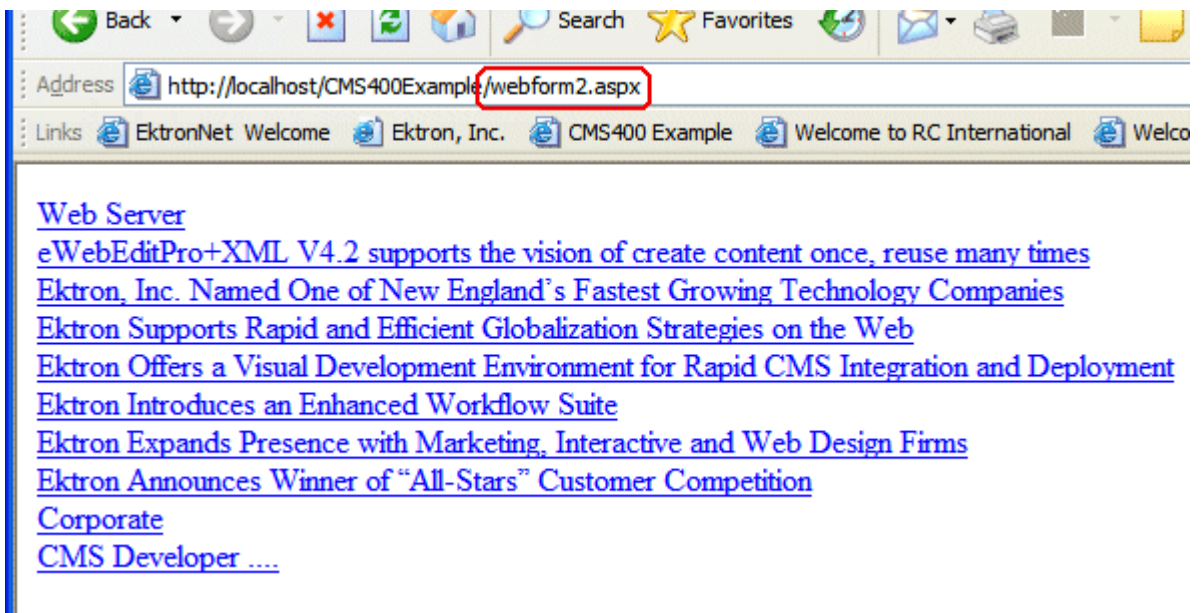
7. Set the Ektron property **ShowSearchBoxAlways** to false.
8. Build the solution.
9. In your browser, go to the first web form.



10. Use the search dialog.



11. The search results are loaded and displayed in the second Web form automatically.

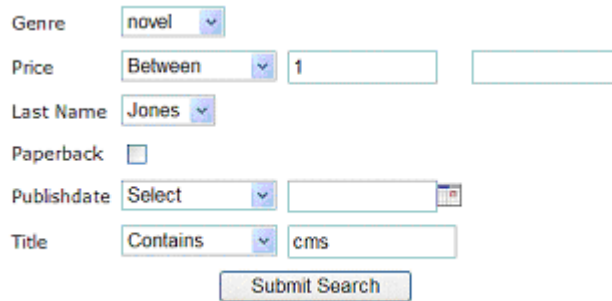


IndexSearch Server Control

Use the IndexSearch Server Control to display an XML configuration's search screen on any Web form. This lets a user search your site based on XML parameters.

See *Also*: The **Ektron CMS400.NET** Administrator Manual section “XML Indexing”.

Below is a sample screen that can be generated via the IndexSearch Server Control.



The screenshot shows a search form with the following fields and values:

- Genre: novel
- Price: Between 1
- Last Name: Jones
- Paperback:
- Publishdate: Select
- Title: Contains cms

A "Submit Search" button is located below the Title field.

You can also predefine a search and have the results appear on a Web form template by using the SearchParmXML property.

This subsection contains the following topics:

- ["IndexSearch Server Control vs. Search Server Control" on page 65](#)
- ["IndexSearch Server Control Properties" on page 65](#)
- ["SearchParmXML Property" on page 70](#)
- ["Using the IndexSearch Server Control Programmatically Example" on page 73](#)

IndexSearch Server Control vs. Search Server Control

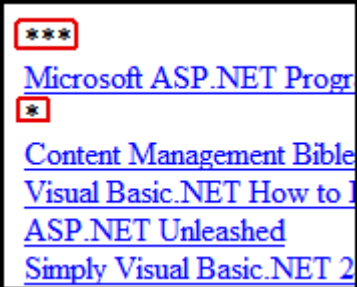
See, ["Search Server Control vs. IndexSearch Server Control" on page 52.](#)

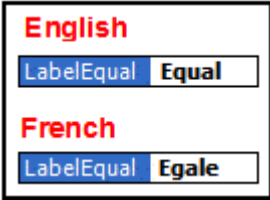
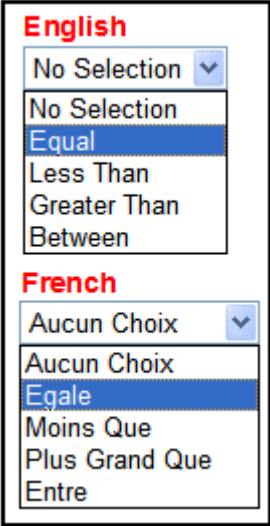
IndexSearch Server Control Properties

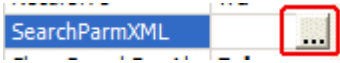
The IndexSearch Server Control properties are described in this table.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Description	Data Type	Values
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to the folder needed for the FolderID property. See Also: " Using CMS Explorer to Browse Your Ektron CMS400.NET Site " on page 10	String	
ButtonText	The text that appears on the button.	String	The default is: Search
ContentParameter	Checks the QueryString for the content block ID value and replaces the search with content a when ID is specified. Leave blank to always display the search.	String	None - Use Default ID - The server control uses the content block passed as a URL parameter.
DisplayXslt	Determines how the information on the page is displayed	String	None -databind only ecmNavigation - lists the title of every content block in the folder See Also: " Example of ecmNavigation Display " on page 89 ecmTeaser - lists the title of every content block in the folder plus the content summary See Also: " Example of ecmTeaser Display " on page 89 Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page

Property	Description	Data Type	Values
EmptyResultMsg	The message that displays if the search returns no hits.	String	The default value is: Your search did not match any documents.
FolderId	The numeric id of the folder that you want to search. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10 The Recursive property determines whether the IndexSearch Server Control will use this folder's child folders in the search.	Integer	Any folder's numeric ID 0 (zero) represents the root folder.
Hide	Used to hide output of the index search in design time and run time. For example, you want to pass the results to code behind for further manipulation.	Boolean	True = Hide results False = show results
HitsMarker	This symbol shows how many search criteria each hit matched. The following example shows the first hit met three of the criteria. The second group met one criterion. 	String	* # + (Custom) Custom lets you choose the marker to use. The custom marker can be any text, numeric, or symbol character except & and <. <p>For more Information, see The Ektron CMS400.NET Administrator Manual section "XML Indexing" > "Displaying Search Results"</p>
	This property is used when the weighted property is set to true. See Also: "Weighted" on page 70.		

Property	Description	Data Type	Values
Labelafter Labelbefore LabelBetween LabelContains LabeldateBetween LabelEqual LabelExactPhrase LabelGreaterThan Labelinputdate Labelinputnumber LabelLessThan Labelon LabelSelect	<p>All of these properties let you change their label names. This is useful for Multi-language issues. For example, if you have a French site, you change Equal to Égale.</p>  <p>Note: These label properties will not change based on the Language property's setting. For example, setting the Language property to 1036 will not change these labels to French.</p>	String	<p>Examples of what the user will see on the site in French and English.</p> 
Language	<p>Design time only: View content blocks, collections, etc. in different languages.</p> <p>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</p>	Integer	

Property	Description	Data Type	Values
OrderBy	The order of the list to be returned. For example, you want to sort search results by last modified date.	String	<ul style="list-style-type: none"> • Title - The title of the content block • ID - The XML content block ID number • Date Created - The date the content block was created • Date Modified - The date the content block was last modified • AuthorLName - The last author's last name • AuthorFname - The last author's first name • StartDate - The GoLive Date
OrderByDirection	The direction the results will be sorted in. This property works with the OrderBy property only.	String	Ascending Descending
Recursive	Determines whether the IndexSearch Server Control will use this folder's child folders in the search.	Boolean	True False
SearchParmXML	<p>Opens a separate window that lets you predefine a search. For example, you want a predefined list of books to appear on a web page.</p> <p>To access the window click the gray button.</p> 		For more information on SearchParmXML, see "SearchParmXML Property" on page 70
ShowSearchBoxAlways	If set to false , the search box does not appear onPostBack	Boolean	True False

Property	Description	Data Type	Values
Weighted	<p>If you set this value to true, content blocks are ranked by the number of criteria matches.</p> <p>For example, if a content block matches all search criteria, it appears at the top of the results page. A hit marker appears for each match.</p> <p>If a second content block only matches one criterion, it also appears on the results page underneath the first content block. One asterisk appears above the page's title to indicate the single match.</p>	Boolean	<p>True - display every content block that matches <i>at least one</i> search criteria</p> <p>False - show only content blocks that match <i>all</i> search criteria</p>
WrapTag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p>	String	<p>Span - The tag is used to designate an in-line portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>
xmlConfigID	ID of the XML configuration whose indexed fields will appear on the search screen.	Integer	

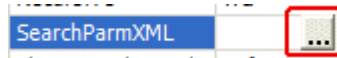
SearchParmXML Property

The SearchParmXML property is used to create a predefined XML search, whose results appear upon page load. Clicking on the gray box in this property opens a window, where you define the search. The parameters in this window are defined by the xmlConfigID you choose. Using this property launches the CMS Explorer login, if you are not already logged in. For more information on the CMS

Explorer login, see ["Using CMS Explorer to Browse Your Ektron CMS400.NET Site"](#) on page 10

The following example uses the SearchParmXML property to create a list of books, based on publishers with "press" in their name.

1. To activate the window, click the gray box in the SearchParmXML property.



2. The Build Search Parameter screen appears.

A screenshot of a dialog box titled 'Build Search Parameter'. The dialog has a blue title bar with standard window controls. The main area is divided into four sections, each for a different search parameter. Each section includes a checkbox, a text field, a dropdown menu, and a data type label. The parameters are: 1. '/root/publisher' (checkbox unchecked, dropdown 'Select One', data type 'string'). 2. '/root/pdate' (checkbox unchecked, dropdown 'Select One', data type 'date'). 3. '/root/price' (checkbox unchecked, dropdown 'Select One', data type 'decimal'). 4. '/root/authors/author/lastname' (checkbox unchecked, dropdown 'Select One', data type 'string'). Each section also has two empty text input fields below the dropdown.

- Click the check boxes for the parameters you want to search with.

The screenshot shows a window titled "Build Search Parameter" with four sections, each representing a search parameter:

- Search Parameter 1:** /root/publisher (string). The dropdown menu is set to "Select One".
- Search Parameter 2:** /root/pdate (date). The dropdown menu is set to "Select One".
- Search Parameter 3:** /root/price (decimal). The dropdown menu is set to "Select One".
- Search Parameter 4:** /root/authors/author/lastname (string). The dropdown menu is set to "Select One".

- Define the parameter.

The screenshot shows the "Build Search Parameter" dialog box with the first parameter defined:

- Search Parameter 1:** /root/publisher (string). The dropdown menu is set to "Contains" and the text input field contains "Press".

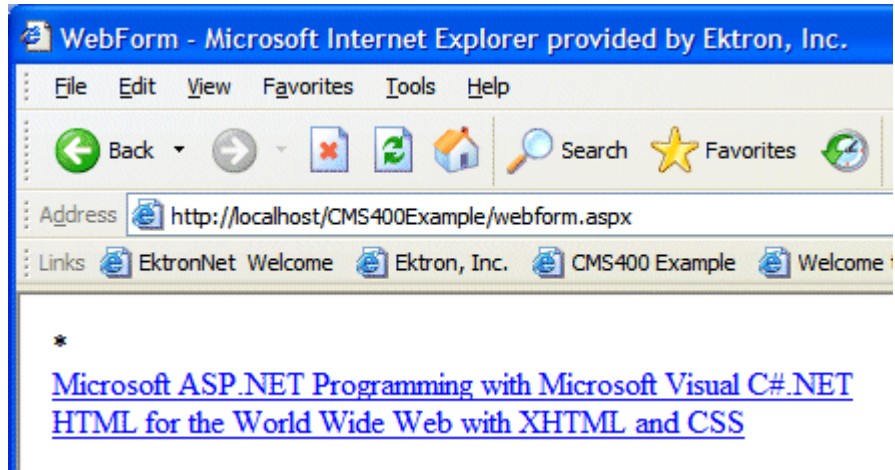
The drop down box in this example contains the following variables:

Select One - Prompts you to choose a variable.

Exact Phrase - You enter a phrase and search for that exact phrase, contained within the XML content.

Contains - You enter a word or phrase and search for any XML content, that has that word or phrase contained within it.

5. Click OK.
6. Build the Web form.
7. Browse to the Web page in your browser.



8. The page appears with a list of XML content blocks that match the parameters you set.

Using the IndexSearch Server Control Programmatically Example

NOTE When creating an IndexSearch Server Control form user interface using code behind only, you need the IndexSearch Server Control to be connected with the page events. Otherwise you will not be able to submit your search parameters.

You can use the IndexSearch Server Control, in code behind, to manipulate the way users view the search results. The following example, using code behind, shows how to return values associated with an XPath. This example shows a list of book subject titles.

1. Open a Web form.
2. Drag and drop an IndexSearch Server Control on your Web form.
3. Set the properties in the properties window.

NOTE In this example, the hide setting must be set to **true**. Remember to set the XmlConfigID. This example uses ID number 4.

4. Drag and drop a Literal on the Web form.

The screenshot shows the Visual Studio IDE with a Properties window open for the **IndexSearch1** control. The control is of type **Ektron.Cms.Controls.IndexSearch**. The Properties window displays a list of properties and their values. Two properties, **Hide** and **XmlConfigId**, are highlighted with red boxes.

Property	Value
Authenticated	True
ButtonText	Search
ContentParameter	
DisplayXslt	ecmNavigation
EmptyResultMsg	Your search did not match any
FolderId	0
Hide	True
HitsMarker	*
Labelafter	After
Labelbefore	Before
LabelBetween	Between
LabelContains	Contains
LabeldateBetween	Between
LabelEqual	Equal
LabelExactPhrase	Exact Phrase
LabelGreaterThan	Greater Than
Labelinputdate	Input needs to be a date
Labelinputnumber	Input needs to be a number
LabelLessThan	Less Than
Labelon	On
LabelSelect	No Selection
Language	1033
OrderBy	Title
OrderByDirection	Descending
Recursive	True
SearchParmXML	
ShowSearchBoxAlways	False
Weighted	True
XmlConfigId	4

5. Add the following code to the code behind.

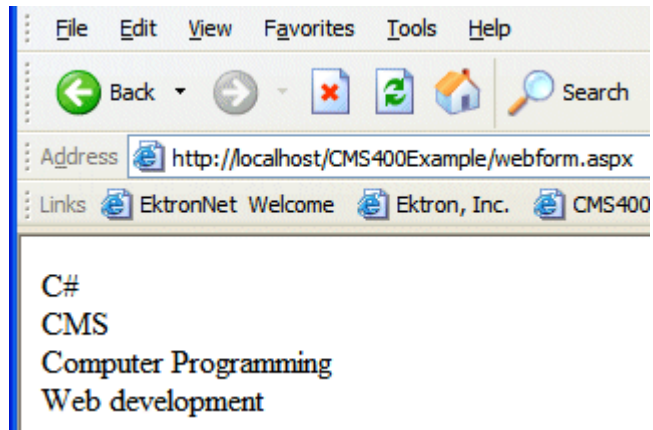
```
IndexSearch1.Parm.XPath = "/root/subject"  
  
Dim arStr As String()  
  
Dim Str As String  
  
Dim strRet As String  
  
arStr = IndexSearch1.GetXPathValues("/root/subject")  
  
For Each Str In arStr  
    strRet += Str & "<br/>"  
  
Next  
  
Literal1.Text = strRet
```

Here is a description of what is happening in the code.

- This line defines the parameter XPath location:
IndexSearch1.Parm.XPath = "/root/subject"
- The next three lines create variables:
Dim arStr As String()
Dim Str As String
Dim strRet As String
- This line set the variable arStr as an array of the XPathValues:
arStr = IndexSearch1.GetXPathValues("/root/subject")
- The for next loop iterates through the array of XPathValues:

```
For Each Str In arStr  
    strRet += Str & "<br/>"  
  
Next
```

- The last line displays the values on your web form:
Literal1.Text = strRet
6. Build the project.
 7. Browse to your web form.
 8. The list of book subject titles appears.



Programmatically Predefined General Search Results Replaced By Specific Search Results.

This Code behind example shows a predefined search appearing on a Web page, then being replaced by more specific search results. This example creates a hyperlink list of book publishers. When you click a specific publisher, it replaces the publisher list with a book list from that publisher.

Follow these steps.

1. Add an IndexSearch Server Control to a Web form.
2. Set the following properties:
 - DisplayXslt = ecmTeaser
 - Hide = True
 - HitsMarker = (blank)
 - ShowSearchBoxAlways = False
 - Weighted = True
 - XmlConfigId = 4
3. Add two Literals.
4. Add the following code to the Page Load Event in code behind.

```
Dim bRet As Boolean

If Request.QueryString("publisher") <> "" Then
    IndexSearch1.Hide = False
```



```
IndexSearch1.Parm.XPath = "/root/publisher"

IndexSearch1.Parm.DataType = Ektron.Cms.Common.EkSearch.XMLDataType.String

IndexSearch1.Parm.SearchType = Ektron.Cms.Common.EkSearch.XMLSearchRangeType.ExactPhrase

IndexSearch1.Parm.Value1 = Request.QueryString("publisher")

bRet = IndexSearch1.AddParm()

IndexSearch1.Search()

Literal1.Text = "<h1>Publisher " & Request.QueryString("publisher") & " - Books</h1>"

Literal2.Text = "<a href=""webform.aspx"">Back</a>"

Else

    Literal1.Text = "<h1>Publishers</h1>"

    Dim arStr As String()

    Dim Str As String

    Dim strRet As String

    arStr = IndexSearch1.GetXPathValues("/root/publisher")

    For Each Str In arStr

        strRet += "<a href=""webform.aspx?publisher=" & Server.UrlEncode(Str) & """">" & Str &
"</a><br/>"

    Next

    Literal1.Text += strRet

End If
```

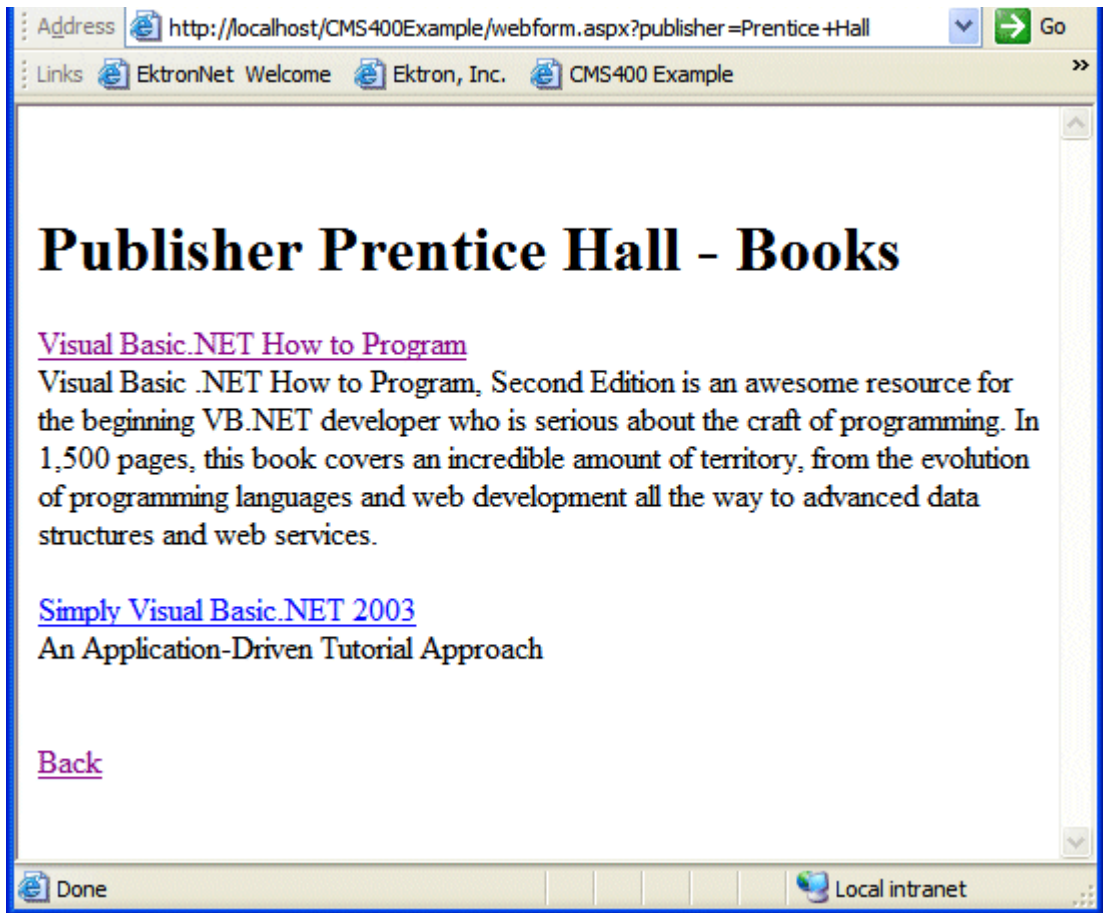
5. Save your Web form.
6. Build your Web form.
7. Browse to the Web form in your browser.

8. The list of publishers appears.



9. Click on a publisher.

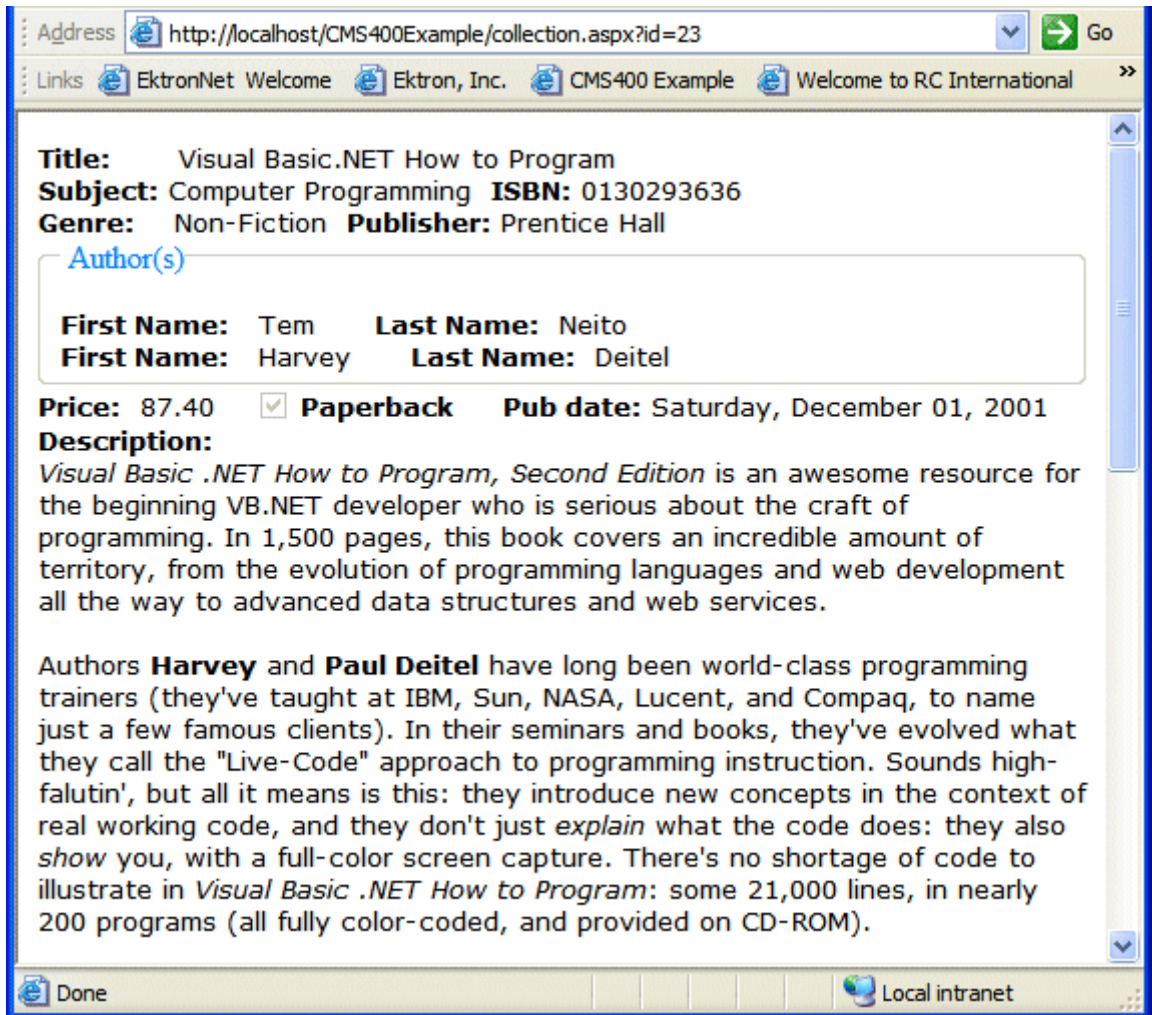
10. A list of books with teasers by that publisher appears.



Notice the second Literal now contains a hyperlink to go back.

11. Click on a book.

12. The XML Content Block for that book appears.



ListSummary Server Control

The List Summary Server Control is used to display a list of content blocks on a Web page.

When added to a template and visited, a List Summary looks like this.



It displays each content block's title and summary information. You can modify the display to suit your needs by modifying its properties. The List Summary Server Control properties are described in this table.

This subsection contains the following topics:

- ["ListSummary Server Control Properties" on page 81](#)
- ["List Summary Example" on page 84](#)
- ["OrderKey Property" on page 85](#)

ListSummary Server Control Properties

The ListSummary Server Control properties are described in this table.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

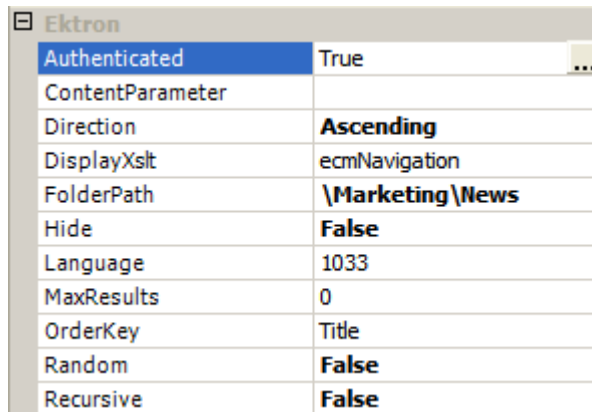
Property	Description	Data Type	Values
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to the folder needed for the FolderPath property. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String	
ContentParameter	Checks the QueryString for this value and replaces the collection with a content block when specified. Leave blank to always display the Collection or List Summary.	String	
Direction	How to order the hyperlinks on the list. The sort field is determined by the OrderKey property.	either ascending or descending	ascending - they are arranged A, B, C or 1,2,3. descending - they are arranged. Z, Y, X or 3,2,1 If sorting by date, descending puts the most recent first.
DisplayXslt	Determines how the information on the page is displayed	String	None -databind only ecmNavigation - lists the title of every content block in the folder See Also: "Example of ecmNavigation Display" on page 89 ecmTeaser - lists the title of every content block in the folder plus the content summary See Also: "Example of ecmTeaser Display" on page 89 Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page

Property	Description	Data Type	Values
FolderPath	<p>The folder that contains the content blocks which appear in the List Summary. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10</p> <p>The Recursive property determines whether content blocks in this folder's child folders also appear.</p>	String	Any folder
Hide	Used to hide output of List Summary in design time and run time.	Boolean	True = Hide results False = show results
Language	<p>Design time only: View content blocks, collections, etc. in different languages.</p> <hr/> <p>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</p> <hr/>	Integer	Integer
MaxResults	Maximum number of summaries displayed on a Web page.	Integer	any integer 0 = unlimited
OrderKey	Sort the list by one of the values.	must be one of the values	Title DateModified DateCreated LastEditorFname LastEditorLname See Also: "OrderKey Property" on page 85
Random	Set to True if you want to randomly display one content block in the specified folder. The content changes each time a user views the page.	Boolean	True False

Property	Description	Data Type	Values
Recursive	Determines if the display includes content blocks in child folders of the selected folder.	Boolean	True False
WrapTag	Allows a developer to specify a server control's tag. The default is Span .	String	Span - The tag is used to designate an inline portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.

List Summary Example

Below is an example of the List Summary Server Control properties.



Property	Value
Authenticated	True
ContentParameter	
Direction	Ascending
DisplayXslt	ecmNavigation
FolderPath	\Marketing\News
Hide	False
Language	1033
MaxResults	0
OrderKey	Title
Random	False
Recursive	False

The example has the following properties:

- Direction - The results are displayed in ascending order by title
- FolderPath - Displays summaries from the \Marketing\News folder
- MaxResults - Unlimited results appear

- OrderKey - Summaries sorted by content block title
- Random - Displays summary for all content blocks in folder, not just one random one
- Recursive - Includes only selected folder, not its child folders
- Xslt - will use the navigation display format

OrderKey Property

The OrderKey property determines what information appears to the right of the hyperlinked title. The values are described below.

Value	Description	Web Page View
DateModified	Displays the date the content was last modified.	, RC Sportster. The RC Sportster is design mid level RC racer currently on the market ference (8/27/2002 2:44:44 PM) growing company in the RC host the 3rd annual RC conference in Atl
DateCreated	Displays the date when the content was first created.	mid level RC racer currently on the mark ference (4/25/2002 4:42:55 PM) growing company in the RC ost the 3rd annual RC conference in A
LastEditorFname	Displays the first name of the user who last edited the content block.	mid level RC racer currentl ference (John) growing company in the R host the 3rd annual RC cc

Value	Description	Web Page View
LastEditorLnames	Displays the last name of the user who last edited the content block.	<p>ct, KC Sportster. The KC Sp d mid level RC racer currentl</p> <p>Conference (Doe) st growing company in the R to host the 3rd annual RC c</p>

Collection Server Control

The Collection Server Control is used to display a list of content block links that were created as a collection in the Ektron CMS400.NET Workarea. The control is highly customizable, allowing you to define the display of the collection on a Web page.

When added to a template and visited, collections can look like this (The following illustration shows three collections on the same page of a sample site.) You can modify the display by editing its properties.

Human Resources Forms

[While you were out...](#)

Employee Forms

[Absence Request Form](#)

[Direct Deposit Form](#)

Manager Forms

[Personnel Action Form](#)

This subsection contains the following topics:

- ["Collection Server Control Properties" on page 87](#)

- ["Example of ecmNavigation Display" on page 89](#)
- ["Example of ecmTeaser Display" on page 89](#)
- ["Using the Collection Server Control Programmatically Example" on page 90](#)

Collection Server Control Properties

The Collection Server Control properties are described in this table.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Description	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
ContentParameter	Checks the QueryString for this value and replaces the collection with a content block when specified. Leave blank to always display the Collection or List Summary.	String
DefaultCollectionID	The ID of a collection that appears where you inserted this server control if no other collection is identified, or is not available. If you don't know the ID number of the collection, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	Integer

Property	Description	Data Type
DisplayXslt	<p>Determines how the information on the page is displayed</p> <p>None-databind only</p> <p>ecmNavigation - lists the title of every content block in the collection</p> <p>See Also: "Example of ecmNavigation Display" on page 89</p> <p>ecmTeaser - lists the title of every content block in the collection plus the content summary</p> <p>See Also: "Example of ecmTeaser Display" on page 89</p> <p>Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page</p>	String
DynamicParameter	To make this collection dynamic, select coll_id . When you do, this server control uses the collection passed as a URL parameter.	String
GetHtml	Set to True if you want to retrieve and display content (html body) for all content blocks in the collection. For example, to display content inside a web server control such as a DataGrid.	Boolean
Hide	Used to hide output of collection in design time and run time.	Boolean
	<p>True = Hide collection</p> <p>False = Display collection</p>	
Language	<p>Design time only: View content blocks, collections, etc. in different languages.</p> <hr/> <p>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</p> <hr/>	Integer
Random	Set to True if you want to randomly display one collection item. The item changes each time a user views the page.	Boolean

Property	Description	Data Type
WrapTag	<p>Allows a developer to specify a server control's tag.</p> <p>The default is Span.</p> <p>Span - The tag is used to designate an inline portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Example of ecmNavigation Display

The following is an example of a collection being used as a navigation menu.



Example of ecmTeaser Display

The following is an example of a collection using the ecmTeaser display style.

Employment Opportunities

[Plastic Molder #123](#)

RC International is looking for an experienced plastics molder.

[Servo Control Engineer #124](#)

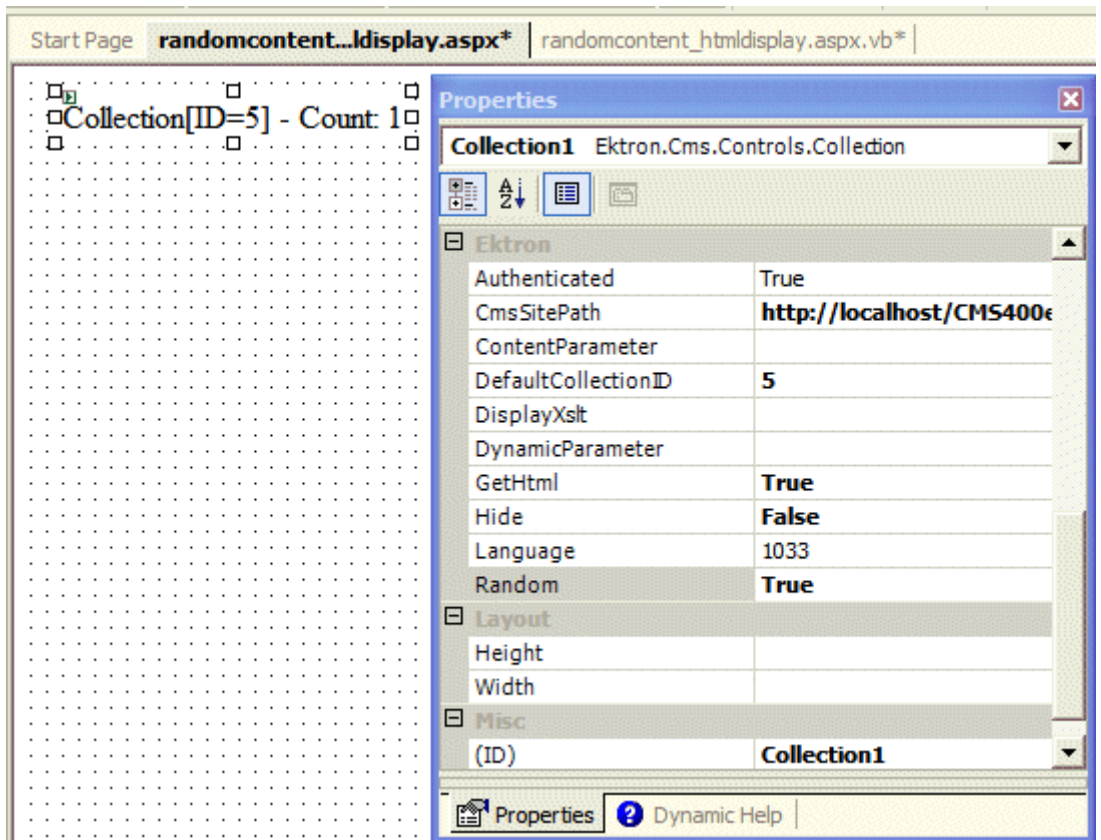
RC International is looked for a highly skilled servo control engineer to join our team.

Using the Collection Server Control Programmatically Example

The following is an example of using code behind, and a drag and drop Collection Server Control to display random content from a collection.

1. Drag and Drop a Collection Server Control on your Web form.
2. Set the properties in the properties window.

NOTE [In this example, the Random property and the GetHtml property must be set to True.](#)



3. Add the following code to the code behind.

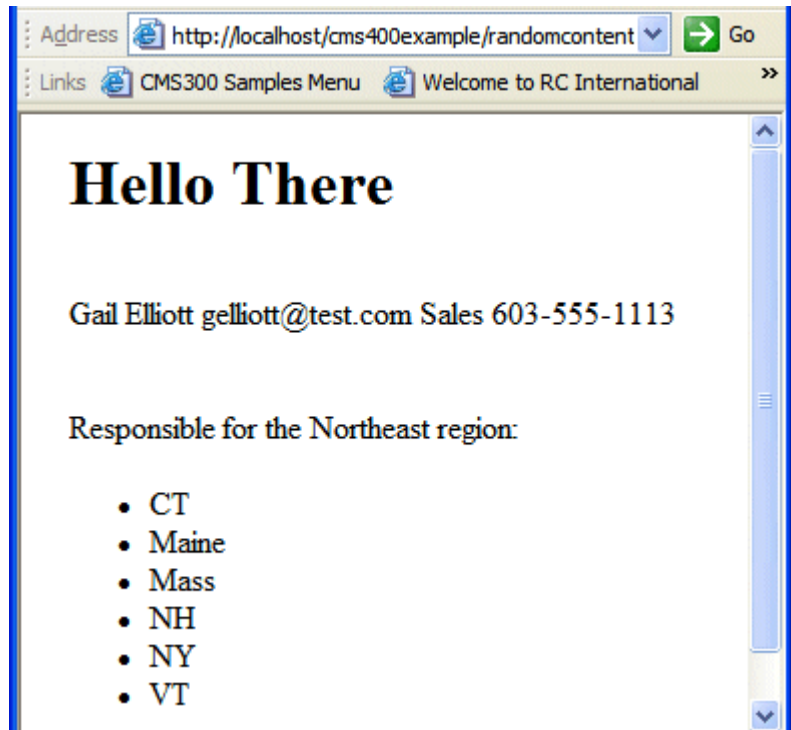
```
Dim str As String

str = "<h1>Hello There</h1><br>"

str &= Collection1.EkItems(0).Html

Collection1.Text = str
```

4. Build and browse your web form.



Calendar Server Control

The Calendar Server Control displays a calendar that was created in the Workarea. By using the Calendar Server Control's properties, you can select multiple views of the calendar to display. See Also: "[DisplayType](#)" on page 95

You can adjust a calendar's look and feel by using a style sheet. See Also: "[Calendar Style Sheet](#)" on page 97.

The Calendar Server Control can output the raw XML data of the calendar. You can use this XML data to create a completely customized calendar.

When added to a template and visited, the Month view calendar display looks like this.

November			December 2004				January
Sun	Mon	Tue	Wed	Thu	Fri	Sat	
			1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16	17	18	
19	20	21	22	23	24	25	
26	27	28	29	30	31		

A calendar can be *static* or *dynamic*.

- A static calendar is selected in the server control properties.
- A dynamic calendar is passed to the page by a URL parameter.

This subsection contains the following topics:

- ["Calendar Server Control Properties" on page 93](#)
- ["Calendar Style Sheet" on page 97](#)
- ["Event Calendar Sample" on page 97](#)

Calendar Server Control Properties


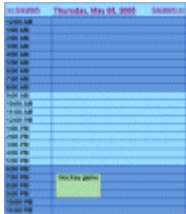
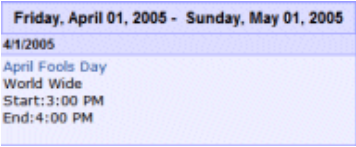


The Calendar Server Control properties are described in this table.

NOTE There are additional settings to help define the calendar in the Ektron CMS400.NET Workarea, these include display weekends, forward only, show all label, etc. For information on these properties, see Ektron CMS400.NET

Administrator's Manual > Calendars > Managing Calendar Modules > Viewing Calendar Modules.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10</i>	String
DefaultCalendarID	The ID number of the calendar that appears where you inserted this server control. If you don't know the ID number of the calendar, use the CMS Explorer to browse to it. See <i>Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10</i>	Integer

Property	Value	Data Type
DisplayType	<p>Specify the Calendar view type. The default is Month.</p> <ul style="list-style-type: none"> Month - A 30 day calendar month view.  Day - Single day view.  Event - Display events for the current day.  Inline - Display events that occur from the start date of the calendar through the end date.  MonthUpWithEvent - Displays a month calendar, events appear to the right of the calendar when you click on a day.  	String

Property	Value	Data Type
DynamicParameter	To make this calendar dynamic, select calendar_id . When you do, this server control uses the calendar passed as a URL parameter.	String
EndDate	Specify the end date the calendar displays. Used in conjunction with the StartDate property. For example you have a calendar that shows only classes for the next three month semester	Date
Hide	Used to hide output of calendar in design time and run time. True = Hide calendar False = Display calendar	Boolean
Language	Design time only: View the calendar. in selected languages. <u>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</u>	Integer
ShowXML	At runtime, output the raw XML data instead of the calendar. With this XML data you can apply your own XSL to create your own custom display formats Default is False . True = Show raw XML data False = Show Calendar	Boolean
StartDate	Specify the start date the calendar displays. Used in conjunction with the EndDate property. For example you have a calendar that shows only classes for the next three month semester.	Date
WrapTag	Allows a developer to specify a server control's tag. The default is Span . Span - The tag is used to designate an inline portion of an HTML document as a span element. Div - The <div> tag is used when you want to apply attributes to a block of code. Custom - Allows you to use a custom tag.	String

Calendar Style Sheet

Before you insert a Calendar into your Ektron CMS400.NET template, Ektron recommends referencing the style sheet `\webroot\cms400example\cms400scripts\csslib\calendarStyles.css` in your template. Copy the style sheet to your webroot folder then add it to your project. Next create a reference to `calendarStyles.css`. For example:

```
html>  
  
<head>  
  
    <title>Company Calendar</title>  
  
    <link href="calendarStyles.css" rel="stylesheet" type="text/  
css" />  
  
</head>
```

This can also be accomplished via drag and drop. If you drag and drop the `calendarStyles.css` from the Solution Explorer to your Web form, the reference is created in the HTML automatically.

Event Calendar Sample

Below is a Calendar displayed on the Web Page. When logged in, you can add and view calendar events. When logged out, calendar events only appear.

Login Status	Calendar display																																										
Logged in	<div style="text-align: center;"> <p>November December 2004 January</p> <table border="1"> <thead> <tr> <th>Sun</th> <th>Mon</th> <th>Tue</th> <th>Wed</th> <th>Thu</th> <th>Fri</th> <th>Sat</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td style="border: 2px solid red;">1</td> <td>2</td> <td>3</td> <td>4</td> </tr> <tr> <td></td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> <td>11</td> </tr> <tr> <td></td> <td>13</td> <td>14</td> <td>15</td> <td>16</td> <td>17</td> <td>18</td> </tr> <tr> <td></td> <td>20</td> <td>21</td> <td>22</td> <td>23</td> <td>24</td> <td>25</td> </tr> <tr> <td></td> <td>27</td> <td>28</td> <td>29</td> <td>30</td> <td>31</td> <td></td> </tr> </tbody> </table> </div>	Sun	Mon	Tue	Wed	Thu	Fri	Sat				1	2	3	4		6	7	8	9	10	11		13	14	15	16	17	18		20	21	22	23	24	25		27	28	29	30	31	
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																					
			1	2	3	4																																					
	6	7	8	9	10	11																																					
	13	14	15	16	17	18																																					
	20	21	22	23	24	25																																					
	27	28	29	30	31																																						

Login Status	Calendar display																																										
Logged Out	<div style="text-align: center;"> <p>November December 2004 January</p> <table border="1"> <thead> <tr> <th>Sun</th> <th>Mon</th> <th>Tue</th> <th>Wed</th> <th>Thu</th> <th>Fri</th> <th>Sat</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td style="border: 2px solid red;">1</td> <td>2</td> <td>3</td> <td>4</td> </tr> <tr> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td></td> </tr> <tr> <td>13</td> <td>14</td> <td>15</td> <td>16</td> <td>17</td> <td>18</td> <td></td> </tr> <tr> <td>20</td> <td>21</td> <td>22</td> <td>23</td> <td>24</td> <td>25</td> <td></td> </tr> <tr> <td>27</td> <td>28</td> <td>29</td> <td>30</td> <td>31</td> <td></td> <td></td> </tr> </tbody> </table> </div>	Sun	Mon	Tue	Wed	Thu	Fri	Sat				1	2	3	4	6	7	8	9	10	11		13	14	15	16	17	18		20	21	22	23	24	25		27	28	29	30	31		
Sun	Mon	Tue	Wed	Thu	Fri	Sat																																					
			1	2	3	4																																					
6	7	8	9	10	11																																						
13	14	15	16	17	18																																						
20	21	22	23	24	25																																						
27	28	29	30	31																																							

Menu Server Control

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 property. Below is a menu display with the SampleMenu Xslt.

- Products
 - RC Cars
 - [RC Cheetah](#)
 - [RC Sportster](#)
 - RC Planes
 - [RC Lilly](#)
 - [RC Redstar](#)
 - [Products Page](#)
 - [Visit Ektron.com](#)

Before you can use the Menu Server Control, you must create a menu in the **Ektron CMS400.NET** work area. To learn about creating menus, see “Working with Menus” in the **Ektron CMS400.NET** User Manual.

This subsection contains the following topics:

- “Menu Server Control vs. DhtmlMenu Server Control” on page 100
- “Menu Server Control Properties” on page 101
- “Using DisplayXslt Samples” on page 102

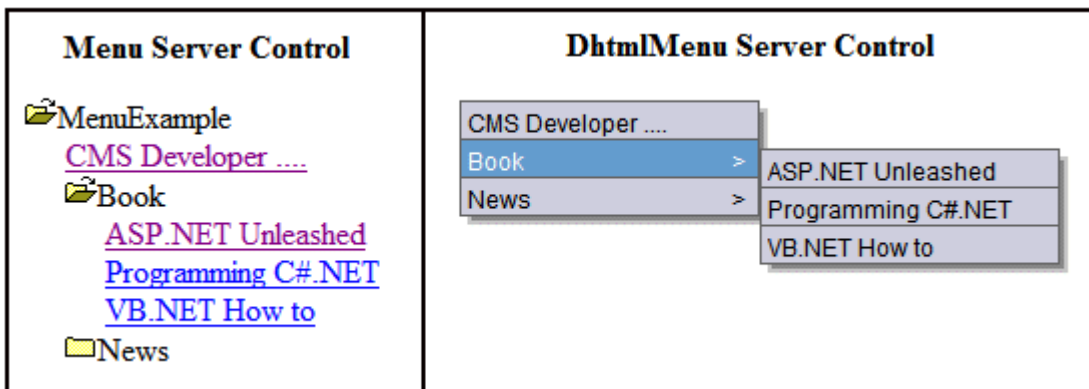
Menu Server Control vs. DhtmlMenu Server Control

The Menu Server Control calls a menu to be displayed on a page. You can manipulate the menu by using the DisplayXslt property.

With the DhtmlMenu Server Control, you display a dynamic menu with drop down boxes and submenus. You manipulate DhtmlMenus with a style sheet. DhtmlMenus save space on the Web page by having the drop down menus and submenus float over content.

WARNING!

You cannot create a DHTML menu using the code behind -- you must drag and drop it. Once the menu is on the template, you can manipulate its properties using the code behind.



Menu Server Control Properties

The Menu Server Control properties are described in this table.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. <i>See Also:</i> "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
DefaultMenuID	The ID of a menu that appears where you insert this server control if no other menu is identified or available. If you don't know the ID number of the menu, use the CMS Explorer to browse to it. <i>See Also:</i> "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	Integer
DisplayXslt	The Xslt to use to render the menu. None - databind only SampleMenu - A sample display, formatted as a bulleted menu list TreeMenu - A sample display, formatted as a folder tree. You can expand the tree by clicking on the folder icon. Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page For more information on using the SampleMenu and TreeMenu DisplayXslt, see "Using DisplayXslt Samples" on page 102.	String
DynamicParameter	Select menu_id . When you do, this server control uses the menu passed as a URL parameter.	String
Hide	Used to hide a menu in design time and run time. True = Hide menu False = Show menu	Boolean

Property	Value	Data Type
Language	<p>Design time only: View content blocks, collections, etc. in different languages</p> <hr/> <p>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</p> <hr/>	Integer
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an inline portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Using DisplayXslt Samples

Two DisplayXslt samples are provided with the Menu Server Control, SampleMenu and TreeMenu. This section explains how to use them.

SampleMenu

The SampleMenu DisplayXslt lets you display an **Ektron CMS400.NET** menu as a bulleted item list. See below for an example.

- MenuExample
 - [CMS Developer](#)
 - Book
 - [ASP.NET Unleashed](#)
 - [Programming C#.NET](#)
 - [VB.NET How to](#)
 - News
 - [“All-Stars” Customer](#)
 - [Web Design Firms](#)
 - [Enhanced Workflow](#)
 - [Visual Rapid CMS Integration](#)
 - [Ektron Partners and Customers](#)

When you are logged in to your **Ektron CMS400.NET** site, two menu items are added to each section of the menu: Add and Edit Menu. The user can use these to add a new menu item or edit an existing one. See example below.

For Information on a adding a menu Item, see **Ektron CMS400.NET** Users Manual > “Working with Menus” > “Managing Menus” > “Adding a New Content Block to a Menu via Navigation Link on a Web Page”.

For Information on a editing a menu, see **Ektron CMS400.NET** Users Manual > “Working with Menus” > “Managing Menus” > “Editing a Menu via Navigation Link on a Web Page”.

- MenuExample
 - [CMS Developer](#)
 - Book
 - [ASP.NET Unleashed](#)
 - [Programming C#.NET](#)
 - [VB.NET How to](#)
 - [Add](#) Add or Edit submenu
 - [Edit Menu](#) submenu
 - News
 - [“All-Stars” Customer](#)
 - [Web Design Firms](#)
 - [Enhanced Workflow](#)
 - [Visual Rapid CMS Integration](#)
 - [Ektron Partners and Customers](#)
 - [Add](#) Add or Edit submenu
 - [Edit Menu](#) submenu
 - [Add](#) Add or Edit whole menu
 - [Edit Menu](#) whole menu

TreeMenu

The TreeMenu DisplayXslt lets you display an **Ektron CMS400.NET** menu as a clickable folder list. You can expand and collapse the menu by clicking on the folder icons. See below for an example.

- 📁 MenuExample
 - [CMS Developer](#)
 - 📁 Book
 - 📁 News
 - [“All-Stars” Customer](#)
 - [Web Design Firms](#)
 - [Enhanced Workflow](#)
 - [Visual Rapid CMS Integration](#)
 - [Ektron Partners and Customers](#)

When you are logged in to your **Ektron CMS400.NET** site, two menu items are added to each section of the menu: Add and Edit Menu.

The user can use these to add a new menu item, or edit an existing menu item. See example below.

For Information on a adding a menu Item, see **Ektron CMS400.NET Users Manual** > “Working with Menus” > “Managing Menus” > “Adding a New Content Block to a Menu via Navigation Link on a Web Page”.

For Information on a editing a menu, see **Ektron CMS400.NET Users Manual** > “Working with Menus” > “Managing Menus” > “Editing a Menu via Navigation Link on a Web Page”.



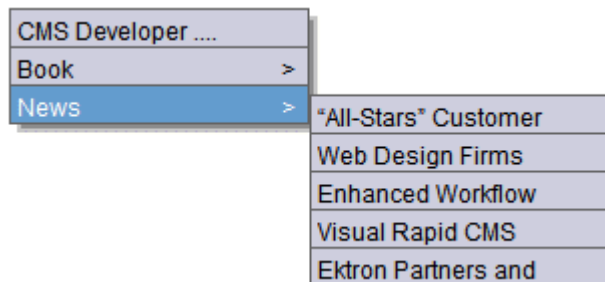
NOTE When using the TreeMenu Server Control with Visual Studio and Windows 2003, the folder images do not display during design-time. They do, however, display correctly during run-time.



DhtmlMenu Server Control

NOTE For optimal performance, Ektron recommends using at minimum Internet Explorer 5.5, Netscape 6.0, or Mozilla Firefox 1.0.

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 of space. You can modify the menu's style via a style sheet.

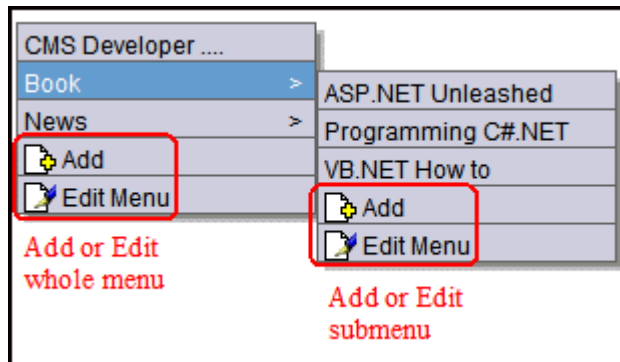


Before using the DhtmlMenu Server Control, you must add items to a menu in the Ektron CMS400.NET workarea. To learn about creating menus, see “Working with Menus” in the CMS400.NET User Manual.

The DhtmlMenu Server Control also lets developers add content to a menu. When you are logged in to your **Ektron CMS400.NET** site, two menu items are added to each menu: Add and Edit Menu. The user can use these to add a new menu item or edit an existing menu item. See example below.

For Information on a adding a menu Item, see **Ektron CMS400.NET** Users Manual > “Working with Menus” > “Managing Menus” > “Adding a New Content Block to a Menu via Navigation Link on a Web Page”.

For Information on a editing a menu, see **Ektron CMS400.NET** Users Manual > “Working with Menus” > “Managing Menus” > “Editing a Menu via Navigation Link on a Web Page”.



This subsection contains the following topics:

- “Menu Server Control vs. DhtmlMenu Server Control” on page 108
- “DhtmlMenu Server Control Properties” on page 108
- “Code Behind Only Properties” on page 112
- “Using a Style Sheet” on page 113

Menu Server Control vs. DhtmlMenu Server Control

See ["Menu Server Control vs. DhtmlMenu Server Control"](#) on page 100.

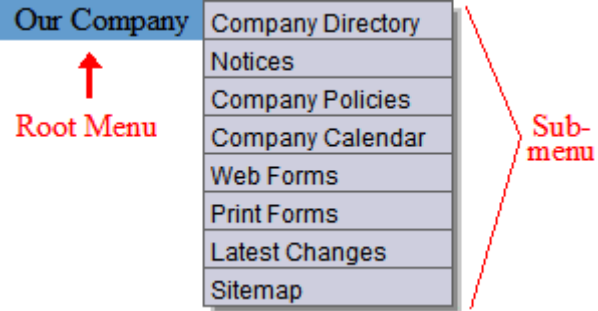
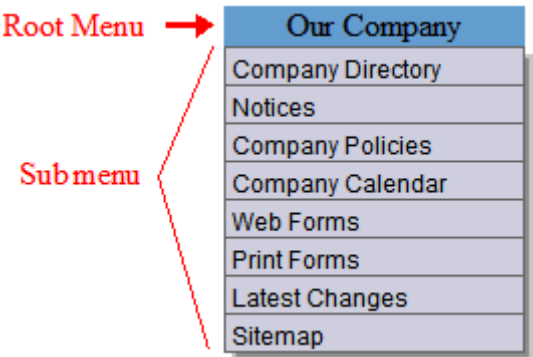
DhtmlMenu Server Control Properties

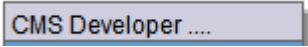
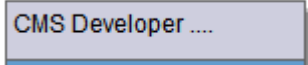
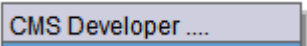
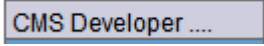
The DhtmlMenu Server Control properties are described in this table.

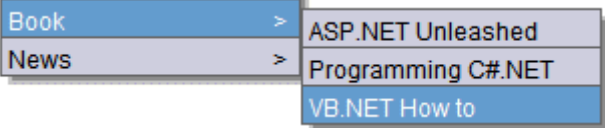
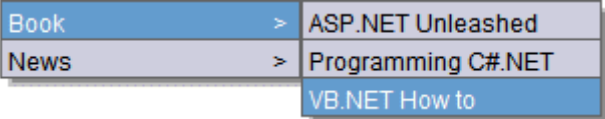
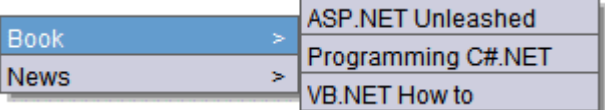
NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Value	Data Type
Align	Specifies the horizontal alignment of the root menu text. The default is Center . NotSet - Use the default setting. Left - Aligns text to the left. Center - Aligns text to the center. Right - Aligns text to the right. Justify - Aligns text justified to both margins.	String
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc... See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
BackColor	Sets the background color for the root menu. This property appears in the Appearance category.	String

Property	Value	Data Type				
CssClass	<p>Apply a .css class to the root menu. Enter the class name from the .css defined in the Stylesheet property. See Also: "Stylesheet" on page 111</p> <p>Since pop_style.css is the default style sheet, you can use it's classes without defining it in the Stylesheet property.</p> <p>For example enter lowtext in the property, the root menu changes as shown below.</p> <table border="1" data-bbox="491 487 1099 626"> <tr> <td>No CssClass applied</td> <td>lowText CssClass applied</td> </tr> <tr> <td>MenuExample</td> <td>MenuExample</td> </tr> </table>	No CssClass applied	lowText CssClass applied	MenuExample	MenuExample	String
No CssClass applied	lowText CssClass applied					
MenuExample	MenuExample					
DefaultMenuID	<p>The ID of a menu that appears where you inserted this server control if no other content block is identified, or is not available.</p> <p>If you don't know the ID number of the menu, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10</p>	Integer				
DynamicParameter	Select menu_id . When you do, this server control uses the menu passed as a URL parameter.	String				
ForeColor	Sets the color of the font for the root menu. This property appears in the Appearance category.					
Hide	Used to hide a menu in design time and run time. True = Hide menu False = Show menu	Boolean				

Property	Value	Data Type
HorizontalDisplay	<p>Displays submenu to the right of the root menu. This only affects the first submenu of the root menu. All subsequent submenus appear to the right of each submenu.</p> <p>The default is False</p> <p>True = Submenu appears to the right of the root menu</p>  <p>False = Submenu appears below the root menu</p> 	Boolean
Language	<p>Design time only: View content blocks, collections, etc. in different languages</p> <hr/> <p>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</p> <hr/>	Integer
MenuHoverOverColor	Sets the background color for the root menu when hovering over it.	String

Property	Value	Data Type
MenuStyle	<p>A specified inline style that defines the root menu. For example enter the following in the MenuStyle property box:</p> <pre>font: bold 18px Arial;color:royalblue; background-color:slateblue</pre> <p style="text-align: center;">Our Company</p>	String
Stylesheet	<p>Specify the location and style sheet for the submenu. For example: <code>\cms400scripts\csslib\mytest.css</code>. Leave blank to use the default.</p> <p><code>pop_style.css</code> is the default style sheet. It is located in <code>\webroot\CMS400Example\cms400scripts\csslib</code>. For more information on style sheets, see "Using a Style Sheet" on page 113</p> <p>The Stylesheet property can be used in conjunction with the <code>CssClass</code> property to define the root menu. See Also: "CssClass" on page 109</p>	String
SubMenuBackColor	Sets the background color for the submenu.	String
SubMenuHoverOverColor	Sets the background color for the submenu when hovering over it.	String
SubMenuItemHeight	<p>Sets the height of each submenu item in pixels. The Default is 20</p> <p>Example set to 20: </p> <p>Example set to 30: </p>	Integer
SubMenuItemWidth	<p>Sets the width of each submenu item in pixels. The Default is 150</p> <p>Example set to 150: </p> <p>Example set to 130: </p>	Integer

Property	Value	Data Type
SubMenuTopOffset	<p>Sets the location of the top of the submenu. This number is in pixels. The default is 5</p> <p>Example set to 5:</p>  <p>Example set to 0 (zero):</p>  <p>Example set to -10:</p> 	Integer
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an inline portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Code Behind Only Properties

The following DhtmlMenu specific properties can only be used programmatically, in code behind, to set the absolute location of the DhtmlMenu.

Property	Value	Data Type
RootMenuLeftOffset	The location, from the left border of the window, that the menu will appear. This number is in pixels.	Integer

Property	Value	Data Type
RootMenuTopOffset	The location, from the top border of the window, that the menu will appear. This number is in pixels.	Integer

Using a Style Sheet

You can create your own style sheet, or modify the existing default one, `pop_style.css`. It is located in
`\webroot\CMS400Example\CMS400scripts\csslib`.

Style Sheets let you choose different text colors, text alignment, text fonts, and text size to create a customized menu.

Best Practice

Ektron recommends copying and renaming the default style sheet. This insures you always have a clean file to start with. This also gives you something to reference if you are not getting the results you expect.

The style sheet must be located in the site root.

You can only define one menu style sheet for each Web page. If you have more than one menu on a single Web page, they share the style sheet.

The following is a list of style sheet properties that can be modified in the `pop_style.css`.

Style Class	Description	Attribute	Default Value
.lowText	Text of the floating menu as displayed on the menu.	<ul style="list-style-type: none"> • color • font • text-align • text-decoration 	<ul style="list-style-type: none"> • #000000 (Black) • 12px Arial, sans-serif • left • none

Style Class	Description	Attribute	Default Value
.highText	Text of the floating menu when highlighted.	<ul style="list-style-type: none"> • color • font • text-align • text-decoration 	<ul style="list-style-type: none"> • #000000 (Black) • 12px Arial, sans-serif • left • none
.roothighText <u>Note: This class only works when called by the <code>CssClass</code> property. Use the <code>Font</code> property, in the Appearance category to define root menu text.</u>	Text of the root menu as displayed on a Web page.	<ul style="list-style-type: none"> • color • font • text-align • text-decoration 	<ul style="list-style-type: none"> • #000000 (Black) • 16px Arial, sans-serif • center • none
.itemBorder	Border color and size of the menu.	<ul style="list-style-type: none"> • border 	<ul style="list-style-type: none"> • 1px solid #666666
.itemBorderBlank	Separator border color and size of the menu.	<ul style="list-style-type: none"> • border 	<ul style="list-style-type: none"> • 1px solid #CCCCDD

For more information on style sheets, visit <http://www.w3.org/Style/CSS/>.

MetaData Server Control

The Metadata Server Control lets you add the metadata from content blocks to a Web page. This will let developers add metadata quickly without having to type it in. You can add metadata from a single content block, multiple content blocks, or by dynamically passing a content ID from a URL.

This subsection contains the following topics:

- ["MetaData Server Control vs. MetaDataList Server Control" on page 115](#)
- ["MetaData Server Control Properties" on page 115](#)

- ["Using the MetaData Server Control" on page 116](#)


MetaData Server Control vs. MetaDataList Server Control

With the MetaData Server Control, you add metadata from content blocks to your Web page. With the MetaDataList Server Control, you create a list of content blocks to display on your site, based on the Metadata in each content block. For Information on the MetaDataList Server Control, see ["MetaDataList Server Control" on page 117](#).

MetaData Server Control Properties

The Metadata Server Control properties are described in this table.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
DefaultContentID	The ID of the content block that the server control will get the metadata from. If you want to add metadata from several content blocks, set this property to zero (0) and use the DefaultItemList property to identify the content blocks. If you don't know the ID number of the content block, use the CMS Explorer to browse to it. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	Integer
DefaultItemList	A comma delimited list of content block IDs to get metadata from. DefaultContentID must be set to zero (0) so you can add IDs to the list. 	Integer

Property	Value	Data Type
DynamicParameter	Gets or sets the QueryString parameter to read a content ID dynamically.	String
Hide	Used to hide a Metadata Server Control in design time and run time. True = Hide Metadata Server Control False = Show Metadata Server Control	boolean

Using the MetaData Server Control

These steps show how to use the MetaData Server Control.

1. Drag a MetaData Server Control into a template.
2. Set the properties of the Metadata Server Control. This will create the following HTML in the HTML body.

```
<cms:MetaData id="MetaData1" runat="server" DefaultContentID="12"></cms:MetaData>
```

Or if you are using multiple content block IDs in the DefaultItemList, the following HTML will be created.

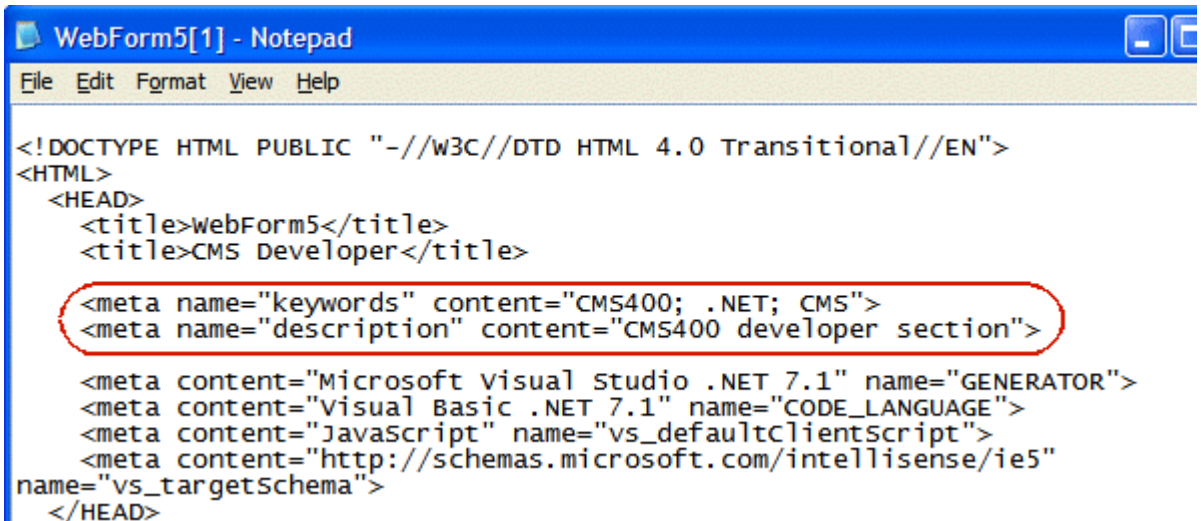
```
<cms:metadata id="MetaData1" runat="server" DefaultItemList="[12,7,4]"></cms:metadata>
```

3. Click on the HTML tab and copy that line from the <body> tag into the <head> tag.

```
<HEAD>
  <title>WebForm5</title>
  <cms:MetaData id="Metadata2" runat="server" DefaultContentID="12"></cms:MetaData>
  <meta name="GENERATOR" content="Microsoft Visual Studio .NET 7.1">
  <meta name="CODE_LANGUAGE" content="Visual Basic .NET 7.1">
  <meta name="vs_defaultClientScript" content="JavaScript">
  <meta name="vs_targetSchema" content="http://schemas.microsoft.com/intellisense/i
</HEAD>
<body>
  <form id="Form1" method="post" runat="server">
  </form>
</body>
```

4. Save the Web form and rebuild the solution.
5. Open the Web page in the browser.
6. Right click on the Web page and click **View Source**. Look in the head tag. The meta tags from the content block are added to the page.

The metadata information added is shown below.



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <title>webForm5</title>
    <title>CMS Developer</title>
    <meta name="keywords" content="CMS400; .NET; CMS">
    <meta name="description" content="CMS400 developer section">
    <meta content="Microsoft Visual Studio .NET 7.1" name="GENERATOR">
    <meta content="Visual Basic .NET 7.1" name="CODE_LANGUAGE">
    <meta content="JavaScript" name="vs_defaultClientScript">
    <meta content="http://schemas.microsoft.com/intellisense/ie5"
name="vs_targetSchema">
  </HEAD>
```

MetaDataList Server Control

Use the MetaDataList Server Control to create lists based on Keyword Names and Keyword Values contained within the metadata of a content block. In the example below the KeyWordNames are circled in red and the KeyWordValues are circled in green.

Title: [English (U.S.)]

Content **Summary** **Metadata** **Schedule** **Comment**

* = Required fields

Title: (500 max.) Separator Character ";"

486 characters left

Keywords: (500 max.) Separator Character ";"

The list can display the information as a list of hyperlinks. You can choose, based on properties you set, if you would like to have a summary displayed. You can also choose how to order the display. For information on adding Metadata to a content block, see the Users Manual > "Adding or Editing Metadata".

This subsection contains the following topics:

- ["MetaData Server Control vs. MetaDataList Server Control" on page 118](#)
- ["MetaDataList Server Control Properties" on page 118](#)

MetaData Server Control vs. MetaDataList Server Control

With the MetaData Server Control, you add metadata from content blocks to your Web page. With the MetaDataList Server Control, you create a list of content blocks to display on your site based on the Metadata in each content block. For Information on the MetaData Server Control, see ["MetaData Server Control" on page 114](#)

MetaDataList Server Control Properties

The MetaDataList Server Control properties are described in this table.

NOTE The following table only lists Ektron-specific properties. It does not describe native .Net properties such as font, height, width and border style. For documentation of these properties, see VisualStudio.Net help.

Property	Value	Data Type
Authenticated	Indicates if you are logged in to the CMS Explorer and can use it to browse to Content, Collections, etc. See Also: "Using CMS Explorer to Browse Your Ektron CMS400.NET Site" on page 10	String
ContentType	One of the following <ul style="list-style-type: none"> • all types • archived content • archived forms • content • forms To learn about archived content, see the Ektron CMS400.NET User Manual, "Scheduling Content to Begin and End" > "Setting Archive Options."	String
DisplayXslt	Determines how the information on the page is displayed None -databind only ecmNavigation - lists the title of every content block in the folder See Also: "Example of ecmNavigation Display" on page 89 ecmTeaser - lists the title of every content block in the folder plus the content summary. See Also: "Example of ecmTeaser Display" on page 89 Path to Custom Xslt - Enter the path to an Xslt that determines the display of the page	String

Property	Value	Data Type
ExactPhrase	<p>Determines whether the KeyWordValue needs to match the phrase exactly.</p> <p>For example, if “site” is the KeyWordValue and the title of a content block is “Welcome to the site” and ExactPhrase is set to true, you would not see the content block in the metadata list. This is because “site” does not equal “Welcome to the site”.</p> <p>True = Match the exact phrase False = Doesn't need to match exact phrase</p>	Boolean
FolderID	The root folder content will be pulled from.	Integer
GetHtml	Set to True if you want to retrieve and display content (html body) for all content blocks in the collection. For example, display content inside a web server control such as a DataGrid.	Boolean
Hide	<p>Used to hide a metadata list in design time and run time.</p> <p>True = Hide metadata list False = Show metadata list</p>	Boolean
KeywordName	<p>The KeywordName is the container where the KeywordValues are located.</p> <p>Some examples of a KeywordName might be “Keywords or Title”</p> <p>For information on creating KeywordNames for Metadata, see the Ektron CMS400.NET Administrators Manual > “Managing Content” > “Metadata” > “Managing Metadata Definitions” > “Adding a Metadata Definitions”</p>	String
KeywordValue	<p>The value that is associated with the KeywordName</p> <p>Some examples of a KeywordValue might be “Home; page; or company”</p>	String

Property	Value	Data Type
Language	<p>Design time only: View content blocks, collections, etc. in different languages.</p> <hr/> <p>Note: The language property shows results only in design-time (in Visual Studio) and does not affect the site. Content on the site is managed by a cookie value that is set by selecting a language from the dropdown list.</p> <hr/>	Integer
MaxNumber	The maximum number of results returned for display. 0 = unlimited	Integer
OrderBy	<p>The order of the list to be returned.</p> <ul style="list-style-type: none"> • Title - The title of the content block • ID - The content block ID number • Date Created - The date the content block was created • Date Modified - The date the content block was last modified • LastEditorLname - The last editor's last name • LastEditorFname - The last editor's first name 	String
Recursive	Whether to search sub-folders of the identified root folder. The root folder is identified in the Folderpath property.	Boolean
SortOrder	Ascending or Descending	String
WrapTag	<p>Allows a developer to specify a server control's tag. The default is Span.</p> <p>Span - The tag is used to designate an inline portion of an HTML document as a span element.</p> <p>Div - The <div> tag is used when you want to apply attributes to a block of code.</p> <p>Custom - Allows you to use a custom tag.</p>	String

Data Binding with Server Controls

With data binding, you can bind Ektron server controls to a DataGrid Control, DataList Control, or Repeater Control. This gives more flexibility when you use data from the Ektron server controls. Some benefits of data binding are ease of data manipulation and the ability to format data.

Title	ID Teaser
Plastic Molder #123	13 RC International is looking for an experienced plastics molder.
Servo Control Engineer #124	14 RC International is looked for a highly skilled servo control eng

The following are data bindable Ektron Server Controls.

- Collection
- Content Block
- Form Block
- List Summary
- Search

The following is an example of code behind that uses a DataGrid to display a Collection:

```
Dim myCol As New Ektron.Cms.Controls.Collection
myCol.DefaultCollectionID = 1
myCol.Fill()
DataGrid1.DataSource = myCol
DataGrid1.DataBind()
```

NOTE When using drag and drop server controls for the Collection and DataGrid, you need to call the DataBind() method in the code behind. An example line of code would be: `DataGrid1.DataBind()` .

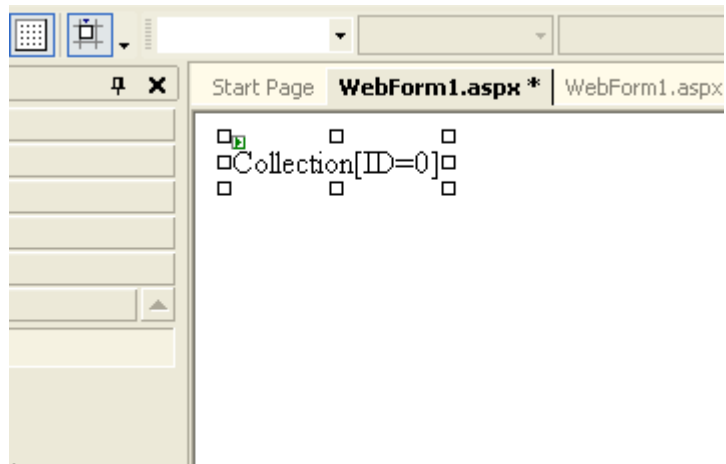
This subsection contains the following topics:

- "Steps to Data Binding Using Drag and Drop Server Controls" on page 123
- "Customizing the DataGrid Table View" on page 125

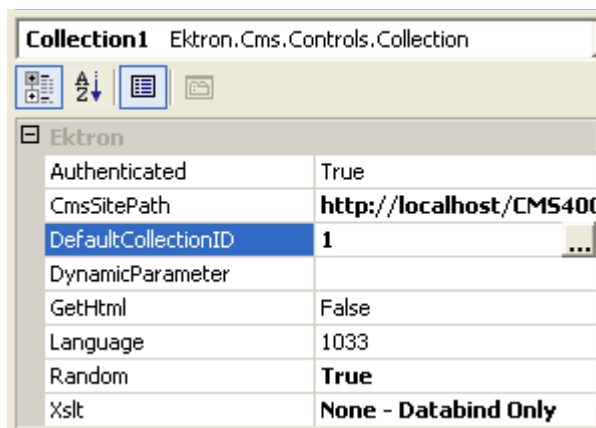
Steps to Data Binding Using Drag and Drop Server Controls

The following example uses a Collection.

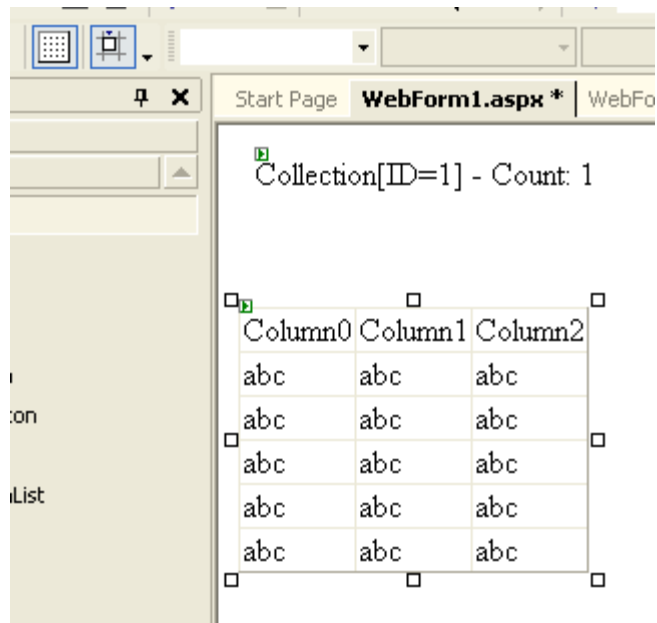
1. Drag and drop a data bindable Server Control on the Web form.



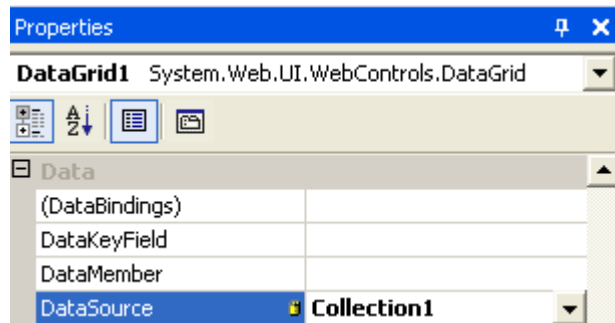
2. In properties, choose the Ektron default ID for the Server Control and make sure the XSLT is set to **None- DataBind Only**.



3. Drag and drop a DataGrid on the web form.



4. In the DataGrid properties, choose the DataSource.



5. The DataGrid is populated with column headers and the type of information that will be in the table.
6. Go to code behind and add `DataGrid1.DataBind()` to the `Page_Load` event.

```
Private Sub Page_Load(ByVal sender As System.Object
    'Put user code to initialize the page here
    DataGrid1.DataBind()
End Sub
```

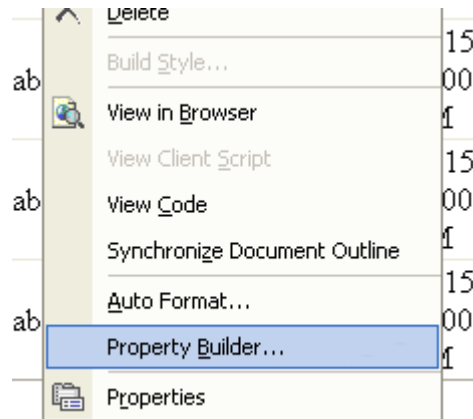

7. Switch to Design mode and right-click on the DataGrid. Then, choose **View in Browser**. The table appears.

LastEditorFname	DateModified	Html	QuickLink	HasErrors	Teaser	ID
Application	2/17/2004 4:12:11 PM		/CMS400Sample/products.aspx?id=5	False	RC Internationals most successful entry in to the competitive RC racing world is the RC Cheetah.	5

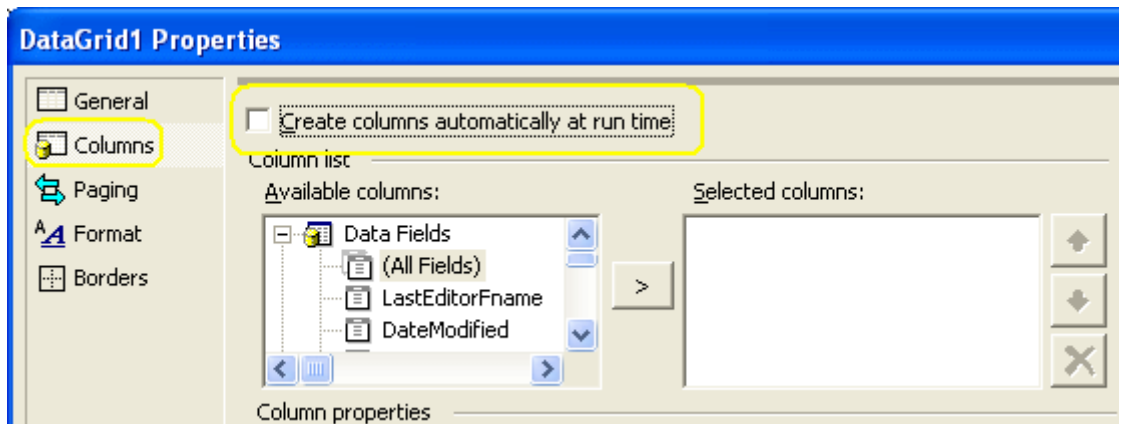
Customizing the DataGrid Table View

Customizing the DataGrid is a great way to display and format specific information from a data source. You can select formatting for columns, paging and borders, plus other general formatting for the table. In the following example, we choose table columns.

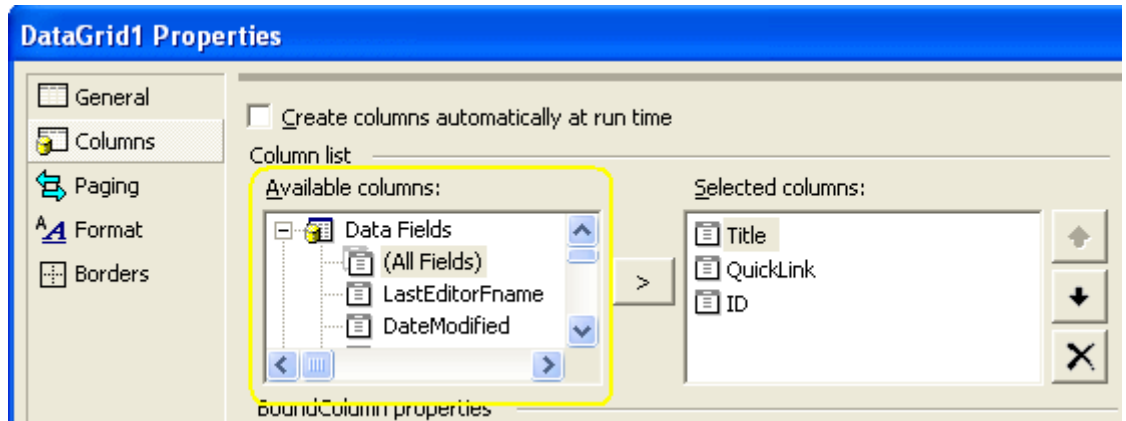
1. In Web form design, right click the DataGrid and choose **Property Builder**.



2. Click **Columns** and uncheck the **Create columns automatically at run time** checkbox.



3. Choose the fields for the custom table from the available columns.



4. Click **OK**.
5. Right click the DataGrid and choose **View in Browser**.
6. The custom table appears.

Title	QuickLink	ID
RC Lilly /CMS400Sample/products.aspx?id=6		6

For more information on working with DataBind() or DataGrid, review the help topics in Visual Studio.

Caching with Server Controls

One of the most important factors in building high-performance, scalable Web applications is the ability to store items in memory the first time they are requested. These items include data objects, pages, or parts of a page.

Known as caching, this process allows you to use a number of techniques to store page output or application data across HTTP requests and reuse it. You can store items on the Web server or other software in the request stream, such as the proxy server or browser.

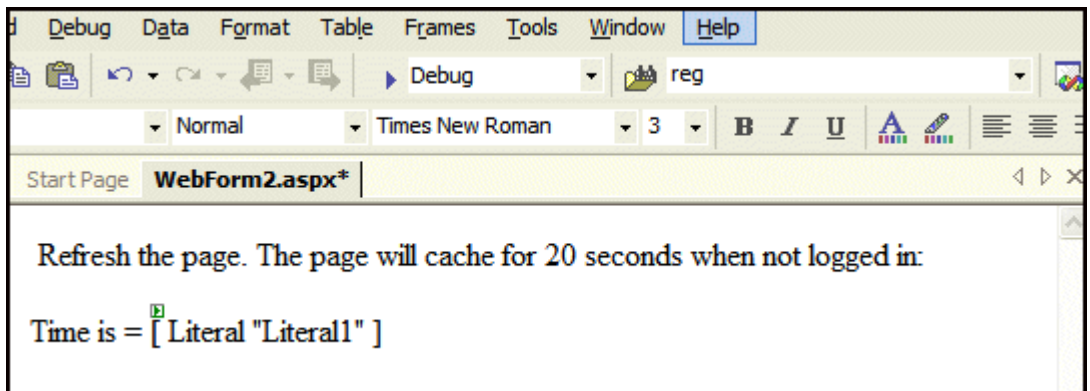
With caching, the server does not have to recreate information, particularly information that demands significant processor time or other resources. This saves valuable time and resources.

Below is an example of a page level cache for non logged in users. In this example, you use a Server Control to define whether a user is logged in or not. Then, further define if the time is cached, based on the user's status.

1. Create a new Web form in your Ektron CMS400.NET project.
2. Add the following text to the Web form:
Refresh the page. The page will cache for 20 seconds when not logged in:

Time is =

3. Next to **Time is =**, add a literal.



4. Below that, add a ContentBlock Server Control.
5. Set the DefaultContentID. For this example, DefaultContentID = 20.
6. Add the following code to the code behind.

```
If Not ContentBlock1.IsLoggedIn Then
```

```
    Response.Cache.SetExpires(DateTime.Now.AddSeconds(20))
```

```
    Response.Cache.SetCacheability(HttpCacheability.Public)
```

```
    Response.Cache.SetValidUntilExpires(True)
```

```
    Response.Cache.VaryByParams("id") = True
```

```
    Response.Cache.SetVaryByCustom("cmsCache")
```

```
End If
```

```
Literal1.Text = Now()
```

7. Build your Web form.
8. Browse to your Web form with your browser.

9. Press the refresh button on your browser.

If you are not logged in to CMS400.NET, the time remains the same for twenty seconds. After twenty seconds, when you refresh, you get the new time.

Standard ASP.NET Methods

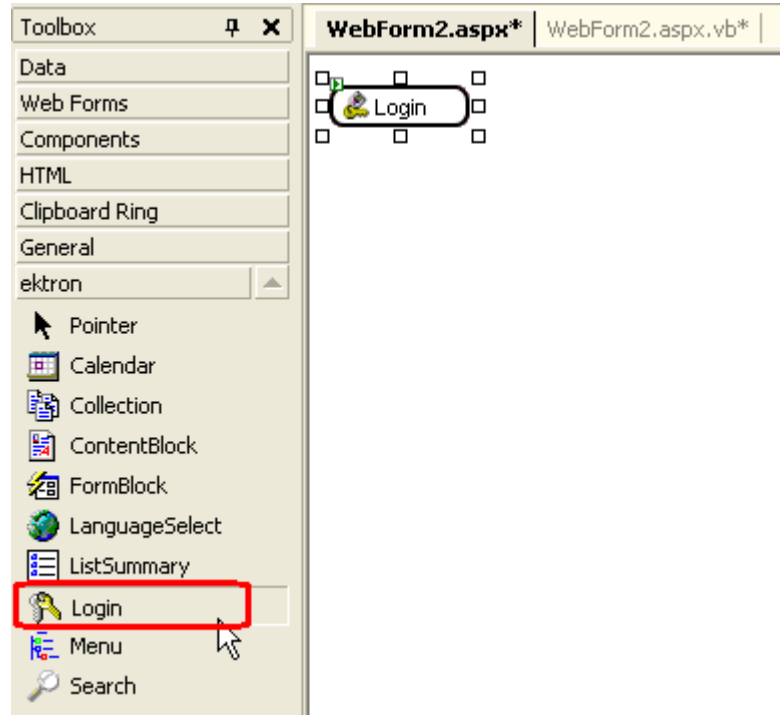
Use these methods to customize **Ektron CMS400.NET**. To use them, insert them onto your template pages to invoke content blocks, list summaries, and other objects stored in your database. They are the primary building blocks of your **Ektron CMS400.NET** Content Management System.

Replacement of Standard Methods

Beginning with Release 4.7, **Ektron CMS400.NET** utilizes ASP.NET server controls, which replace most Standard ASP.NET methods. With **Ektron CMS400.NET**'s new toolbox of pre-built, reusable, data-bindable ASP.NET controls, Ektron empowers developers working in Visual Studio .NET to rapidly integrate Ektron CMS components into their templates and sites. For more Information on Ektron CMS400.NET controls, see "[DhtmlMenu Server Control](#)" on [page 106](#).

The following examples shows the Login Server Control added by drag and drop vs. adding a Login method by the ASP.NET method.

Drag and Drop Login Server Control



Adding a Login function via an ASP.NET method.

```
<tr>
  <td>
    <%=AppUI.ecmLogin()%>
  </td>
</tr>
```

Best Practice

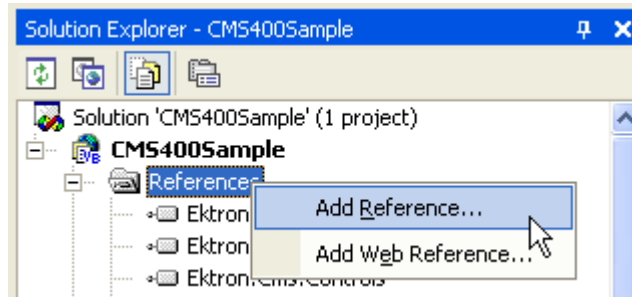
Ektron recommends using CMS400.NET server controls when available over Ektron standard ASP.NET methods. The methods are being replaced by the server controls.

You can still use the standard ASP.NET methods in **Ektron CMS400.NET**.

Referencing Methods in Visual Basic

Before you can use methods with Visual Basic, you must add a reference to them from Visual Basic. Follow these steps to do so.

1. Open Visual Basic.
2. Choose or add a project.
3. In Solution Explorer, right click **References** and choose **Add Reference**.

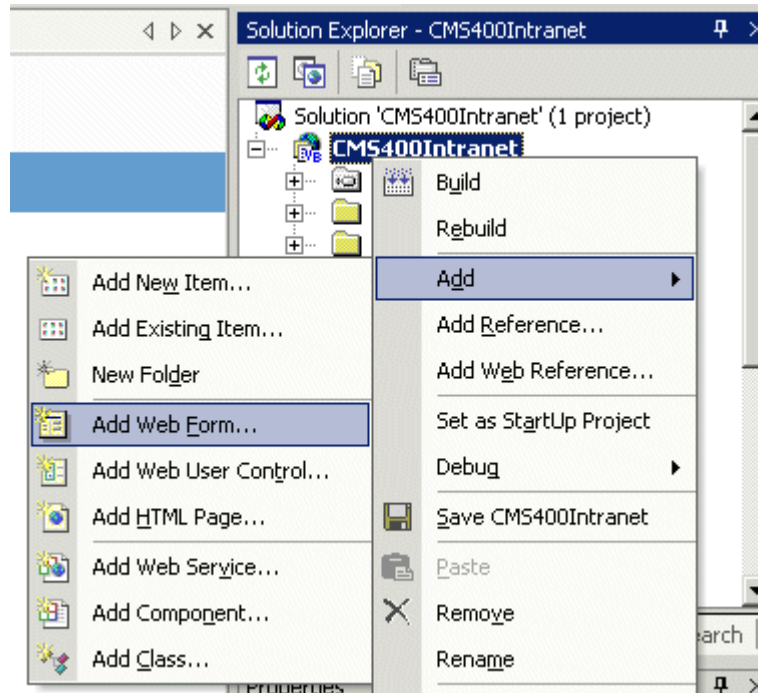


4. In the Add Reference dialog box, click **Browse**.
5. Go to `Inetpub/wwwroot/CMS400Example/Bin`.
6. To load the standard methods click `Ektron.Cms.UI.CommonUI.dll`.
To load the XML Indexing methods, click `Ektron.Cms.Common.dll`.
7. Click **Open**, then **OK**.

Adding a New Web Form

To add a new Web form, follow these steps.

1. Open your project in Visual Studio .NET.
2. Right click the project.
3. Click **Add > Add Web Form**.



4. Click **Web Form**.
5. In the code behind, add an `Imports Ektron.Cms.UI.CommonUI` statement.

Requirements for Standard Methods

Ektron CMS400.NET maintains and displays content through the use of Ektron ASPX methods and ASP.NET assembly objects. To use the methods, include the following code in your template:

```
<%@ Page Language="vb" AutoEventWireup="false" %>

<%@ Import Namespace="Ektron.Cms.UI.CommonUI" %>

<% Dim AppUI As New ApplicationAPI %>
```

Each line is explained below.

Line	Description
<code><%@ Page Language="vb" AutoEventWireup="false" %></code>	A generic .net line of code that should always be placed in the template
<code><%@ Import Namespace="Ektron.Cms.UI.CommonUI" %></code>	An import statement that calls the namespace <code>Ektron.CMS.UI.CommonUI</code> to display information from the .net assembly file
<code><% Dim AppUI As New ApplicationAPI %></code>	Creates an instance of the <code>ApplicationUI</code> class that can be used by each method

These lines call the Ektron .net assembly object to allow the page to function properly. They must appear at the top of each template file.

Standard Methods

To customize your Ektron CMS400.NET application, you need to be familiar with these methods.

Method	Description	More Information
Add Content	Programmatically adds a content block(s) to Ektron CMS400.NET	"Add Content" on page 135
Collection	Displays links generated by collections.	"Collection Server Control" on page 86
Content Block	Displays and manages a content block.	"ContentBlock Server Control" on page 39
Event Calendar	Displays an event calendar	"Calendar Server Control" on page 92
Form	Displays a content block that is associated with a form.	"FormBlock Server Control" on page 48

Method	Description	More Information
IndexSearch	Displays on a Web page a set of search fields. A site visitor uses the page to search for information on your Web site.	"IndexSearch Server Control" on page 64
List Summary	Displays a list of content block summaries.	"ListSummary Server Control" on page 80
Login	Displays the login or logout button, which lets the user log in and use Ektron CMS400.NET.	"Login Server Control" on page 33
Metadata	Displays the metadata in the source of the Web page displaying the content.	"MetaData Server Control" on page 114
Menu	Customizes the DHTML menus feature	"DhtmlMenu Server Control" on page 106
Search	Inserts a text box with a search button in the template. When the user completes the search screen, it looks through XHTML content on the Web site.	"Search Server Control" on page 51
ShowSiteLanguage	Lets a site visitor select a language in which to view site.	"LanguageSelect Server Control" on page 35

Add Content

The add content method lets you programmatically add a new content block to Ektron CMS400.NET without logging in. This method can be used for several applications, such as

- A Web form allowing you to collect data from users about the content block
- A loop function that moves information from a database to an Ektron CMS400.NET database

Here is the format for the add content method.

```

<% AddContent
(   UserName,
    Password,
    Domain,
    content_title,
    content_comment,
    ContentHtml,
    SummaryHTML,
    MetalInfo,
    ContentLanguage,
    FolderID,
    GoLive,
    EndDate,
    ErrString
)%>

```

The table below lists attributes of the add content block method.

Attribute	Description	Type
UserName	The user to be authenticated for permissions for eligibility to add a content block.	As String
Password	The password for the username being authenticated.	As String
Domain	If using Active Directory, specify the domain needed to authenticate the user.	As String
Content_title	Title of content block being added. <i>See Also:</i> "Managing Content" in the Ektron CMS400.NET Administrator Manual	As String
Content_comment	Historical comment for content being added.	As String
ContentHTML	Content for content block.	As String
SummaryHTML	Summary information about content block.	As String
MetalInfo	Information for the metadata that will be used as a reference point for finding other words or information.	As Object

Attribute	Description	Type
ContentLanguage	Language of the content. See <i>Also</i> : “Multi-Language Support” in the Ektron CMS400.NET Administrator Manual	As String
FolderID	ID number of content folder to which content block will be added. See <i>Also</i> : “Managing Content Folders” in the Ektron CMS400.NET Administrator Manual	As String
GoLive	Date and time content will go live on the Web site. <u>Must match date/time format set in web.config.</u>	As String
EndDate	Date and time content will no longer display on Web site. <u>Must match date/time format set in web.config.</u>	As String
ErrString	String returned if content fails to get added to Ektron CMS400.NET.	As Object

Add Content Example

The following is an example of the add content method hard coded in a template.

```
<tr>
  <td>
    <% AddContent ("admin", "admin", "", "Content Title", "Comment",
      "This is the <b>body</b> of the content.", "This is the
      <i>summary</i> for the content.", 1, 31, "06-14-2004 11:55 AM",
      "", "<metadata><meta id=""3"">ContentTitle</meta></metadata>",
      "") %>
  </td>
</tr>
```

The previous example has the following properties:

- The user admin is used for authentication
- The title of the new content block is “Content Title”

- The comment for the content is “Comment”
- The body of the content is “This is the **body** of the content”
- The summary of the content block is “This is the *summary* for the content
- The language ID = 1033, or English
- The new content block is added to content folder ID=31
- The start date is set at June 14, 2004 at 11:55 AM
- No end date is specified
- The metadata “Content title” is set for the metadata definition ID=3

When this method is executed, the content is added to Ektron CMS400.NET. Displayed here is the record of the content in the workarea.

The screenshot displays the Ektron CMS400 Workarea interface within a Microsoft Internet Explorer browser window. The title bar reads "CMS400 Workarea for Application Administrator - Microsoft Internet Explorer". The interface includes a navigation pane on the left with a tree view containing folders: Content, Human Resources, Marketing, Products, Support, and Syndication. Below the tree view are buttons for Content, Forms, Library, Modules, and Settings. The main content area is titled "View Content 'Content Title'" and shows the following details:

- Content Title:** Content Title
- Content ID:** 31
- Status:** Approved (Published)
- Last User To Edit:** admin
- Last Edit Date:** 14-Jun-2004 22:54:00 PM
- Start Date:** 14-Jun-2004 11:55:00 AM
- End Date:** [None Specified]
- Date Created:** 14-Jun-2004 22:49:00 PM
- Approvals:** [None Specified]
- XML Configuration:** [None Specified] HTML Content Assumed(*inherited*)
- Path:** \
- History Comment:** Comment

Metadata:

- Title:** ContentTitle
- Keywords:**

Summary:
This is the *summary* for the content.

Content:
This is the **body** of the content.

Web Services

Introduction to Web Services

Web Services (WS) are reusable software components that can be consumed by variety of clients, independent of operating system, programming language, or Web server, using standard Internet technologies. Because Web Services allow heterogeneous systems to communicate with each other in a standard way, they have received quick acceptance for data interchange over the Web.

It's easy for anyone to integrate your WS into their application. The client application can be a Web application, a Windows application, a Java application, a Visual Studio.net application, or a host of others -- it does not matter as long as it supports Web Services.

A Web Service consists of one or more methods that expose functionality. In this sense, a Web Service is an API that can be accessed using SOAP over HTTP. (SOAP is an XML-based, lightweight protocol that defines the message format of Web Services.)

For example, you might develop a purchasing application that automatically obtains price information from several vendors, lets the user select a vendor, submits the order and then tracks the shipment until it is received. The vendor application, in addition to exposing its services on the Web, might in turn use XML Web services to check a customer's credit, charge the customer's account and set up the shipment with a shipping company.

This manual does not explain how Web Services work. To learn more about them, try these sites.

- <http://msdn.microsoft.com/webservices/understanding/readme/default.aspx>

- <http://aspalliance.com/jnuckolls/articles/introws/default.aspx>
- <http://aspnet.4guysfromrolla.com/articles/062602-1.aspx>
- <http://www.15seconds.com/Issue/010430.htm>

CMS400.NET's Implementation of Web Services

CMS400.NET supports XML Web Services. This means that you can create pages that offer your content to any Web-enabled device, and you can create pages that consume existing Web Services. CMS400.NET's Web Services feature can return data separated from presentation. In combination with XML support, this feature lets consumers retrieve your XML data independent of presentation information and act on it however they see fit.

Benefits of Using Web Services

Web Services make it easy to connect with partners. They can also deliver more personal, integrated experiences to users via smart devices, including PCs. WSs can save time and money by reducing development time, and increase revenue by making your own XML Web services available to others.

WS Components and CMS400.NET Architecture

Web Services require the installation of the .NET Framework on the server.

Ektron's Web Service, ContentWS.asmx, runs within the .NET Framework. Through the Application API class in the Ektron.cms.commonUI.dll, .NET managed code can communicate with asp.net assembly code. ContentWS.asmx utilizes the ektron.com.commonui.dll file to provide access to the Content Business Objects (content blocks, menus, collections, list summaries, etc.).

Software and Hardware Requirements

- .NET framework, version 1.1 SP2, must be installed on the server (for more information, see <http://msdn.microsoft.com/netframework/>).

- Ektron CMS400.NET, version 4.5 or higher

Installation of Files

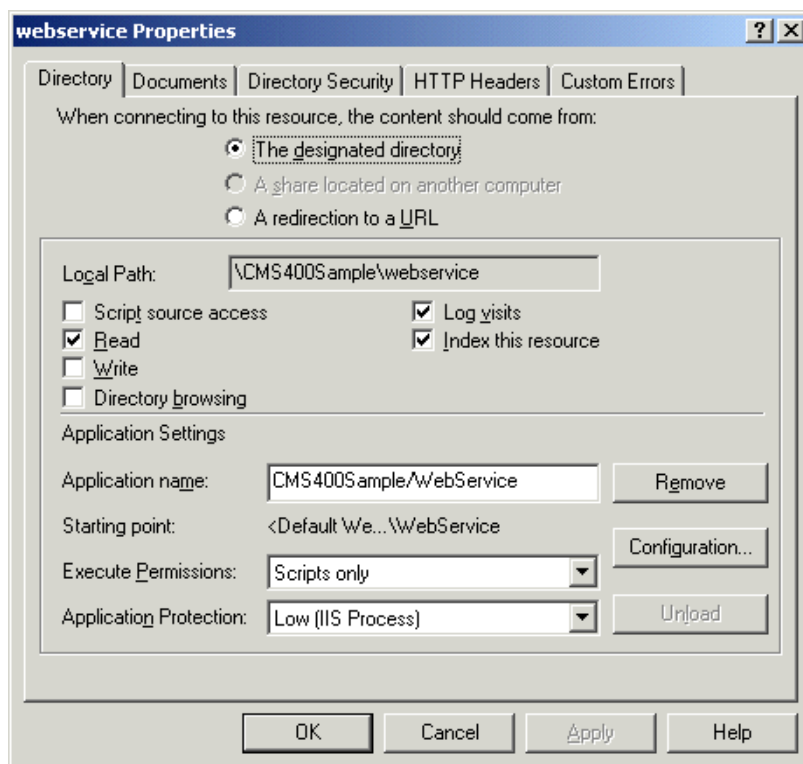
Below is a list of the files and their locations that make up the Web Services feature. They are installed when you install CMS400.NET.

File or folder location	Description
<code>webroot/CMS400Example/cms400scripts/ContentWS.aspx</code>	Entry point for Web Service. This Web page displays examples of all methods.
<code>webroot/CMS400Example/bin</code>	Directory for compiled Web Service code
<code>webroot/CMS400Example/bin/WebService.dll</code>	Web Service dll

Setup Instructions

After you install CMS400.NET, you need to open IIS, select `CMS400Example/cms400scripts`, and make it an application. To make a folder an application, follow these steps.

1. Open IIS.
2. Right click one of the folders listed above.
3. Select **Properties**. The Properties dialog appears.



4. Make sure you are on the **Directory** tab, which is selected by default.
5. Press the **Create** button next to the Application Name field.
6. Assign the folder's name in the **Application name** field.
7. Press **OK**.

If you are creating a new directory for your Web site, you must copy the following files into it from `webroot/CMS400Example/bin`.

- WebService.dll
- Ektron.Cms.BE.Content.dll
- Ektron.Cms.BE.Font.dll
- Ektron.Cms.BE.Library.dll
- Ektron.Cms.BE.License.dll
- Ektron.Cms.BE.Message.dll
- Ektron.Cms.BE.Module.dll

- Ektron.Cms.BE.Site.dll
- Ektron.Cms.BE.ToolBar.dll
- Ektron.Cms.BE.User.dll
- Ektron.Cms.BE.UserMgmt.dll
- Ektron.Cms.Common.dll
- Ektron.Cms.DataAccess.dll
- Ektron.Cms.DataRW.dll
- Ektron.Cms.Global.dll
- Ektron.Cms.UI.CommonUI.dll
- Ektron.Cms.UI.dll
- Ektron.Cms.WebserviceUI.dll

Providing Web Service Information in Your CMS

CMS400.NET Methods

CMS400.NET implements a single Web Service, ContentWS.asmx, that exposes several methods for retrieving CMS400.NET content. These methods mirror those in the WebService.dll file.

The methods determine the *kind* of content that is made available (for example, a content block, a collection of content blocks, a summary list) along with the *presentation* of that information (for example, an HTML string or an XML data structure representing content block information).

Below is a list of methods that ContentWS exposes with a brief description of functionality and differences from the applicationAPI implementation.

All methods are demonstrated on the sample page `webroot/CMS400Example\cms400scripts\ContentWS.aspx`. For more information, see ["The CMS400.NET Content Sample Page"](#) on page 148.

Method	Exposes this applicationApi method as a web service method	Returns	Changes to standard parameter values
ContentBlock	ecmContentBlock	The content block as an HTML string.	
ContentBlockEx	ecmContentBlockEx	The content block as an XML string. If necessary, performs XSLT transformation on content prior to its return.	
GetContentBlock	ecmContentBlock	An XML data structure representing information in the ContentBlock.	
Collection	ecmCollection	A Collection presented as an HTML string whose format is determined by the display function that you specify. You can create a custom display function in a separate .NET dll.	
GetCollection	ecmCollection	An XML data structure representing information about a Collection.	additional parameter: GetHTML. Possible values: 1 -returns html content for each collection item 0 - does not return it
ListSummary	ecmListSummary	List Summary information as an HTML string.	
GetListSummary	ecmListSummary	An XML data structure representing information about the List Summary.	parameters ShowSummary, StyleInfo, ShowInfo not available.

SingleSummary	ecmSingleSummary	Single Summary information as an HTML string.	
GetSingleSummary	ecmSingleSummary	An XML data structure representing the Single Summary information.	parameters StyleInfo, ShowSummary, ShowInfo not available.
MetaData	ecmMetaData	Meta Data information as an HTML string	does not have the unused spare parameter.
GetMetaData	ecmMetaData	An XML data structure representing Meta Data information	does not have the unused spare parameter.
SearchDisplay	ecmSearchDisplay See Also: " The Search Display Method Parameters " on page 150	Search Display information as an HTML string	
GetSearchDisplay	ecmSearchDisplay See Also: " The Search Display Method Parameters " on page 150	An XML data structure representing the Search Display information	StyleInfo and ShowDate parameters not available
GenerateHTMLMenu	ecmGenerateMenu	Menu details as an html string	Parameters as menu ids (comma separated value)
DropHTMLMenu	ecmDropMenu	Required menu as an html string	Parameters: <ul style="list-style-type: none"> • Root menu id (that is, the generateHTMLMenu value) • MenuId - required menuid • title - required title

GetFormBlock	ecmFormBlock	Form information as an HTML string	Parameters: <ul style="list-style-type: none"> • formId • formdata generated through webserviceUI • formtag required flag (set to true to embed <code><form></code> tag; otherwise, set to false)
GetEventsByCalendar	ecmEvtCalendar	Calendar information as an HTML string	<ul style="list-style-type: none"> • Calendar id • event id • month • year • url • querystring
Show RandomSummary	ecmShowRandomSummary	Randomly displays the summary content of a collection item.	<ul style="list-style-type: none"> • collection ID
ShowRandomContent	ecmShowRandomContent	Randomly displays the content of a content block in a specified collection	<ul style="list-style-type: none"> • collection ID • xsltID

AddContent	ecmAddContent	Lets you programmatically add a new content block to Ektron CMS400 without logging in.	<ul style="list-style-type: none"> ● Username ● password ● Domain ● content_title ● content_comment ● ContentHTML ● SummaryHTML ● ContentLanguage ● FolderID ● GoLive ● EndDate ● MetaInfo ● ErrString
------------	---------------	--	---

The CMS400.NET Content Sample Page

The CMS400.NET content sample page ([webroot\cms400scripts\ContentWS.asmx](#)) lists the methods described in the table above. Click each method on the page to see sample calls and output using various protocols.

When you open the detail page, you see **Test** followed by parameters.

Click [here](#) for a complete list of operations.

GetContentBlock

Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Value
ContentId:	<input type="text"/>

SOAP

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /cms300sample/cms300ws/CMS300Content.asmx HTTP/1.1
Host: 192.168.0.219
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "urn:ektron:webservices:content:cms300:v21/GetContentBlock"
```

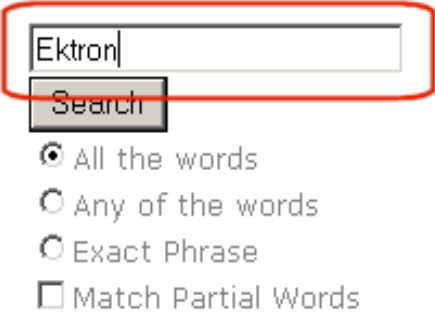
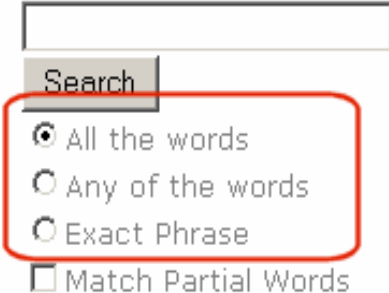
The top section of the page lets you test the method by using the HTTP Get protocol to retrieve data from the sample database. To do this, you must insert valid parameter values.

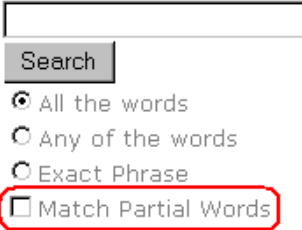
The next section of the CMS400.NET content sample page displays the following kinds of sample output.

- A sample SOAP request and response.
- A sample HTTP GET request and response.
- A sample HTTP POST request and response.

Review the test pages to learn the kind of content you can retrieve from this method.

The Search Display Method Parameters

Parameter	Where Defined in Application API.xxx	How Implemented by CMS
Search Text	n/a	<p>Search text used in search method.</p> 
Search Type	n/a	<p>Type of search used in search method.</p>  <p>Options and - all the words or - any of the words exact phrase - exact phrase</p> <hr/> <p>Note: This parameter is handled automatically by CMS when displayed on Web page. For Web Services, however, this needs to be defined.</p>
Recursive	Search	<p>Defines whether search is recursive .-</p> <p>1 - recursive search 0 - non-recursive search</p>

Parameter	Where Defined in Application API.xxx	How Implemented by CMS
Starting Folder	Search	<p>Describes path to folder in which search begins. - Here are some examples.</p> <ul style="list-style-type: none"> • \ - root content folder • \Products\RC Cars - begins search in the RC Cars folder.
Allow Fragments	n/a	<p>Selected by user on page that performs search.</p>  <p>Options</p> <p>0 (false) - do not allow fragments. 1 (true) - allow fragments.</p> <p>If fragments are allowed, and the user enters "AT" as the search string, the search returns topics with these words: pat, ate, hatter.</p> <p>If fragments are <i>not</i> allowed, and the user enters "AT" as the search string, the search only returns topics with the word: at.</p> <hr/> <p>Note: This parameter is handled automatically by CMS when displayed on Web page. For Web Services, this needs to be defined.</p> <hr/>
Max Number	Search Display	<p>Specify the maximum number of topics to display when a search is performed. For example:</p> <p>0 - unlimited 1 - one result 7 - seven results</p>

Parameter	Where Defined in Application API.xxx	How Implemented by CMS
Style Info	Search Display	Specify an inline style that is applied to the hyperlinked portion of topics that result from a search. For example: "font-family:arial; font-weight:bold; backgroundcolor:#cccccc; border:solid blue 1pt; margin:2px; width:50%;"
Show Date	Search Display	Specify whether the last modified date is displayed with search results. 0 - date is not displayed. 1 - date is displayed.

Sample Output from One Method

The following is an example of the content that can be retrieved from one of the methods, `GetContentBlock`. As you can see, the HTML content is provided within the `<ContentHtml>` XML element.

GetContentBlock

```
<?xml version="1.0" encoding="utf-8" ?>

<ContentBlockResult xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:ektron:webservices:content:cms400:v45">

  <ErrorString />

  <Item>

    <ContentTitle>Plastic Molder #123</ContentTitle>

    <ContentHtml>

<?xml version="1.0"?> <jobposting> <job-title>Plastics Molder</job-title> <job-number>123</job-number> <description><p>RC International, a leading supplier of consumer level radio controlled vehicles, is looking for an experienced plastics molder to join our team. The candidate will join our highly skilled team of enthusiastic molders to bring to life our next generation products. RC International is experiencing extremely rapid growth and offers many opportunities for advancement.</p></description> <qualifications> <ul> <li>3 years experience in commercial grade plastic molding technologies.</li> <li>Excellent written and oral communication skills.</li> <li>Ability to work with very little supervision.</li> <li>An AS degree in chemistry.</li> </ul> </qualifications> <offering> <p>First and second shift opportunities exist. RC International offers competitive pay, an extensive benefits package including a 401k plan, profit sharing, and health coverage.</p> </offering> <contact>
```

```

<contact-name>Human Resources Manager</contact-name> <contact-email>jobs@CMS400site.com</
contact-email> <contact-fax>1-555-555-5555</contact-fax> </contact> </jobposting>

</ContentHtml>

<Xslt1>C:\Inetpub\wwwroot\CMS400Example\XMLFiles\xslt\rc_jobs_display.xsl</Xslt1>

<Xslt2 />

<Xslt3 />

<Xslt4 />

<Xslt5 />

<DefaultXslt>1</DefaultXslt>

</Item>

</ContentBlockResult>

```

The News.aspx Sample (non-XML)

To help you understand how the samples work, this section explains the news sample .NET page, `news.aspx`, which is installed to `webroot\CMS400Example\WebServiceDemo/`. Specifically, four key sections of code are explained:

- [Identifying the Code Behind File](#)
- [Setting Up an Instance of Web Services](#)
- [Retrieving the Web Service's URL](#)
- [Getting Content Blocks](#)

Identifying the Code Behind File

Here is the first line in the `news.aspx` file:

```

<%@ Page CodeBehind="news.aspx.vb" Language="vb" AutoEventWireup="false"
Inherits="WebServicesDemo.news" %>

```

This line indicates that the page's functionality is set up in the Code Behind Visual Basic file, `news.aspx.vb`. The rest of the `news.aspx` file contains only page display information.

Setting Up an Instance of Web Services

The following code in news.aspx.vb sets up an instance of the Web Services. Every page that uses Web Services must contain such a declaration.

```
Dim wsCms400WebService As New ContentWS.ContentWS
```

Getting Content Blocks

The following lines of news.aspx.vb code retrieve all content blocks in the sample database's news folder formatted as an HTML string.

```
If (iId > 0) Then

    strHTML = wsCms400WebService.ContentBlock(iId)

    strHTML = Replace(strHTML, ".asp", ".aspx")

    NewsInfo.InnerHtml = strHTML

Else

    strHTML = wsCms400WebService.ListSummary("\Marketing\News", 0, 1, "", "Title,asc",
    "DateModified", 0, "", "")

    strHTML = Replace(strHTML, "news.asp", "WebServiceDemo/news.aspx")

    NewsInfo.InnerHtml = strHTML

End If
```

NOTE Since the database was created with .NET, you could use this code to get the HTML for the requested ID:

```
NewsInfo.InnerHtml = wsCms400WebService.ContentBlock(iId)
```

As you can see, we are assigning to the variable NewsInfo.InnerHtml the Web service (wsCms400WebService) followed by the ContentBlock method. (For a list of methods, see ["CMS400.NET Methods" on page 144.](#))

If the user selects a content block on the List Summary page (that is, clicks the blue text of the content title), that action passes an id value. The blue text is circled in the illustration below.

[RC International Ships RC Sportster](#) (8/27/2002 2:46:12 PM)

RC International, an emerging leader in the RC racing industry, has begun shipments of its Sportster. The RC Sportster is designed to compete with the entry and mid level RC racer

A new page appears, using the ContentBlock method to display the selected block.

```
If (iId > 0) Then
    strHTML = wsCms400WebService.ContentBlock(iId)
```

The ContentBlock method retrieves the content block as an HTML string, so the result looks like this.

Click [here](#) for a complete list of operations.

GetContentBlock

Test

To test the operation using the HTTP GET protocol, click the 'Invoke' button.

Parameter	Value
ContentId:	<input type="text"/>
<input type="button" value="Invoke"/>	

SOAP

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /cms300sample/cms300ws/CMS300Content.asmx HTTP/1.1
Host: 192.168.0.219
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "urn:ektron:webservices:content:cms300:v21/GetContentBlock"
```

If the user does not select a content block (which is true when the page first appears), the page uses the ListSummary method to display all blocks in the `marketing/news` folder using the List Summary format.

```
Else
    strHTML = wsCms400WebService.ListSummary("\Marketing\News", 0, 1, "",
    "Title,asc", "DateModified", 0, "", "")
```

The ListSummary method retrieves the content blocks as an HTML string, so the page looks like this.

Latest News

RC International Announces the RC Cheetah (8/27/2002 2:43:01 PM)

RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.

RC International Hosts RC Conference (8/27/2002 2:44:44 PM)

RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.

RC International Ships RC Sportster (8/27/2002 2:46:12 PM)

RC International, an emerging leader in the RC racing industry, has begun shipments of its first product, RC Sportster. The RC Sportster is designed to compete with the entry and mid level RC racer currently on the market.

The News.aspx Sample (XML)

The news.aspx sample file in the /cmd400Example/WebServiceDemo folder includes all elements described in ["The News.aspx Sample \(non-XML\)" on page 153](#) plus demonstrations of how to retrieve XML content.

Setting Up Instances of Web Services

In addition to creating an instance of the Web Service, the XML version of news.aspx.vb creates instances of the ContentBlock and ListSummary methods, which are used on the page.

```
Dim wsCms400WebService As New localhost.CMS400Content()

Dim gcbCms400ContentXml As localhost.ContentBlockResult

Dim glsCms400ListSummaryResult As localhost.ListSummaryResult
```

Getting Content Blocks

Look at this code from news.aspx.vb.

```
glsCms400ListSummaryResult = wsCms400WebService.GetListSummary("\Marketing\News",
0, 1, "", "Title,asc", "DateModified", 0, "", "")

If (Len(glsCms400ListSummaryResult.ErrorString) = 0) Then

strHTML = "<table><tr><td><b>Content Block Information Return from the      GetListSummary Web
service call</b></td></tr>"
```



```
strHTML = strHTML & "<tr><td>&nbsp;</td></tr>"

For iLoop = 0 To (glsCms400ListSummaryResult.Item.Length - 1)

strHTML = strHTML & "<tr><td><b>Content ID: </b>" &
    glsCms400ListSummaryResult.Item(iLoop).ContentID & "</td></tr>"

strHTML = strHTML & "<tr><td><b>Content Title: </b>" &
    glsCms300ListSummaryResult.Item(iLoop).ContentTitle & "</td></tr>"

strHTML = strHTML & "<tr><td><b>Date Created: </b>" &
    glsCms400ListSummaryResult.Item(iLoop).DisplayDateCreated & "</td></tr>"

strHTML = strHTML & "<tr><td><b>Date Modified: </b>" &
    glsCms400ListSummaryResult.Item(iLoop).DisplayDateModified & "</td></tr>"

strHTML = strHTML & "<tr><td><b>Modified By: </b>" &
    glsCms400ListSummaryResult.Item(iLoop).EditorFname & " " &
    glsCms400ListSummaryResult.Item(iLoop).EditorLname & "</td></tr>"

strHTML = strHTML & "<tr><td><b>Content Teaser: </b></td></tr>"

    strHTML = strHTML & "<tr><td>" & glsCms400ListSummaryResult.Item(iLoop).ContentTeaser &
"</td></tr>"

strHTML = strHTML & "<tr><td>&nbsp;</td></tr>"

Next
```

This code loops through all content blocks in the `Marketing\News` folder. For each block, the page uses the Web Service method `glsCms400ListSummaryResult.Item(iLoop)` to display the following XML data.

- Content ID
- Content Title
- Date Created
- DateModified
- Modified By
- Content Teaser

The result looks like this.

Latest News

Content Block Information Return from the GetListSummary Web service call

Content ID: 11

Content Title: RC International Announces the RC Cheetah

Date Created: 25-Apr-2002 04:43:42 PM

Date Modified: 27-Aug-2002 02:43:01 PM

Modified By: Application Administrator

Content Teaser:

RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.

Content ID: 10

Content Title: RC International Hosts RC Conference

Date Created: 25-Apr-2002 04:42:55 PM

Date Modified: 27-Aug-2002 02:44:44 PM

Modified By: Application Administrator

Content Teaser:

RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.

Content ID: 9

Content Title: RC International Ships RC Sportster

Date Created: 25-Apr-2002 04:42:10 PM

Date Modified: 27-Aug-2002 02:46:12 PM

The lower portion of the page also displays the content block in a ListSummary format, as described in ["Getting Content Blocks" on page 154](#).

Latest News

RC International Announces the RC Cheetah (8/27/2002 2:43:01 PM)

RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.

RC International Hosts RC Conference (8/27/2002 2:44:44 PM)

RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.

RC International Ships RC Sportster (8/27/2002 2:46:12 PM)

RC International, an emerging leader in the RC racing industry, has begun shipments of its first product, RC Sportster. The RC Sportster is designed to compete with the entry and mid level RC racer currently on the market.

If the user selects a content block and clicks the blue text of any title, a new page appears with

- information about the block (title and any XSLTs used for rendering).
- a demonstration of the GetContentBlock method to display the content block in “raw” HTML or XML format.
- a demonstration of the ContentBlock method to display the content block in processed HTML or XML format.

Here is the code that displays the top section of the page.

```
strHTML = "<table><tr><td><b>Content Block Information from GetContentBlock Web Service: </b></td></tr>"

strHTML = strHTML & "<tr><td><b>Content Block Title: </b>" &
gcbCms400ContentXml.Item.ContentTitle & "</td></tr>"

strHTML = strHTML & "<tr><td><b>Xslt Used for Rendering: </b>"

Select Case (gcbCms400ContentXml.Item.DefaultXslt)

    Case 1

        strTmp = gcbCms400ContentXml.Item.Xslt1

    Case 2

        strTmp = gcbCms400ContentXml.Item.Xslt2

    Case 3

        strTmp = gcbCms400ContentXml.Item.Xslt3

    Case 4

        strTmp = gcbCms400ContentXml.Item.Xslt4

    Case 5

        strTmp = gcbCms400ContentXml.Item.Xslt5

End Select

If (Len(strTmp) = 0) Then

    strTmp = "None Used (HTML Content assumed)"
```

This code uses the `gcbCms400ContentXml` Web Service to get the content title: `gcbCms400ContentXml.Item.ContentTitle`.

The code also uses the `gcbCms3400ContentXml` Web Service to retrieve any XSLTs assigned to the content. If no XSLTs are

assigned (as is the case with this sample page), the message "None Used (HTML Content assumed)" appears.

This code results in the following section of the page.

Latest News

Content Block Information from GetContentBlock Web Service:

Content Block Title: RC International Announces the RC Cheetah

Xslt Used for Rendering: None Used (HTML Content assumed)

Next, the page uses the GetContentBlock method to display the content block in "raw" HTML or XML format. Here is the code that displays that section of the page.

```
strHTML = strHTML & "<tr><td><b>Raw HTML (or XML) from GetContentBlock Web Service: </b></td></tr>"
strHTML = strHTML & "<tr><td>&nbsp;</td></tr>"
strHTML = strHTML & "<tr><td>" & strRawHtml & "</td></tr>"
strHTML = strHTML & "<tr><td>&nbsp;</td></tr>"
```

Finally, the page uses the ContentBlock Web method to display the content block in "processed" format. Here is the code that displays that section of the page.

```
strHTML = strHTML & "<tr><td><b>Processed HTML (or XML) from ContentBlock Web Service: </b></td></tr>"
strHTML = strHTML & "<tr><td>" & strProcessedHtml & "</td></tr>"
strHTML = strHTML & "</table>"
```

Earlier in the news.aspx.vb file, the `strProcessedHtml` variable was defined as `strProcessedHtml = wsCms400WebService.ContentBlock(iId)`. So, the `strProcessedHtml` variable uses the `wsCms400WebService.ContentBlock` method to retrieve content in processed format.

Here is the resulting section of that page. As you can see, the content appears as within a browser.

RC International Announces the RC Cheetah

January 14, 2001 - RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah. After nearly two years of development the RC Cheetah is designed to compete at the highest level in the RC racing arena. The excitement level of the hard core enthusiasts can be felt at all the demonstrations of the RC Cheetah.

"The RC Cheetah price/performance ration is simply incredible", stated Bob Dickerson, RC International VP of engineering. "The RC Cheetah has been devastating its competition in all the pre-release races and we expect that to continue..."

The major industry magazines are agreeing with Mr. Dickerson's assessment. RC Magazine has tested the RC Cheetah head-to-head with the competition's best and they concluded:

"There is no comparison between the RC Cheetah and its nearest competitor. In speed, handling, and price the RC Cheetah out shines them all".

The RC Cheetah is now available at all RC International distributors and major Hobby stores. Please contact RC International to locate a local dealer.

Creating a Page that Provides a Web Service

Below are the typical steps you would follow to create your own page that provides a Web Service.

1. Decide which content blocks you want to provide. Here are some examples:
 - a specific content block (for example `wsCms400WebService.ContentBlock(2)` would display content block 2 regardless of the URL parameters)
 - all blocks in a folder (the `GetListSummary` method's first parameter is folder)
 - all blocks in a collection (the `Collection` method's first parameter is collection ID)
2. Decide what information you want to provide about those blocks, and how to present it. Review "[CMS400.NET Methods](#)" on page 144 to find the type of content and the kind of information that is available about it.

For example, if you want to display XML data about content (such as content title, date modified, date created), use a method that begins with "Get," such as `GetContentBlock`, `GetCollection`, and `GetListSummary`.

3. Create a new folder within the CMS400Example folder. This folder and some of its subfolders have files that are needed to run Web Services. Place all of your files within the new folder.
4. Using a development tool that supports Web Service integration (for example, Visual Studio .NET), create a .NET (.aspx) file that manages page layout. The file must refer to a Visual Basic or C# file (via a `CodeBehind` command), which manages the page's functionality.

Note: You can refer to the samples provided in `wwwroot/CMS400Example/WebServiceDemo` as examples. Rename and modify them as needed.

5. Add to each .NET project a Web Reference to the WSDL file. (by default, `webroot/CMS400Example/WebServiceDemo/WebReferences/ContentWS/ContentWS.wsdl`). The environment then builds the appropriate proxy class to access the Web Service's methods. You can use the proxy class within the application to invoke those methods. CMS300's sample .aspx pages demonstrate how to do this.
6. Provide to your consumers a URL to the .NET (.aspx) file.

Syndicating Web Content as RSS

This section explains how to add RSS syndication to Ektron CMS400.NET for ASP.NET. RSS syndication allows you to prepare Ektron CMS400.NET content for reading by RSS news feed aggregators.

To make this task easier, Ektron provides sample files to illustrate RSS syndication. You should customize the files to meet your needs.

What is RSS Syndication?

RSS stands for Really Simple Syndication, a common method of syndicating Web content as an XML document. There are several versions of RSS. Ektron has implemented version 2.0.

In Ektron CMS400.NET, you use RSS to prepare a collection list or a list summary for reading by an *aggregator*. An aggregator consumes an RSS feed and displays it to the user. (An aggregator is similar in function to a portal.) There are several popular aggregators, including:

- AmphetaDesk (<http://www.disobey.com/amphetadesk/>)
- NewzCrawler (<http://www.newzcrawler.com/>).

The RSS Specification

The RSS specification is a document: <http://backend.userland.com/rss>. Ektron has implemented the specification, but you can customize the RSS classes to support advanced features.

RSS Syndication in Ektron CMS400.NET

Ektron CMS400.NET allows you to syndicate content by using .NET. Syndicating via .NET requires that you have the Microsoft .NET framework installed on your computer.

With Ektron CMS400.NET, you can syndicate a collection and a list summary via RSS.

Syndicating via .NET

Architecture of RSS Syndication

aspx pages

You can use an aspx page as a URL to dynamically create and return RSS documents. To display an RSS feed on a Web site, the site administrator identifies the URL of the aspx page. The aggregator determines how to use the document's elements, such as <title>, <description> and <link>, to display the syndicated content.

Three sample aspx templates ([news.aspx](#), [hr.aspx](#), and [products.aspx](#)) are installed by default to `localhost/CMS400Example` folder. They use the `WebService.ContentWS` class to retrieve the collection or list summary data. Then, they create `CollectionToRSS20` or `ListSummaryToRSS20`. These classes fill the RSS 2.0 data classes and serialize them to an XML document, which can be returned as an HTTP response.

Since these pages are part of the CMS400.NET sample, you should modify them or create your own aspx pages for your Web site. Place your custom pages in the `yourwebsite/WebServices/` folder so that the `web.config` file is within the path.

Retrieving a Collection or List Summary as an RSS Feed

The left frame of the `corporate.aspx` page is hardcoded to retrieve the News Collection. The right frame retrieves a ListSummary.

Creating Your Own aspx Collection Page

To create your own collection page, copy the collection object to an RSS 2.0 object and serialize it. To accomplish this, use these methods.

```
ConvertToRSS(ByVal result CMS400WS.CollectionResult, ByVal
```

```
title As String, ByVal description As String) As RSS20.rss

Serialize(Optional ByVal stream As System.IO.Stream = Nothing,
Optional ByVal o As Object = Nothing)
```

Example

```
Dim objRSSConverter As New CollectionToRSS20(Request, Response)

objRSSConverter.ConvertToRSS(objResult,
objResult.CollectionTitle, objResult.CollectionDescription)

objRSSConverter.Serialize()
```

Remarks

The `ConvertToRSS` method returns the RSS 2.0 object so that you can fill in or modify additional fields as desired before calling `Serialize`. The `Serialize` method has optional parameters that let you send the RSS to a stream other than `HttpResponse`.

You can add code to serialize a different object (such as an RSS 1.0 data class) or to run an XSLT on the serialized XML document before returning it as the HTTP response.

Example

```
Dim objRSSConverter As New CollectionToRSS20(Request, Response)

Dim objRSS20 As rss.RSS20.rss =

objRSSConverter.ConvertToRSS(objResult,
objResult.CollectionTitle, objResult.CollectionDescription)

objRSS20.channel.language = "en-us"

objRSSConverter.Serialize()
```

Creating Your Own aspx List Summary Page

To create your own list summary page, copy a list summary object to an RSS 2.0 object and serialize it. To accomplish this, use these methods.

```
ConvertToRSS(ByVal result As CMS400WS.ListSummaryResult, ByVal  
title As String, ByVal description As String) As RSS20.rss  
  
Serialize(Optional ByVal stream As System.IO.Stream = Nothing,  
Optional ByVal o As Object = Nothing)
```

Example

```
Dim objRSSConverter As New ListSummaryToRSS20(Request, Response)  
  
objRSSConverter.ConvertToRSS(objResult, "News", "The latest news")  
  
objRSSConverter.Serialize()
```

Remarks

The `ConvertToRSS` method returns the RSS 2.0 object so that you can fill in or modify additional fields as desired before calling `Serialize`. The `Serialize` method has optional parameters that let you send the RSS to a stream other than `HttpResponse`.

You can add code to serialize a different object (such as an RSS 1.0 data class) or to run an XSLT on the serialized XML document prior to returning it as the HTTP response.

Example

```
Dim objRSSConverter As New ListSummaryToRSS20(Request, Response)  
  
Dim objRSS20 As rss.RSS20.rss =  
  
objRSSConverter.ConvertToRSS(objResult, "Title", "Description")  
  
objRSS20.channel.language = "en-us"  
  
objRSSConverter.Serialize()
```

Learning About Visual Studio .NET

This section introduces some background information about using Microsoft's Visual Studio .NET. For more information, use the help feature installed with Visual Studio.NET and Microsoft's developer Center (<http://msdn.microsoft.com/vstudio/>).

Grid Layout vs. Flow Layout

NOTE The following definitions are from Visual Studio Help:

Grid Layout - Absolute positioning attributes are inserted into elements that are added, and updated in elements that are moved. Elements can be dragged across the Design view surface. The positioning grid and Snap to Grid are available.

Flow Layout - Elements are added without absolute positioning attributes. Web browsers arrange elements in the order that they occur on the page, from top to bottom. You cannot drag elements across the Design view surface or use the positioning grid.

Grid layout is the default, which means that all controls drawn to the Web form in the designer window will have absolute positioning. Here is an example.

```
<body MS_POSITIONING="GridLayout">

  <form id="Form1" method="post" runat="server">

    <asp:Button id="Button1" style="Z-INDEX: 101; LEFT: 160px; POSITION: absolute; TOP: 80px"
runat="server" Text="Button"></asp:Button>

    <asp:Button id="Button2" style="Z-INDEX: 102; LEFT: 480px; POSITION: absolute; TOP: 88px"
runat="server" Text="Button"></asp:Button>

    <asp:DataGrid id="DataGrid1" style="Z-INDEX: 103; LEFT: 208px; POSITION: absolute; TOP:
152px" runat="server"></asp:DataGrid>
```

</form>

</body>

In Grid layout, you can position your controls like a WYSIWYG editor with no knowledge of HTML. However, because absolute positioning is not rendered consistently by all browsers, the page layout can be flexible based on the size of other controls on the page, and the Web browser window.

When other controls are dynamically populated, such as a DataGrid, controls that appear beneath it in the Web form would be obscured if they were positioned absolutely at design time. In addition, when utilizing globalization of pages with different languages, the size of text areas can vary and cause obstructions.

Additional Information

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

Support

If you are experiencing trouble with any aspect of Ektron CMS400.NET, please contact the Ektron Support Team via our Web Site at

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

Sales

For questions and answers about purchasing Ektron CMS400.NET from Ektron, contact the sales department by sending an e-mail to:

sales@ektron.com

Information

For general information about Ektron and its products, you may send an e-mail to:

info@ektron.com

Ektron, Inc.
5 Northern Blvd., Suite 6
Amherst, NH 03031
Email: sales@ektron.com
Tel: +1 603-594-0249
Fax: +1 603-594-0258