



Ektron CMS200/300 Developer Manual

Version 4.7, 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 2004 Ektron®, Inc. All rights reserved.

EKTRON is a registered trademark of Ektron, Inc.

Version 4.7, Revision 1, September 28, 2004

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 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

Summary Table of Contents

Author's Note	1
About This Manual	2
Web Services	3
Syndicating Web Content as RSS	25
Custom ASP Functions	43
Custom ASP.Net Functions	154
Custom ColdFusion Functions	236
Custom PHP Functions	302
Dreamweaver Extension Support	372
Additional Information	436

Detailed Table of Contents

Author's Note	1
About This Manual	2
What Else Is Included in this Manual?	2
Web Services	3
Introduction to Web Services	3
Ektron CMS200/300's Implementation of Web Services	4
Benefits of Using Web Services	4
WS Components and Ektron CMS200/300 Architecture	4
Software and Hardware Requirements	5
Installation of Files	6
Setup Instructions	7
Providing Web Service Information in Your CMS	8
Ektron CMS200/300 Methods	8
The Ektron CMS200/300 Content Sample Page	10
Sample Output from One Method.....	14
The News.aspx Sample (non-XML).....	15
The News.aspx Sample (XML).....	18
Creating a Page that Provides a Web Service.....	23
Syndicating Web Content as RSS	25
What is RSS Syndication?	25
The RSS Specification.....	25
RSS Syndication in Ektron CMS200/300.....	26
Syndicating via ASP	27
RSS Functions	27
ecmRssCollection.....	27
ecmRssSummary	28
Syndicating Collections	29
Example.....	29
Creating Your Own Collection News Feed Files	29
Configuring News Feed Application	31
Alternate Dynamic Method	32
Syndicating Summary Lists.....	32
Example.....	33
Creating Your Own Summary News Feed Files	33
Configuring News Feed Application	35
Alternate Dynamic Method	36

Syndicating via .Net	39
Architecture of RSS Syndication	39
aspx pages	39
Creating Your Own aspx Collection Page	40
Example.....	40
Example.....	41
Creating Your Own aspx List Summary Page	41
Example.....	41
Example.....	42
 Custom ASP Functions	 43
Login Function	44
Login Tag Example	45
Placing Your Login Button.....	46
Partner Login	46
Content Block Function.....	48
Content Block Examples.....	50
Static Content Block	51
Dynamic Content Block	52
XML Content Block.....	52
Dynamic and Static Combination.....	55
Add Content.....	56
Add Content Example	58
Add Content Web Form Example	60
Basic XML Search	61
Basic XML Search Tag Example	62
XML Search Tag Variables	63
Advanced XML Search	66
Advanced XML Search Tag Example	67
Search Tag Variables.....	68
XML Search Display.....	72
XML Search Display Example.....	73
Search Function	74
Search Tag Example	75
Search Tag Variables.....	76
Starting Folder	77
Recursive.....	77
Target Page	77
Text Box.....	78
Maximum Characters	78
Button Image Source.....	79
Button Text.....	79
Font Face	80
Font Size	81
Horizontal	82

Spare 1	82
Search Display Function	83
Search Display Example	83
Search Display with Default Content	84
Search Display Variables	85
Max Number of Content Blocks Returned	85
Style Info.....	86
Show Date	86
List Summary Function	88
List Summary Example	89
List Summary Variables	90
Starting Folder	90
Recursive.....	91
Show Summary	91
Style Info.....	92
Order By	92
Show Info.....	94
Max Number	94
Spare 1 & 2.....	95
Single Summary Function.....	96
Single Summary Example.....	97
Single Summary Variables.....	97
Content ID	98
Show Summary	98
Style Info.....	98
Show Info.....	99
Spare 1	99
Collection Function.....	100
Collection Examples.....	100
Example Using the ecmNavigation Display Function	101
Example Using the ecmTeaser Display Function	102
Collection Variables.....	104
ID	104
DisplayFunction	104
Creating Custom Display Functions	105
Function Structure	105
Building Blocks	108
Defining Display Functions	111
Using a Custom Display Function	111
Random Content	113
Random Summary	114
Event Calendar Function.....	115
Event Calendar Style Sheet	115
Event Calendar Sample.....	116
Abbreviating Day Names	118
Configuring ApplicationAPI.asp	119

Form Function	121
MultiPurpose Function	123
Metadata Function.....	124
Creating an Item List	124
ecmMetadata Example	125
More ecmMetadata Examples	127
ecmMetadata Variables	128
Item List.....	128
Spare 1	130
Show Site Language Function.....	131
Custom API.....	132
Multi-Level Collection	133
Customizing the Menu	133
ek_Menu.js	133
Version.....	135
Type.....	136
FontSize	136
FontWeight	136
FontFamily.....	137
FontColor.....	138
FontColorHilite.....	138
bgColor	139
menuBorder.....	140
menuItemBorder.....	140
menuItemBgColor.....	141
menuLiteBgColor.....	142
menuBorderBgColor.....	142
menuHiLiteBgColor	143
menuContainerBgColor	143
childMenuIcon	144
childMenuIconHilite	144
Application API	145
Customizing ApplicationAPI	145
menuItemBgColor.....	146
menuLiteBgColor.....	146
menuBorderBgColor.....	147
menuHiliteBgColor.....	148
menuContainerBgColor	148
menuItemBgColor.....	149
menuLiteBgColor.....	150
menuBorderBgColor.....	151
menuHiliteBgColor.....	151
menuContainerBgColor	152
Customizing ek_menu.js versus ApplicationAPI.....	153

Custom ASP.Net Functions154

Login Function	155
Login Tag Example	156
Placing Your Login Button	157
Partner Login	158
Content Block Function.....	159
Content Block Examples.....	161
Static Content Block	162
Dynamic Content Block	162
XML Content Block.....	163
Dynamic and Static Combination.....	164
Add Content.....	165
Add Content Example	167
Add Content Web Form Example	169
Search Function	170
Search Tag Example	171
Search Tag Variables	172
Starting Folder	173
Recursive.....	173
Target Page	173
Text Box.....	174
Maximum Characters	174
Button Image Source.....	175
Button Text.....	176
Font Face	177
Font Size	179
Horizontal	180
Spare 1	180
Search Display Function	181
Search Display Example	181
Search Display Variables	182
Max Number of Content Blocks Returned	182
Style Info.....	184
Show Date	184
List Summary Function	186
List Summary Example	187
List Summary Variables	188
Starting Folder	189
Recursive.....	189
Show Summary	190
Style Info.....	191
Order By	192
Show Info.....	193
Max Number	193
Spare 1 & 2.....	194
Single Summary Function.....	195

Single Summary Example	196
Single Summary Variables	197
Content ID	197
Show Summary	198
Style Info.....	198
Show Info.....	199
Spare 1	200
Collection Function	201
Collection Examples	201
Example Using the ecmNavigation Display Function	202
Example Using the ecmTeaser Display Function	203
Collection Variables	205
ID	205
DisplayFunction	206
Creating Custom Display Functions	207
Function Structure	207
Building Blocks	212
Defining Display Functions	215
Using a Custom Display Function	215
Random Content	217
Random Summary	218
Event Calendar Function	219
Event Calendar Style Sheet	219
Event Calendar Sample	220
Abbreviating Day Names	222
Configuring ApplicationAPI.aspx	223
Form Function	225
Form Sample	226
MultiPurpose Function	227
Show Site Language Function	228
Metadata Function	229
Creating an Item List	229
ecmMetadata Example	230
ecmMetadata Variables	232
Item List.....	232
Spare 1	234
Custom API	235
 Custom ColdFusion Functions	236
Login Function	238
Login Tag Example	238
Placing Your Login Button.....	239
Content Block Function.....	240
Content Block Examples.....	242

Static Content Block	243
Dynamic Content Block	244
XML Content Block.....	245
Random Content	248
Random Summary	249
Search Function	250
Search Tag Example	251
Search Tag Variables	252
Starting Folder	252
Recursive.....	253
Target Page	253
Text Box.....	253
Maximum Characters	254
Button Image Source	254
Button Text.....	256
Font Face	257
Font Size	259
Horizontal	260
Spare 1	260
Search Display Function	261
Search Display Example	261
Search Display Variables	262
Max Number of Content Blocks Returned	263
Style Info.....	263
Show Date	264
List Summary Function	266
List Summary Example	267
List Summary Variables	268
Starting Folder	268
Recursive.....	269
Show Summary	269
Style Info.....	270
Order By	271
Show Info.....	272
Max Number	273
Spare 1 & 2.....	274
Single Summary Function	275
Single Summary Example	275
Single Summary Variables	276
Content ID	277
Show Summary	277
Style Info.....	277
Show Info.....	278
Spare 1	279
Collection Function.....	280
Collection Examples.....	280

Example Using the ecmNavigation Display Function	280
Example Using the ecmTeaser Display Function	282
Collection Variables.....	284
ID	284
DisplayFunction	284
Event Calendar Function.....	288
Event Calendar Style Sheet	288
Event Calendar Sample	289
Abbreviating Day Names	290
Configuring ecmEvtCalendar.cfm	291
Form Function	292
MultiPurpose Function	294
Show Site Language Function.....	295
Metadata Function.....	296
Creating an Item List	296
ecmMetadata Example	297
ecmMetadata Variables	298
Item List.....	299
Custom ColdFusion Tags	301
 Custom PHP Functions	 302
Login Tag	304
Login Tag Example	304
Placing Your Login Button.....	305
Content Block Tag.....	306
Content Block Examples.....	308
Static Content Block	309
Dynamic Content Block	309
XML Content Block.....	310
Search Tag	312
Search Tag Example	313
Search Tag Variables	314
Starting Folder	315
Recursive.....	315
Target Page	315
Text Box.....	316
Maximum Characters	316
Button Image Source	317
Button Text.....	318
Font Face	319
Font Size	321
Horizontal	322
Spare 1	322
Search Display Tag	323

Search Display Example	323
Search Display Variables	324
Max Number of Content Blocks Returned	325
Style Info.....	326
Show Date	326
List Summary Tag	328
List Summary Example	329
List Summary Variables	330
Starting Folder	331
Recursive.....	331
Show Summary	331
Style Info.....	332
Order By	333
Show Info.....	334
Max Number	335
Spare 1 & 2.....	336
Single Summary Tag.....	337
Single Summary Example	337
Single Summary Variables	338
Content ID	339
Show Summary	339
Style Info.....	339
Show Info.....	340
Spare 1	341
Collection Function.....	342
Collection Examples.....	342
Example Using the ecmNavigation Display Function	342
Example Using the ecmTeaser Display Function	344
Collection Variables.....	346
ID	346
DisplayFunction	347
Random Content	351
Random Summary	353
Event Calendar Function.....	355
Event Calendar Style Sheet	355
Event Calendar Sample.....	356
Abbreviating Day Names	357
Configuring ApplicationAPI.php.....	358
Form Function	361
MultiPurpose Function	363
Metadata Tag.....	364
Creating an Item List	364
ecmMetadata Example	365
ecmMetadata Variables	366
Item List.....	367

Spare 1	369
Show Site Language Function.....	370
Custom API.....	371
 Dreamweaver Extension Support	 372
Requirements	373
Installing the Extension	374
Setup	379
Using the Ektron CMS200/300 Extension	383
Custom Function Symbols	384
Ektron CMS200/300 Commands Menu.....	386
Inserting Custom Functions	390
Manage Content.....	390
Accessing Manage Content Commands...	390
Edit Content.....	391
Save Content.....	393
Create Folder.....	399
CMS Content...	402
Accessing CMS Content Commands...	403
Insert a Login Tag...	403
Inserting a Partner Login Tag	405
Inserting a Static Content Block.....	406
Inserting a Dynamic Content Block	408
Inserting a Collection Tag	410
Inserting a List Summary Function	411
Inserting a Metadata Function	413
Inserting a Search Function.....	415
Inserting a Search Display Function.....	418
Inserting a Single Summary Function.....	420
Inserting a Random Content Function	422
Inserting a Random Summary Function	423
Inserting an Event Calendar Function	425
Inserting a Form Content Function	426
XML Functions...	428
Accessing XML Function Commands.....	428
Insert Content XSLT Tag...	429
Insert Meta XSLT Tag.....	431
Insert Collection XSLT Tag...	433
 Additional Information	 436
Supporting Documents	436
Support	436
Sales	436
Information	436

Author's Note

This manual documents both Ektron CMS200 and Ektron CMS300. With the exception of a few features that are labeled CMS300 only, Ektron CMS300 and Ektron CMS200 have the same functionality.

In this manual, some screens show CMS300 in the title bar, and some directory paths mention CMS300, such as `C:/Program Files/Ektron/CMS300/Utilities/`. If you are using CMS200, substitute 200 for 300 in these cases.

About This Manual

To utilize the full potential of an Ektron CMS200/300 driven Web site, learning and using the custom functions supplied by Ektron is essential. These custom functions will help make your site more robust, and easier to maintain.

This manual can be used as a reference for the developer, or the person is who is setting up your Ektron CMS200/300 Web site. This manual will explain all the custom tags for each of the platforms supported by Ektron CMS200/300 including:

- ASP
- ColdFusion
- ASPX (.NET)
- PHP

What Else Is Included in this Manual?

Along with explaining how to insert the functions to your Web templates manually, this manual also explains creating Ektron CMS200/300 templates in a WYSIWYG environment by using the Dreamweaver extension supplied with the installation. See ["Dreamweaver Extension Support" on page 372](#) for additional information.

Ektron CMS200/300 has also implemented support for Web Services as well as syndication using RSS. Information about configuration and use of these features is also found in this manual.

Web Services

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

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 Cold Fusion 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>

Ektron CMS200/300's Implementation of Web Services

Ektron CMS200/300 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. **Ektron CMS200/300'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 Ektron CMS200/300 Architecture

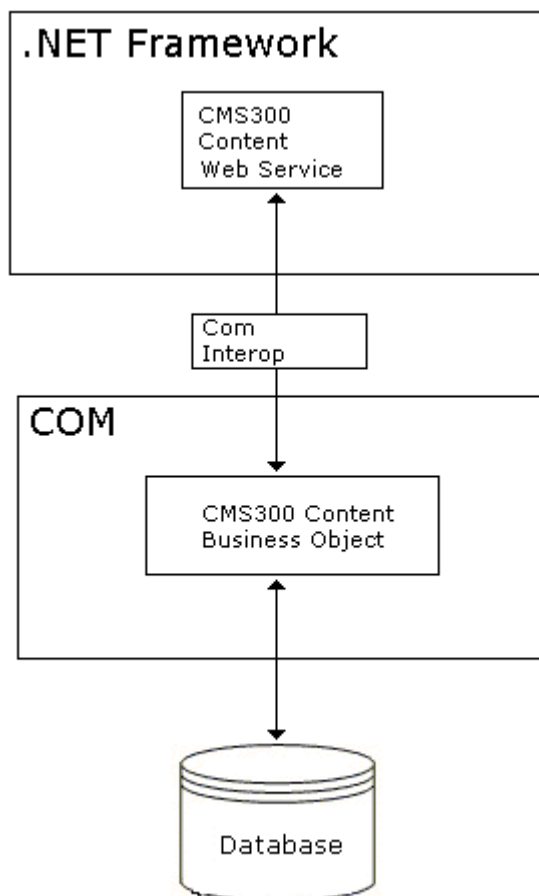
The following diagram shows how Ektron's Web Services interact with **Ektron CMS200/300**. Web Services require the installation of the .NET Framework on the server.

NOTE

For a detailed description of **Ektron CMS200/300's** architecture, see <http://www.ektron.com/whitepaper/Ektron-unique-component-based-approach-to-CM.pdf>.

Ektron's Web Service, cms300content.asmx, runs within the .NET Framework. Through COM Interop, a facility provided with the

.NET Framework, .NET managed code can communicate with COM code. cms300content.asmx utilizes the Content Business Object as applicationApi.asp to ensure that business logic functionality is consistent throughout **Ektron CMS200/300**.



Software and Hardware Requirements

- The .NET Framework, version 1 SP2, must be installed on the server (for more information, see <http://msdn.microsoft.com/netframework/>).
- **Ektron CMS200/300** version 2.5 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 **Ektron CMS200/300**. However, if you do not choose the .aspx (.NET) sample files during installation, the Web Service demos (cmswsdemo, cmsxmlwsdemo) are not installed.

File or folder location	Description
<code>webroot/cms300sample/cms300ws</code>	Root directory of the Web Services files
<code>webroot/cms300sample/cms300ws/web.config</code>	XML-based configuration file for web services; similar to setup.asp
<code>webroot/cmd300sample/cmsxmlwsdemo</code>	Samples that demonstrate the consumption of content block information as XML. For example, a page could process the WS page's date created, "go live" date, etc.
<code>webroot/cmd300sample/cmswsdemo</code>	Contains only the HTML version of the Web Service. That is, the samples only prepare a string of HTML data for reading by a browser.
<code>webroot/cms300sample/cms300ws/cms300content.asmx</code>	Entry point for Web Service. This page displays examples of all available methods.
<code>webroot/cms300sample/cms300ws/global.asax</code>	Application start file
<code>webroot/cms300sample/cms300ws/bin</code>	Directory for compiled Web Service code
<code>webroot/cms300sample/cms300ws/bin/CMS300WS.dll</code>	Web Service dll
<code>webroot/cms300sample/cms300ws/bin/WSContentDisplay.dll</code>	Collection display dll for four supplied display methods: <ul style="list-style-type: none"> • ecmNavigation • ecmTeaser • ecmNavigationPop • ecmTeaserPop
<code>webroot/cms300sample/cms300ws/plugins</code>	User-defined display dlls for collection method call. This is configurable within the web.config file.

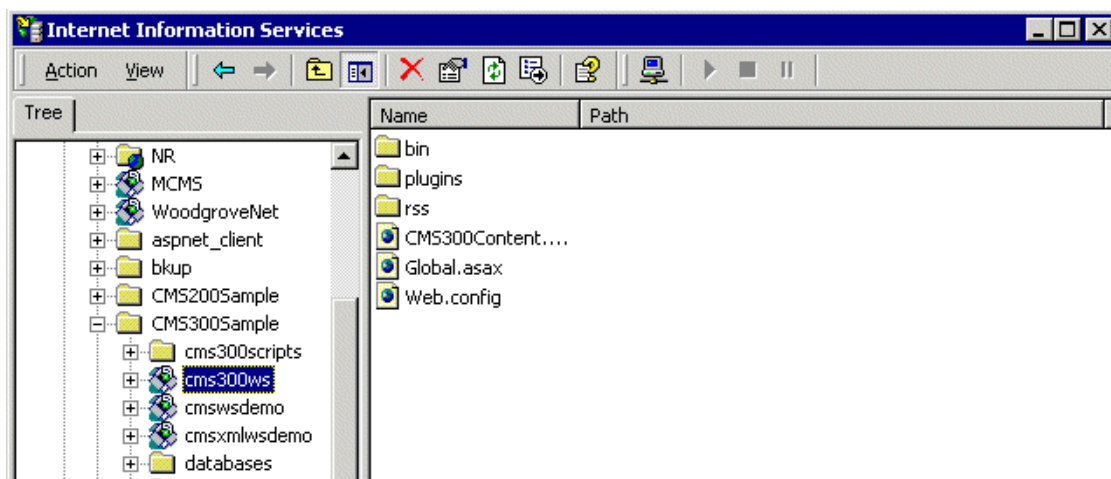
Setup Instructions

After you install **Ektron CMS200/300**, you need to open IIS, select the following folders, and make each an application.

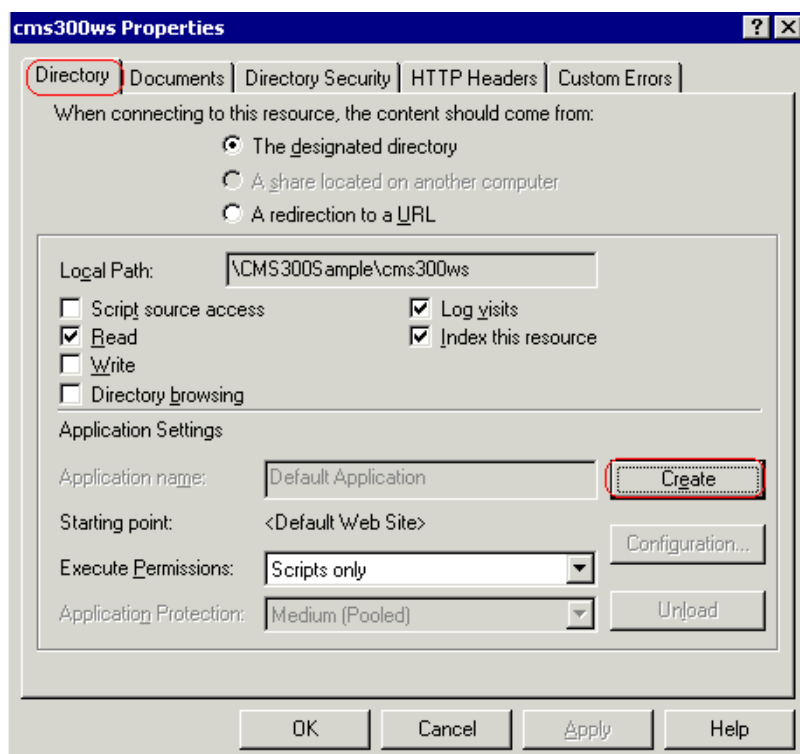
- /CMS300sample/cms300ws
- /CMS300sample/cmswsdemo
- /CMS300sample/cmsxmlwsdemo

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 (circled above).
6. Assign the folder's name in the **Application name** field.
7. Press **OK**.

Providing Web Service Information in Your CMS

Ektron CMS200/300 Methods

Ektron CMS200/300 implements a single Web Service, `cms300content.asmx`, that exposes 13 methods for retrieving content. These methods mirror those in the "applicationApi.asp" 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 cms300content exposes with a brief description of functionality and differences from the applicationAPI implementation.

All methods are demonstrated on the sample page `webroot/cms300sample/cms300ws/cms300content.aspx`. For more information, see ["The Ektron CMS200/300 Content Sample Page" on page 10](#).

Method	Exposes this applicationApi.asp function as a Web Service Method	Returns	Changes to standard parameter values
ContentBlock	ecmContentBlock See also: "Content Block Function" on page 159	The ContentBlock as an HTML string	
ContentBlockEx	ecmContentBlockEx See also: "XML Content Block" on page 163	The ContentBlockEx as an HTML string. If necessary, performs XSLT transformation on content prior to its return.	
GetContentBlock	ecmContentBlock See also: "Content Block Function" on page 159	An XML data structure representing information in the ContentBlock	
Collection	ecmCollection See also: "Collection Function" on page 201	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. Be sure to implement the IDisplayFormatter interface defined in the WSContentDisplay dll. Then, place the dll in the <code>CMS300Sample\cms300ws\plugins</code> directory so the Web Service code can locate it. To access a custom display function, use the syntax <code>"Dllname.Classname"</code> .	

GetCollection	ecmCollection See also: "Collection Function" on page 201	an XML data structure representing information about a Collection	additional parameter available: GetHTML. Possible values: 1 -returns html content for each collection item 0 - does not return it
ListSummary	ecmListSummary See also: "List Summary Function" on page 186	List Summary information as an HTML string	
GetListSummary	ecmListSummary See also: "List Summary Function" on page 186	an XML data structure representing information about the List Summary	parameters ShowSummary, StyleInfo, ShowInfo not available
SingleSummary	ecmSingleSummary See also: "Single Summary Function" on page 195	Single Summary information as an HTML string	
GetSingleSummary	ecmSingleSummary See also: "Single Summary Function" on page 195	an XML data structure representing the Single Summary information	parameters StyleInfo, ShowSummary, ShowInfo not available
MetaData	ecmMetaData See also: "Metadata Function" on page 229	Meta Data information as an HTML string	does not have the unused spare parameter
GetMetaData	ecmMetaData See also: "Metadata Function" on page 229	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 12	Search Display information as an HTML string	
GetSearchDisplay	ecmSearchDisplay See also: "The Search Display Method Parameters" on page 12	An XML data structure representing the Search Display information	parameters StyleInfo and ShowDate not available

The Ektron CMS200/300 Content Sample Page

The content sample page ([webroot/cms300sample/cms300ws/CMS300Content.aspx](#)) 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.

CMS300Content

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"
```


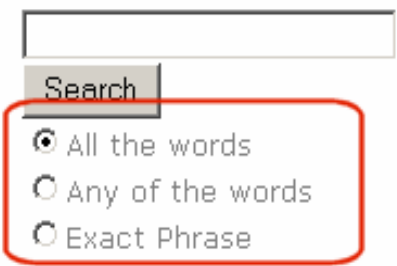
The top section of the page lets you test the method by using the HTTP Get protocol to retrieve data from the sample database. To do this, you must insert valid parameter values. (For documentation of a method's parameters, see the section listed after "see also" in the above table.)

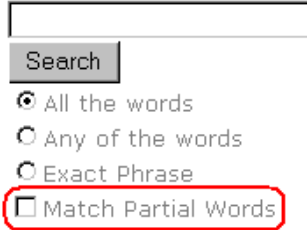
The next section of the content sample page displays the following kinds of sample output.

- 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 function.</p>  <p> <input checked="" type="radio"/> All the words <input type="radio"/> Any of the words <input type="radio"/> Exact Phrase <input type="checkbox"/> Match Partial Words </p>
Search Type	n/a	<p>Type of search used in search function.</p>  <p>Options</p> <p>and - all the words</p> <p>or - any of the words</p> <p>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</p> <p>0 - non-recursive search</p> <p>See Also: "Recursive" on page 77</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</p> <p>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</p> <p>1 - one result</p> <p>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 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:cms300:v21">
  <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@cms300samplesite.com</
contact-email> <contact-fax>1-555-555-5555</contact-fax>    </contact> </
jobposting></ContentHtml>

<Xslt1>C:\Inetpub\wwwroot\CMS300Sample\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/cms300sample/cmswsdemo/`. 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="CmsXmlWsDemo.news" %>

```

This line indicates that the page's functionality is set up in the "CodeBehind" 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 wsCms300WebService As New localhost.CMS300Content()
```

Retrieving the Web Service's URL

Another noteworthy line of news.aspx.vb code, which follows the above line, is:

```
wsCms300WebService.Url = g_strWebServiceUrl
```

This line retrieves the URL of the Web Service pages from the web.config file in the *webroot/cms300sample/cmswsdemo/* folder. As a result, if someone changes the URL in web.config, the news.aspx page's address is automatically updated.

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 = wsCms300WebService.ContentBlock(iId)
    strHTML = Replace(strHTML, ".asp", ".aspx")
    NewsInfo.InnerHtml = strHTML

Else
    strHTML = wsCms300WebService.ListSummary("\Marketing\News", 0, 1, "",
    "Title,asc", "DateModified", 0, "", "")
    strHTML = Replace(strHTML, "news.asp", "CmsWsDemo/news.aspx")
    NewsInfo.InnerHtml = strHTML
End If
```

NOTE

This code converts the file types because the sample database was written for ASP pages, not ASP.NET pages. If the database were created with .NET, you could use this code to get the HTML for the requested ID:

```
NewsInfo.InnerHtml = wsCms300WebService.ContentBlock(iId)
```

As you can see, we are assigning to the variable NewsInfo.InnerHtml the Web service (wsCms300WebService) followed by the ContentBlock method. (For a list of methods, see ["Ektron CMS200/300 Methods" on page 8.](#))

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 first product, RC Sportster. The RC Sportster is designed to compete with the entry and mid level RC racer currently on the market.

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

```
If (iId > 0) Then
  strHTML = wsCms300WebService.ContentBlock(iId)
```

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

CMS300Content

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"
```

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 =
wsCms300WebService.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)

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

The news.aspx sample file in the `/cmd300sample/cmsxmlwsdemo` folder includes all elements described in "[The News.aspx Sample \(non-XML\)](#)" on page 15 *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 wsCms300WebService As New localhost.CMS300Content()  
Dim gcbCms300ContentXml As localhost.ContentBlockResult  
Dim glsCms300ListSummaryResult As localhost.ListSummaryResult
```

Getting Content Blocks

Look at this code from news.aspx.vb.

```
glsCms300ListSummaryResult =  
wsCms300WebService.GetListSummary("\Marketing\News", 0, 1, "", "Title,asc",  
"DateModified", 0, "", "")
```

```

        If (Len(glsCms300ListSummaryResult.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 (glcCms300ListSummaryResult.Item.Length - 1)
strHTML = strHTML & "<tr><td><b>Content ID: </b>" &
        glsCms300ListSummaryResult.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>" &
        glsCms300ListSummaryResult.Item(iLoop).DisplayDateCreated & "</td></
tr>"
strHTML = strHTML & "<tr><td><b>Date Modified: </b>" &
        glsCms300ListSummaryResult.Item(iLoop).DisplayDateModified & "</td></
tr>"
strHTML = strHTML & "<tr><td><b>Modified By: </b>" &
        glsCms300ListSummaryResult.Item(iLoop).EditorFname & " " &
        glsCms300ListSummaryResult.Item(iLoop).EditorLname & "</td></tr>"
strHTML = strHTML & "<tr><td><b>Content Teaser: </b></td></tr>"
        strHTML = strHTML & "<tr><td>" &
        glsCms300ListSummaryResult.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 `glcCms300ListSummaryResult.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 16](#).

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 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>" &
    gcbCms300ContentXml.Item.ContentTitle & "</td></td>"
strHTML = strHTML & "<tr><td><b>Xslt Used for Rendering: </b>"
Select Case (gcbCms300ContentXml.Item.DefaultXslt)
    Case 1
        strTmp = gcbCms300ContentXml.Item.Xslt1
    Case 2
        strTmp = gcbCms300ContentXml.Item.Xslt2
    Case 3
        strTmp = gcbCms300ContentXml.Item.Xslt3
    Case 4
        strTmp = gcbCms300ContentXml.Item.Xslt4
    Case 5
        strTmp = gcbCms300ContentXml.Item.Xslt5
End Select

If (Len(strTmp) = 0) Then
    strTmp = "None Used (HTML Content assumed)"
```

This code uses the `gcbCms300ContentXml` Web Service to get the content title: `gcbCms300ContentXml.Item.ContentTitle`.

The code also uses the `gcbCms300ContentXml` 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>"
```

Earlier in the news.aspx.vb file, the `strRawHtml` variable was defined as `strRawHtml = gcbCms300ContentXml.Item.ContentHtml`. So, the `strRawHtml` variable uses the `gcbCms300ContentXml` Web Service to retrieve content in “raw” HTML format.

Here is the resulting section of that page. As you can see, HTML tags are embedded within the content.

Raw HTML (or XML) from GetContentBlock Web Service:

```
<h4>RC International Announces the RC Cheetah</h4> <p><strong>January 14, 2001</strong> - 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.</p> <blockquote> <p><em>"The RC Cheetah price/performance ration is simply incredible",</em> stated  
Bob Dickerson, RC International VP of engineering. <em>"The RC Cheetah has been devastating its competition in  
all the pre-release races and we expect that to continue..."</em></p> </blockquote> <p>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:<br /> </p> <blockquote><em>"There is no comparison between  
the RC Cheetah and its nearest competitor. In speed, handling, and price the RC Cheetah out shines them  
all".</em></blockquote> <p>The RC Cheetah is now available at all RC International distributors and major Hobby  
stores. Please contact RC International to locate a local dealer.</p>
```

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 = wsCms300WebService.ContentBlock(iId)`. So, the `strProcessedHtml` variable uses the `wsCms300WebService.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 `wsCms300WebService.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 "[Ektron CMS200/300 Methods](#)" on page 8 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 `CMS300sample` 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/CMS300sample/cmswsdemo` and `wwwroot/CMS300sample/cmsxmlwsdemo` as examples. Rename and modify them as needed.

5. Add to each .NET project a Web Reference to the WSDL file. (by default, `webroot/cms300sample/cms300ws/cms300content.asmx?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. **Ektron CMS200/300'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 CMS200/300 for ASP.NET. RSS syndication allows you to prepare Ektron CMS200/300 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 CMS200/300, 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 CMS200/300

Ektron CMS200/300 allows you to syndicate content by either using ASP or .Net. Syndicating via .Net requires that you have the Microsoft .Net framework installed on your computer.

With Ektron CMS200/300, you can syndicate a collection and a list summary via RSS.

The following sections explain how to configure Ektron CMS200/300 to syndicate content via ASP and .Net.

Syndicating via ASP

Syndicating via ASP requires you to create a news feed type file in ASP calling a RSS function defined in Ektron CMS200/300' applicationAPI.asp file.

The following section explains each of the RSS functions, as well as the steps required to syndicate your Web site content.

RSS Functions

The file ApplicationAPI.asp has included two additional functions used to syndicate collections and summary lists on your Web site.

The two functions are named:

- ecmRssCollection
- ecmRssSummary

By creating news feed files that call these two functions, you can easily syndicate content.

Explained in the following section are the functions.

ecmRssCollection

Displayed here is the function, ecmRssCollection:

```
Function ecmRssCollection(id)
    dim rssCont,dictobj,ErrorString
    dim rStr,srvName,scriptName,qStr
    Set dictobj = Server.CreateObject("Scripting.Dictionary")
    qStr = Request.ServerVariables("URL")
    srvName = Request.ServerVariables("SERVER_NAME")
    'scriptName = Request.ServerVariables("SCRIPT_NAME")
    dictobj.Add "Url", qStr
    dictobj.Add "ServerName", srvName
    dictobj.Add "ColId", id
    if (Request.ServerVariables("QUERY_STRING") <> "" ) then
        dictobj.Add "UrlParam", Request.ServerVariables("QUERY_STRING")
    else
```

```
        dictobj.Add "UrlParam",""
    end if
    Set rssCont = Server.CreateObject(CONTENT_OBJ)
    rStr = rssCont.GetRssCollectionByID(AppConfStr,
dictobj,Request.Cookies("ecm")("site_id"),ErrorString)
    ecmRssCollection = rStr
    Set dictobj = nothing
    Set rssCont = nothing
end function
```

When called, this function only required on parameter, the ID of the collection to be syndicated.

ecmRssSummary

Displayed here is the ecmRssSummary function:

```
Function ecmRssSummary(Folder, Recursive, OrderBy, MaxNumber)
    Dim ContentObj, SumObj, rStr, ErrorString
    dim srvName,url,qStr,order

    'UrlParam = Request.ServerVariables("QUERY_STRING")
    url = Request.ServerVariables("URL")
    srvName = Request.ServerVariables("SERVER_NAME")
    Set ContentObj = Server.CreateObject(CONTENT_OBJ)
    Set SumObj = Server.CreateObject("Scripting.Dictionary")
    SumObj.Add "Folder", Folder
    SumObj.Add "Recursive", Recursive
    SumObj.Add "MaxNumber", MaxNumber
    SumObj.Add "ObjType", "Folder"
    SumObj.Add "PreviewMode", false
    SumObj.Add "Url", url
    SumObj.Add "ServerName", srvName
    if (Request.ServerVariables("QUERY_STRING") <> "" ) then
        SumObj.Add "UrlParam", Request.ServerVariables("QUERY_STRING")
    else
        SumObj.Add "UrlParam",""
    end if
    if( lcase(OrderBy) = "date") then
        order = "DateModified"
    else
        order = "Title,asc"
    end if

    SumObj.Add "OrderBy", order
    rStr = ContentObj.GetRssSummary(AppConfStr,SumObj, Request.Cookies("ecm")("site_id"),
ErrorString)
    ecmRssSummary = rStr

End Function
```

Similar to the ecmRssCollection function, this function requires one parameter to be passed, “folder,” however, there are three additional parameters that can be used as well:

- Recursive
- OrderBy
- MaxNumber

Syndicating Collections

To syndicate a collection on your Web site, Ektron recommends that you create a new file in your Ektron CMS200/300 site root folder that contains all the information needed to syndicate the content.

Once the file is created, you may distribute the link to the newly created file where users may create new news channels in their news feeder programs to successfully display the content.

Example

Ektron CMS200/300 supplies a sample file, `/CMS200/300Sample/productsfeed.asp`, which syndicates a collection. The parameter for the collection is hard-coded within the file.

Displayed here is the file `productsfeed.asp`.

```
<%option explicit%>
<!-- #include file="site_scripts_path.asp" -->
<%
'Need to specify collection ID number in this file
dim cID,rssData
    'ecmRssCollection(1) will return collection id 1 from the CMS database
    rssData = ecmRssCollection(1)
    'write the data out to the port
    response.write(rssData)
%>
```

As you can see, a very minimal amount of code is needed to syndicate content. However, all of the displayed code is required within your news feed files to properly function.

Creating Your Own Collection News Feed Files

In the previous section, the code required to syndicate content using RSS was displayed.

To create your own news feed file to syndicate a collection on your Web site, perform the following steps;

1. Open your ASP editing tool
2. Enter the following text:

```
<%option explicit%>
<!-- #include file="site_scripts_path.asp" -->
```

Be sure to specify the proper location of the file "site_scripts_path.asp" for the include file. For this example, we are saving this file in the same directory as the include file.

3. Enter the following ASP code:

```
<%
    dim cID,rssData
    rssData = ecmRssCollection(cID)
    response.write(rssData)
%>
```

Replace the cID parameter with the ID of the collection you wish to syndicate (for example, "2").

NOTE

For information about retrieving the ID number for a collection, refer to "Collections" in the Ektron CMS200/300 Administrator Manual.

Displayed here is another example of a news feed file that syndicates a summary list:

```
<%option explicit%>
<!-- #include file="site_scripts_path.asp" -->
<%
    dim cID,rssData
    rssData = ecmRssCollection(2)
    response.write(rssData)
%>
```

Using this example displays the collection ID=2 in the news feeder application.

4. Save the file as an ASP file in the same directory as the file "site_scripts_path.asp"

Now that the news feed file has been successfully created, you may distribute the link to the file. When the file is called in the

user's news feeder application, the collection defined in the file will be displayed.

Configuring News Feed Application

Once you have created and configured your own news feed file, consumers may configure their news feed application to view the content.

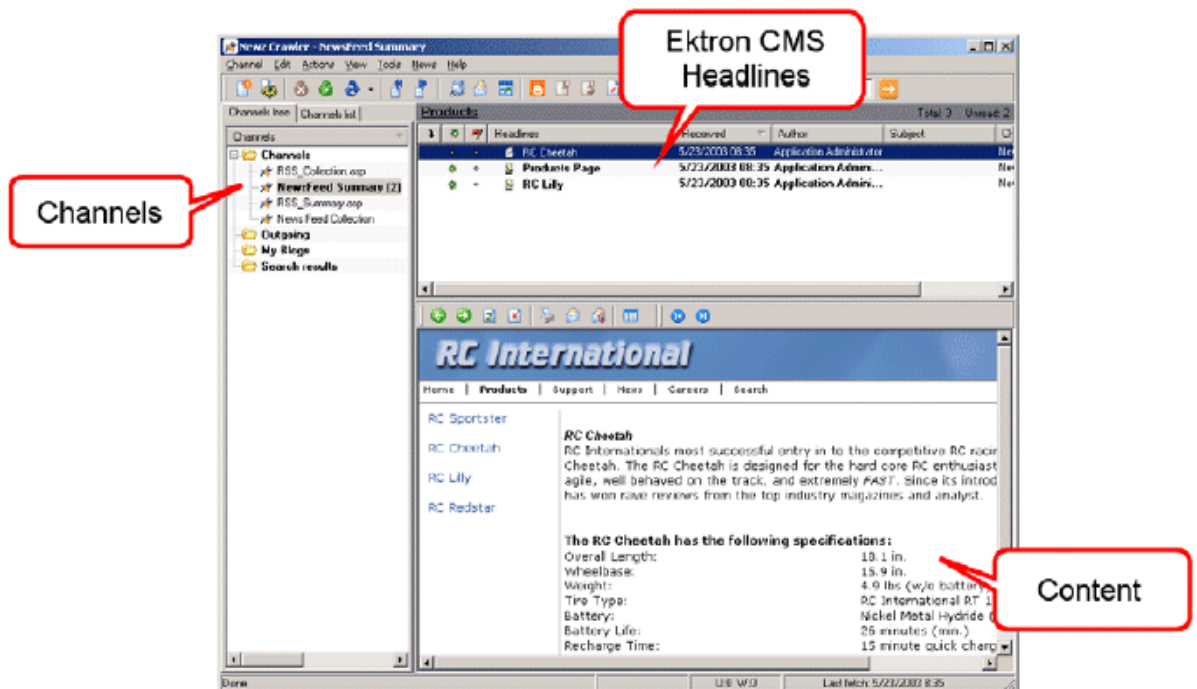
Simply send them the link to the news feed file, for example:

`http://www.YourSite.com/CMS200/300sample/productsfeed.asp`

Once they have added the file as a new channel, the content is successfully syndicated to them.

Displayed here is the example displayed above in the news feed application, Newz Crawler.

NOTE Depending on news feed application, the display and format may appear different.



As you can see, each content that matches the criteria specified in your news feed file is displayed.

Alternate Dynamic Method

NOTE

Ektron recommends that you use the previous method to syndicate content via RSS. Using the previous method gives you more control on what content is syndicated.

Instead of hard-coding the collection ID into the file, you may configure the file to allow consumers to specify any ID they desire. When the consumer adds the file to their news feed application, they simply specify the parameter within the URL.

Displayed here is sample code to allow this:

```
<%option explicit%>
<!-- #include file="site_scripts_path.asp" -->
<%
'this page expects one URL parameter which is cId. This cId is the
collection ID.
dim cID,rssData
    'Get cID from URL
    cID = request.QueryString("cId")
    rssData = ecmRssCollection(cID)
    'write the data out to the port
    response.write(rssData)

%>
```

When this code is used in a news feed application, the user must also add the URL parameter **?cid=#** to the end of the URL.

Displayed here is an example:

<http://www.YourSite.com/CMS200/300sample/productsfeed.asp?cID=1>

This URL will display the collection with an ID of 1 in the news feeder application.

NOTE

If this method is used, and no parameter is passed through the URL, the user may receive an error.

Syndicating Summary Lists

To syndicate a summary list on your Web site, Ektron recommends that you create a new file in your Ektron CMS200/300 site root

folder that contains all the information needed to syndicate the content.

Once the file is created, you may distribute the link to the newly created file where users may create new news channels in their news feeder programs to successfully display the content.

Example

Ektron CMS200/300 supplies a sample file

/CMS200/300Sample/newsfeed.asp

which syndicates a summary list. The parameters for the summary list are hard-coded within the file.

Displayed here is the file newsfeed.asp.

```
<%option explicit%>
<!-- #include file="site_scripts_path.asp" -->
<%
    dim rssData
    rssData = ecmRssSummary("\Marketing\News", 1, "date",0)
    'write the data out to the port
    response.write(rssData)
%>
```

As you can see, a very minimal amount of code is needed to syndicate content. However, all of the displayed code is required within your news feed files to properly function.

Creating Your Own Summary News Feed Files

In the previous section, the code required to syndicate content using RSS was displayed.

To create your own news feed file to syndicate a summary list on your Web site, perform the following steps;

1. Open your ASP editing tool
2. Enter the following text:

```
<%option explicit%>
```

```
<!-- #include file="site_scripts_path.asp" -->
```

Be sure to specify the proper location of the file "site_scripts_path.asp" for the include file. For this example, we are saving this file in the same directory as the include file.

3. Enter the following ASP code:

```
<%
    dim rssData
    rssData = ecmRssSummary(Folder, Recursive, OrderBy, MaxNumber)
    response.write(rssData)
%>
```

Use the following table as a guide to properly enter the correct parameters in the ecmRssSummary functions.

Parameter	Description	Options/Examples
Folder	Specify the content folder in Ektron CMS200/300 where the summaries will be called from.	"\" = Root content folder "Products" = Products folder
Recursive	Specify whether summaries are called from the starting folder's sub-folders as well.	0 = Not Recursive 1 = Recursive
OrderBy	Specify the order in which the results are displayed in the news feeder application.	"Date" = chronologically "title" = alphabetically
MaxNumber	Specify the maximum amount of results to display.	0 = All 4 = four 13 = thirteen

Displayed here is another example of a news feed file that syndicates a summary list:

```
<%option explicit%>
<!-- #include file="site_scripts_path.asp" -->
<%
    dim rssData
    rssData = ecmRssSummary("\Products", 1, "title", 3)
    response.write(rssData)
%>
```

This example has the following properties:

- Summaries in the \Products content folder, as well as its sub-folders, are displayed
 - The results are sorted alphabetically by title.
 - Only the first three results are displayed.
4. Save the file as an ASP file in the same directory as the file "site_scripts_path.asp"

Now that the news feed file has been successfully created, you may distribute the link to the file. When the file is called in the user's news feeder application, the summary list defined in the file will be displayed.

Configuring News Feed Application

Once you have created and configured your own news feed file, consumers may configure their news feed application to view the content.

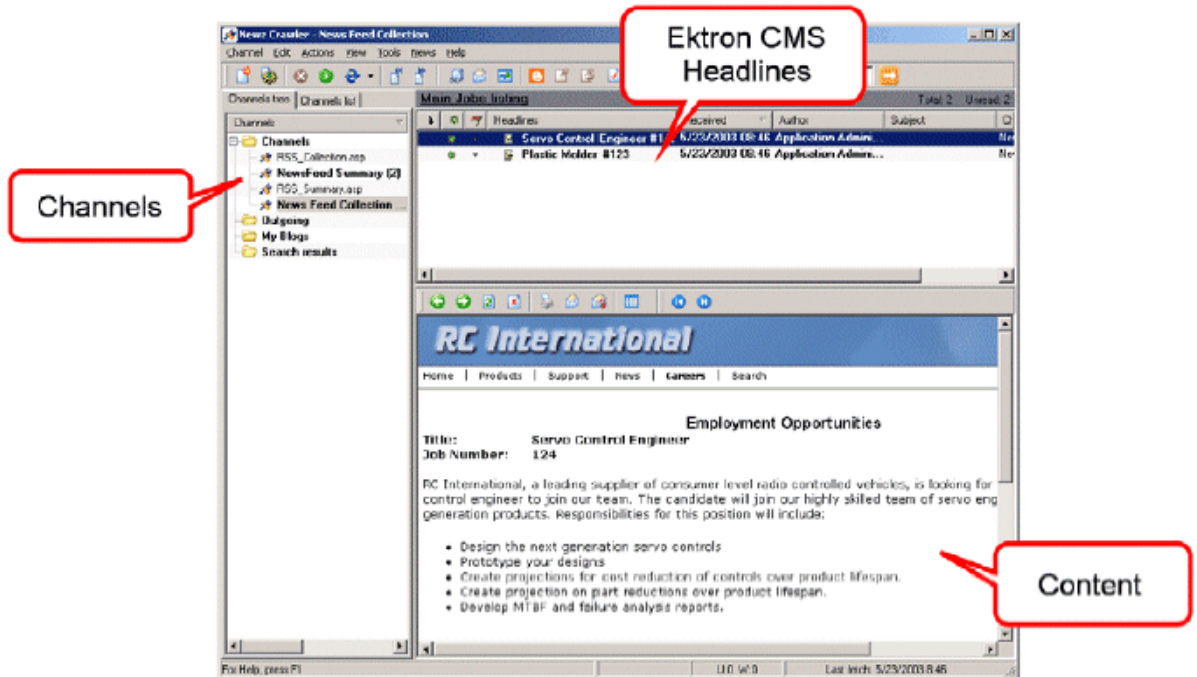
Simply send them the link to the news feed file, for example:

`http://www.YourSite.com/CMS200/300sample/newsfeed.asp`

Once they have added the file as a new channel, the content is successfully syndicated to them.

Displayed here is the example displayed above in the news feed application, Newz Crawler.

NOTE Depending on news feed application, the display and format may appear different.



As you can see, each content that matches the criteria specified in your news feed file is displayed.

Alternate Dynamic Method

NOTE Ektron recommends that you use the previous method to syndicate content via RSS. Using the previous method gives you more control on what content is syndicated.

Instead of hard-coding the summary parameters into the file, you may configure the file to allow consumers to specify any parameters they desire. When the consumer adds the file to their news feed application, they simply specify the parameters within the URL.

Displayed here is sample code to allow this:

```
<%option explicit%>
<!-- #include file="applicationAPI.asp" -->
<%
dim Folder, Recursive, OrderBy, MaxNumber, rssData
'Get parameters from URL
```

```

Folder = request.QueryString("Folder")
Recursive = request.QueryString("Recursive")
OrderBy = request.QueryString("OrderBy")
MaxNumber = request.QueryString("MaxNumber")
'Call ecmRssSummary function
rssData = ecmRssSummary(Folder, Recursive, OrderBy,MaxNumber)
'write the data out to the port
response.write(rssData)
%>

```

When this code is used in a news feed application, the user must also specify at least the folder parameter, and may also enter three others.

The following table explains each of the possible parameters.

Parameter	Description	Examples
Folder <i>Required!</i>	Specify the content folder in Ektron CMS200/300 From which the summaries will be called. <u>IMPORTANT: Due to escaping issues, a backslash("\") must be replaced by "%5c" in the URL. for example, %5cMarketing%5cNews instead of \marketing\news</u> <u>This may not work in all news feed applications!</u>	folder=%5cMarketing%5cNews folder=%5c
Recursive	Specify whether summaries are called from the starting folder's sub-folders as well.	recursive=1 recursive=0
OrderBy	Specify the order in which the results are displayed in the news feeder application.	orderby=date orderby=title
MaxNumber	Specify the maximum number of results to display.	maxnumber=0 maxnumber=32

Here is an example of a URL string that contains all parameters:

```

http://www.YourSite.com/CMS200/300sample/
newsfeed.asp?folder=%5cProducts&recursive=1&orderby=title&maxnumber=5

```

This example has the following properties:

- Summaries in the `\Products` content folder, as well as its sub-folders, are displayed
- The results are sorted alphabetically by title
- Only the first five results are displayed

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 pages ([news.aspx](#), [jobs.aspx](#), and [products.aspx](#)) are installed by default to `webroot/CMS300Sample/CMS300WS/rss` folder. They use the CMS300WS 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 sample, you should modify them or create your own aspx pages for your Web site. Place your custom pages in the `/cms300ws/rss/` or `/cms300ws/` folder so that the web.config file is within the path.

news.aspx

The news.aspx page is hardcoded to retrieve the `/Marketing/News` list summary in the sample.

The news.aspx page accepts one optional URL parameter, max, which specifies the maximum number of items to return. If the max parameter is missing or its value is zero (0), all items are returned.

Sample URLs:

```
http://localhost/cms300sample/cms300ws/rss/news.aspx  
http://localhost/cms300sample/cms300ws/rss/news.aspx?max=10
```

jobs.aspx

The jobs.aspx page returns a list of job openings in the sample application. In the sample, collection id 2 is the list of job openings.

Sample URL:

```
http://localhost/cms300sample/cms300ws/rss/jobs.aspx
```

products.aspx

The products.aspx page returns a list of products in the sample application. In the sample, collection id 1 is the list of products.

Sample URL:

```
http://localhost/cms300sample/cms300ws/rss/products.aspx
```

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 CMS300WS.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 HttpResponseMessage.

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 CMS300WS.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 HttpResponseMessage.

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()
```

Custom ASP Functions

Ektron CMS200/300 maintains and displays content through the use of custom Ektron ASP functions and ComObjects. On your template, you must include the following file in order to be able to use the custom functions:

```
<!-- #include file="site_scripts_path.asp" -->
```

This file (**site_scripts_path.asp**) will call the file with the custom Ektron functions to allow the page to function properly and must appear at the top of every template file that uses an Ektron CMS200/300 custom function.

To customize your Ektron CMS200/300 application, you will need to be familiar with these tags:

Function	Description	More Information
Add Content	Allows you to programmatically add content to Ektron CMS200/300 from a simple function.	"Add Content" on page 56
Advanced XML Search	Displays an advanced XML search on the Web page.	"Advanced XML Search" on page 66
Basic XML Search	Displays a basic indexed XML search on the Web page.	"Basic XML Search" on page 61
XML Search Display	Displays the results when either a basic or advanced XML search is performed.	"XML Search Display" on page 72
Collection	Displays the links generated by collections.	"Collection Function" on page 100
Content Block	Displays and manages a content block.	"Content Block Function" on page 48
Event Calendar	Displays an event calendar	"Event Calendar Function" on page 115
Form	Displays a content block that is associated with a form.	"Form Function" on page 121

Function	Description	More Information
List Summary	Displays a list of content block summaries.	"List Summary Function" on page 88
Login	Adds the login or logout button which allows the user to login and use the Ektron CMS200/300 application.	"Login Function" on page 44
Metadata	Displays the metadata in the source of the Web page the content is displayed on.	"Metadata Function" on page 124
MultiLevel Collection	Customize the DHTML menus feature	"Multi-Level Collection" on page 133
Multipurpose	Can display either a content block or a form	"MultiPurpose Function" on page 123
Random Content	Randomly displays the body content of a collection item.	"Random Content" on page 113
Random Summary	Randomly displays the summary content of a collection item.	"Random Summary" on page 114
Search	Inserts a text box on the template with a search button.	"Search Function" on page 74
Search Display	Creates an area where search results will be displayed.	"Search Display Function" on page 83
Single Summary	Displays the summary for a single content block.	"Single Summary Function" on page 96
ShowSiteLanguage	Lets a site visitor select a language in which to view site.	"Show Site Language Function" on page 131



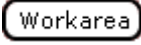
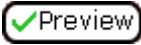

Login Function

The login function adds a login button, on the template the login function is inserted on, when displayed in a browser.

Displayed here is the format for the login function.

```
<% ecmLogin %>
```

This tag is responsible for displaying the following buttons on the Web page:

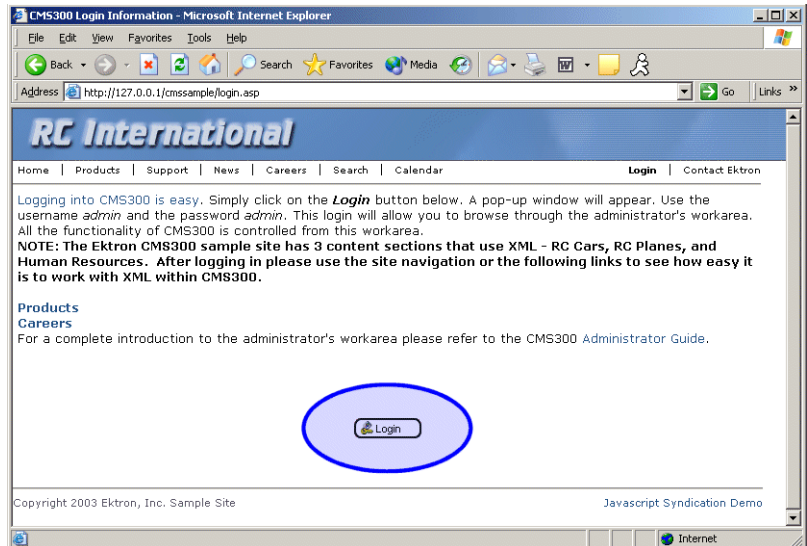
Button	Name	Description
	Login	When user is not logged into Ektron CMS200/300, this button will be displayed. Clicking this button will allow the user to log into the application.
	Logout	Once logged in, this button will replace the login button to allow the user to log out from the Ektron CMS200/300 Web site.
	Workarea	When logged in, this button will be displayed under the logout button allowing the user to access their workarea.
	Preview On	This button will allow the user to preview the entire Web site as if all checked-in content was published.
	Preview Off	Turns off the sites preview mode.

Login Tag Example

The following example places a login button in a table cell inside an asp template.

```
<tr>
  <td>
    <% ecmLogin %>
  </td>
</tr>
```

When inserted properly, a login button will be placed on the Web page.



Clicking on the login tag will open the login window where a user will be able to enter their username and password. Upon successful completion, the user will then be logged into the Ektron CMS200/300 Web site.

Placing Your Login Button

You are able to add as many login buttons to a template as you would like. You might want to include a login button on each template, or have a special Web page called login.asp where you can log from without the public being able to access the login page.



Partner Login

Ektron CMS200/300 also includes a special login function that only displays a login or logout button on the template. Here is the format for this special login function:

```
<% ecmLoginPartner %>
```

As stated before, when inserted onto a template, only a login or logout button is displayed.

The following table displays how the function appears when the user is logged in or logged out.

Logged Out	Logged In
<div>administrator's workarea please refer to the</div> <div></div> <div></div>	<div>administrator's workarea please refer to the CM:</div> <div></div> <div></div>

Content Block Function

The content block tag is used to display a content block on an Ektron CMS200/300 Web page.

Shown here is the format for the content block function:

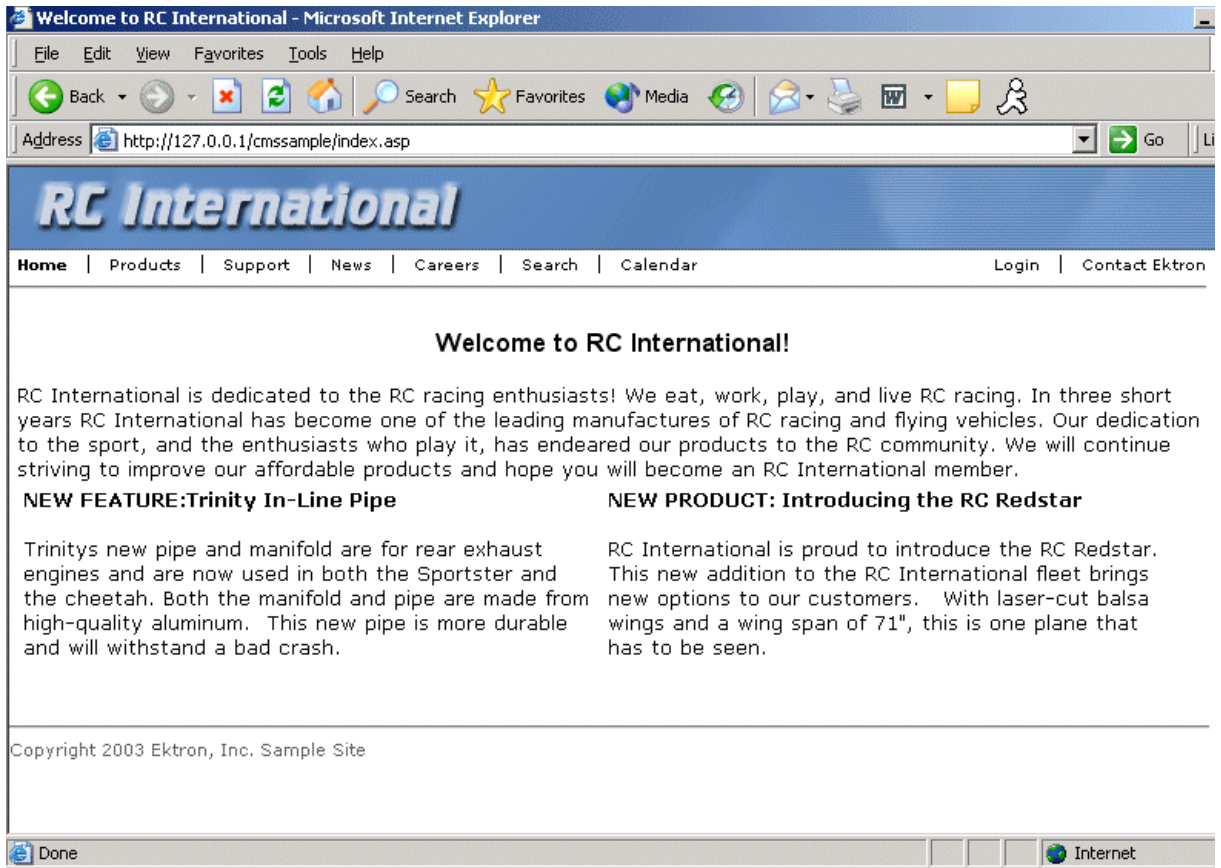
```
<% ecmContentBlock ID,
                    "XSLT",
                    Override %>
```

NOTE As an alternative, you can use the multipurpose function, which can display either a content block or a content block associated with a form. For more information, see ["MultiPurpose Function" on page 123](#).

Explained in the table below are the attributes for the content block function.

Attribute	Description	Required
ID	ID number assigned to a content block when created in Ektron CMS200/300.	Yes
XSLT	File name and path to a valid XSLT to transform the XML content. This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.	No
Override	If no XSLT is specified, then enter 1 to use XSLT1, 2 for XSLT2, or 3 for XSLT3.	No

In normal view, this tag retrieves the content block from the database and displays it in the calling template.



If a user is logged into Ektron CMS200/300, the content block tag displays a border around the content block and provides a menu for managing it.



Content Block Examples

Ektron CMS200/300 has three types of content blocks.

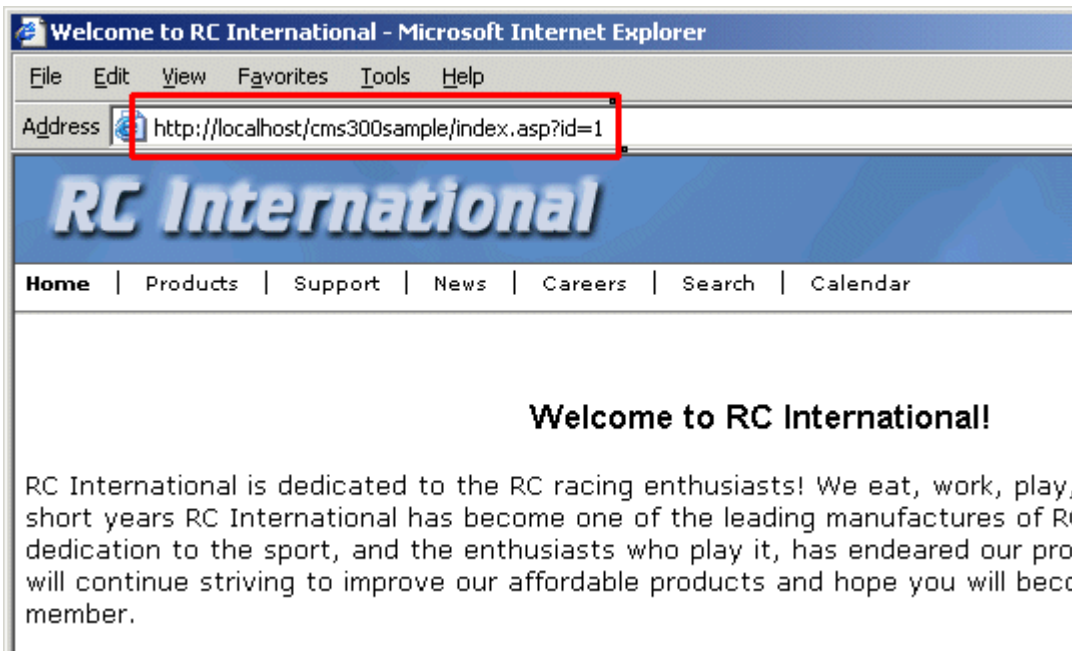
Type	Displays	More information
Static	One content block on the Web page	"Static Content Block" on page 51
Dynamic	A content block whose ID is passed through the URL	"Dynamic Content Block" on page 52
XML	XML content transformed by an XSLT specified in the function	"XML Content Block" on page 52

Static Content Block

The following illustrates how to insert a static content block on a template.

```
<tr>
  <td>
    <% ecmContentBlock(1) %>
  </td>
</tr>
```

In the example above, the function calls the content block with an id of 1 from the database to the browser.

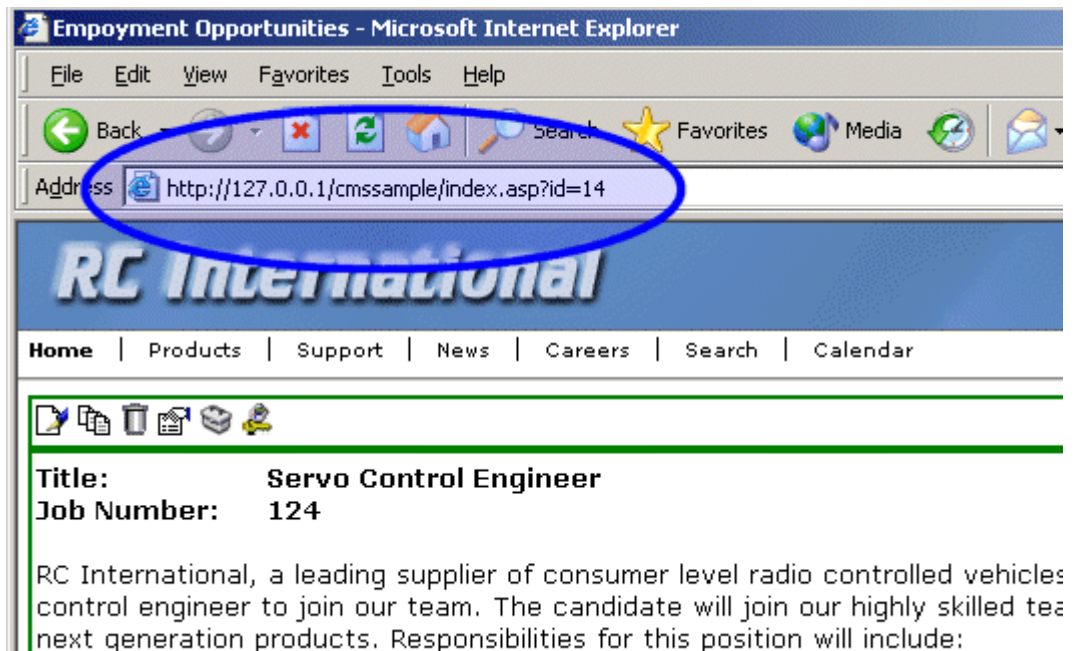


Dynamic Content Block

The following example shows how the content block tag can be used where the id is being passed in as a URL parameter. This is the tag which would be used on a dynamic template.

```
<tr>
  <td>
    <% ecmContentBlock(request.QueryString("id")) %>
  </td>
</tr>
```

When a user accesses this page and adds the ID number to the template this tag lives on, the content block with the corresponding ID appears on the Web page in the specified template.



XML Content Block

NOTE This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

Here is an example of displaying an XML content block. Note that you specify both a content block and an XSLT that determines how the content block is displayed.

```
<tr>
  <td>
    <% ecmContentBlockEx 13, "", 1 %>
  </td>
</tr>
```

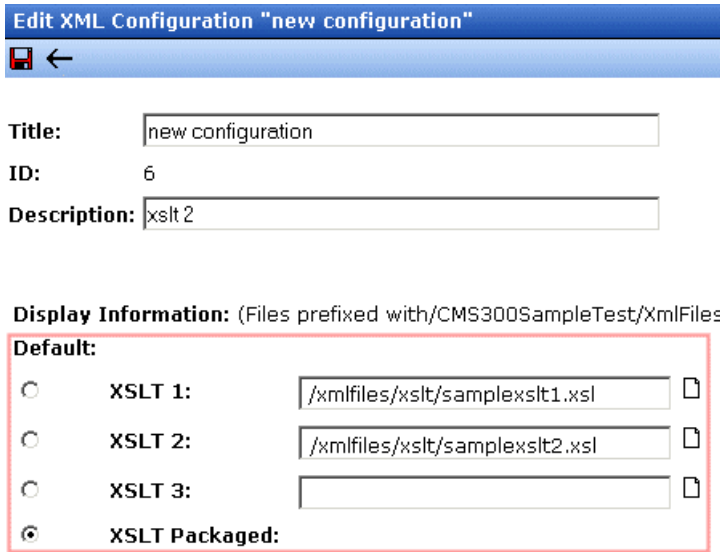




This example would display the content block ID=13, and use the XSLT1 as the display XSLT.

The Function's Arguments

The `ecmContentBlockEx` takes three arguments. The first argument identifies the content block to display. The second one is optional and, if used, specifies an external XSLT file.

If the second argument does not exist, the third argument specifies an XSLT identified in the Edit XML Configuration screen. The

following table provides additional detail about the second and third arguments.

To use this display XSLT	Enter this for argument 2	Enter this for argument 3
<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>Edit XML Configuration "new configuration"</p> <p>   </p> <p> Title: <input type="text" value="new configuration"/> ID: <input type="text" value="6"/> 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"/>  <input type="radio"/> XSLT 2: <input type="text" value="/xmlfiles/xslt/samplexslt2.xsl"/>  <input type="radio"/> XSLT 3: <input type="text" value=""/> <input checked="" type="radio"/> XSLT Packaged: </p>	""	none
XSLT Packaged: from the Edit XML configuration screen (that is, the XSLT created in the Data Designer)	""	0
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 for argument 2	Enter this for argument 3
An absolute or relative path to an XSLT	<p>An external XSLT file not specified in the Edit XML configuration screen.</p> <p>For example</p> <pre><% ecmContentBlockEx 13, "samplexslt.xsl"%></pre>	<p>does not matter</p> <p>- if a value exists in argument 2, argument 3 is ignored</p>

See Also: The “Managing XML” chapter of the **Ektron CMS200/300** Administrator manual

Dynamic and Static Combination

Here is an example of how you can use the two different content tags in the same table cell by using an IF statement

```
<tr>
  <td>
    <% if request.QueryString("id")<> "" then
      ecmContentBlock(request.QueryString("id"))else
      ecmContentBlock(14) end if %>
  </td>
</tr>
```

In this sample code, when a person accesses this page without passing through the content's ID number, the content with an ID=14 will be displayed. If you pass a content id through the URL like **index.asp?id=4**, the content block ID=4 will be displayed.

Add Content

The add content function allows you to programmatically add a new content block to Ektron CMS200/300 without the need of logging in.

This function can be for several applications including, but not limited to:

- Applied to a Web form allowing you to collect data from users about the content block.
- Creating a loop function that moves information from a database to an Ektron CMS200/300 database.

Here is the format for the add content function:

```
<% AddContent    Username,  
                 Password,  
                 Domain,  
                 content_title,  
                 content_comment,  
                 ContentHtml,  
                 SummaryHTML,  
                 MetaData Title  
                 MetaData Keyword  
                 ContentLanguage,  
                 FolderID,  
                 StartDate,  
                 EndDate,  
                 ErrString                    %>
```


Explained in the table below are the attributes for the add content block function.

Attribute	Description
Username	The username of the user to be authenticated for permissions to see whether the user is able to add a content block.
Password	The password for the username being authenticated.
Domain	If using Active Directory, specify the domain needed to authenticate the user.
Content_title	Specify a title for the content block being added.
Content_comment	Specify a historical comment for the content being added.
ContentHTML	Actual content for the content block.
SummaryHTML	Summary information about the content block.
Metadata_title	Specify a title for the metadata being added.
Metadata_keywords	Specify the information for the metadata which will be used as a reference point for finding other words or information.
ContentLanguage	Language of content.
FolderID	ID number of the content folder where the content block will be added to.
StartLive	Date and time the content will go live. Must match date/time format set in setup.asp.
EndDate	Date and time the content will expire. Must match date/time format set in setup.asp.
ErrString	String returned if the content fails to get added to Ektron CMS200/300.

Add Content Example

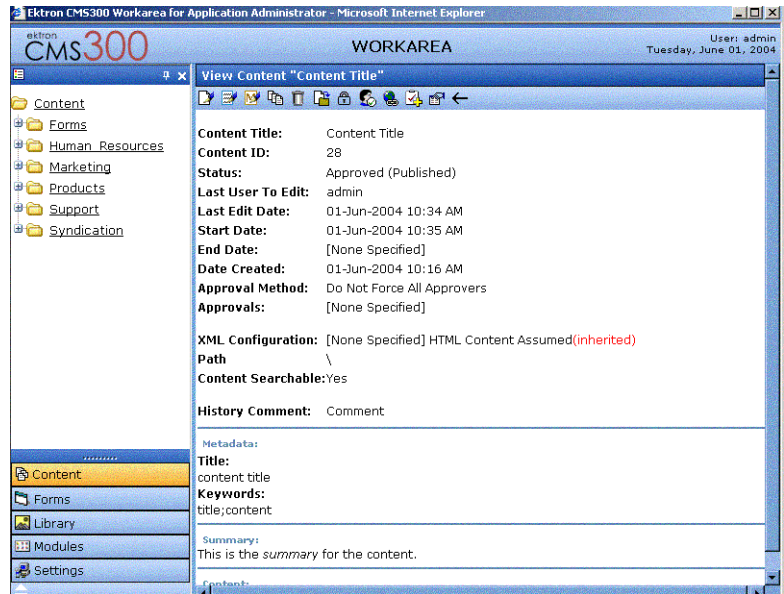
The following is an example of the add content function hard coded in a template:.

```
<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, 28,
      "06-01-2004 10:35 AM", "",
      "<metadata><meta id=""3"">ContentTitle</meta></metadata>",
      "" %>
    </td>
  </tr>
```

The previous example has the following properties:

- The user jedit is used to be authenticated.
- The title of the new content block is “Control 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 = 1, or English.
- The new content block is added to content folder ID=28.
- The start date is set at June 1, 2004 at 10:35 AM.
- No end date is specified.
- The metadata “Content title” is set for the metadata definition ID=3.

When this function is executed, the content is added to Ektron CMS200/300. Displayed here is the record of the content in the workarea.



Add Content Web Form Example

Displayed here is an example of a Web form that is included with Ektron CMS200/300. This form can be found by default in the Ektron CMS200/300 root directory.

Add Content Programatically To CMS

Username:

Password:

Domain: (when using Active Directory)

Title: (as text)

Comment: (as text)

Content: (as HML content)

Summary: (as HML content)

Meta Data (Title): (as text)

Meta Data (Keywords): (separated by ;)

Language: (1 for English, 1031 for German, 1036 for French)

FolderID: (numeric, 0 for root)

Start Date: (as mm-dd-yy)

End Date: (as mm-dd-yy)

The form is set up to pass the data into the addcontent function.

Basic XML Search

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

The basic XML search tag is used to display a text box, with a search button on the template. Unlike the regular Ektron CMS200/300 search tag, this search box allows you to search XML data in your Web site.

When used, the basic XML search returns results that match all indexed content from every indexed XML configuration. Using the advanced XML search function allows users to filter and narrow their search results.

Shown here is the format for the ecmSearch tag:

```
<% ecmFTSearch      "postpage",
                    inputsize,
                    inputmax,
                    "buttonimage",
                    "buttontext"      %>
```

The following table provides information about all the variables for the search tag.

Attribute	Description
Post page	The Web page to output the search results.
Input size	The size of the display text box for user input.
Input max	The maximum characters the input text box should accept.
Button image	The image location for the search button. Blank ("") = standard submit button.

Attribute	Description
Button Text	The text used for the button if the standard submit button is chosen, or alternative text for the button if an image is chosen

Basic XML Search Tag Example

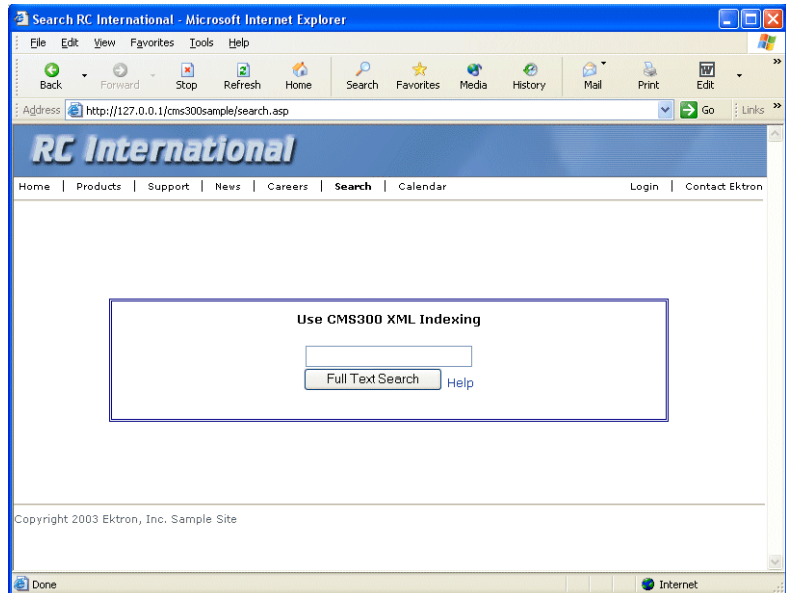
The following example places a text field and a search button, which allows you to search XML data, in a table cell inside an ASP template .

```
<tr>
  <td>
    <% ecmFTSearch "ftsearch-display.asp",25,200,"","Full Text Search" %>
  </td>
</tr>
```

The previous example has the following properties:

- The target page is **"ftsearch-display.asp"**
- The size of the text box is 25.
- The search will allow a maximum of 200 characters.
- No image is used for a button, instead a standard submit button is used.
- The text on the submit button is **"Full Text Search"**.

When added to a template, and visited, the basic XML search tag will appear similar to the following:



XML Search Tag Variables

Listed below are all the different variables and their descriptions.

Post Page

The target page is the page where the search results will be displayed. You may enter any form of a URL for the Web page you will have the search result displayed to.

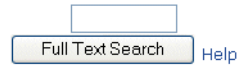
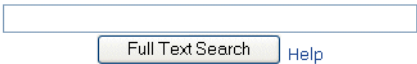
You can enter any Web page that has the **FTSearchDisplay** tag inserted in it. If you want to use the same page, you can enter “#” into the field.

NOTE

Target page **MUST** have the **FTSearchDisplay** function inserted in it for the search results to be displayed.

Text Box

This variable will set the size of the text box that is displayed. This can be any number depending on how big you would like the search text box to be.

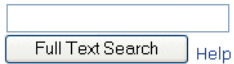
Text Box Size	Web Page View
10	<p>Use CMS300 XML Indexing</p> 
50	<p>Use CMS300 XML Indexing</p> 



Maximum Characters

The value in this attribute determines the amount of characters.

Button Image Source

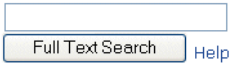
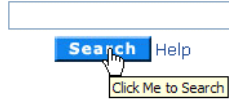
If you would like to use an image for the search button instead of the default, you may enter the filename and path of the image.

Button Image Source	Web Page View
""	<p>Use CMS300 XML Indexing</p> 

Button Image Source	Web Page View
"search.jpg"	<p>Use CMS300 XML Indexing</p> 
"http://www.ektron.com/images/search.jpg"	<p>Use CMS300 XML Indexing</p> 

Button Text

If no search button image was specified, then the text that is entered in this field will be the text that appears on the button. If an image is being used for the search, then this text will be the alt text for the button.

Button Text	Web Page View
"Full Text Search"	<p>Use CMS300 XML Indexing</p> 
"Click Me to Search"	<p>Use CMS300 XML Indexing</p> 

Advanced XML Search

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

The advanced XML search function displays an advanced search form which is dynamically created using the information set when you indexed your XML configuration(s).

Shown here is the format for the ecmSearchAdvanced tag:

```
<% ecmFTSearchAdvanced      "postpage",
                             "submitbuttonimage",
                             "submitbuttontext",
                             "resetbuttonimage",
                             "resetbuttontext",
                             Template                                %>
```

The following table provides information about all the variables for the search tag.

Attribute	Description
Post page	The Web page to output the search results.
Submit Button image	The image location for the search button. Blank ("") = standard submit button.
Submit Button Text	The text used for the button if the standard submit button is chosen, or alternative text for the submit button if an image is chosen.
Reset Button image	The image location for the reset button. Blank ("") = standard reset button.
Reset Button Text	The text used for the button if the standard reset button is chosen, or alternative text for the reset button if an image is chosen.

Attribute	Description
Template	The ID of the XML configuration the advanced search page will search. "0" to search all.

Advanced XML Search Tag Example

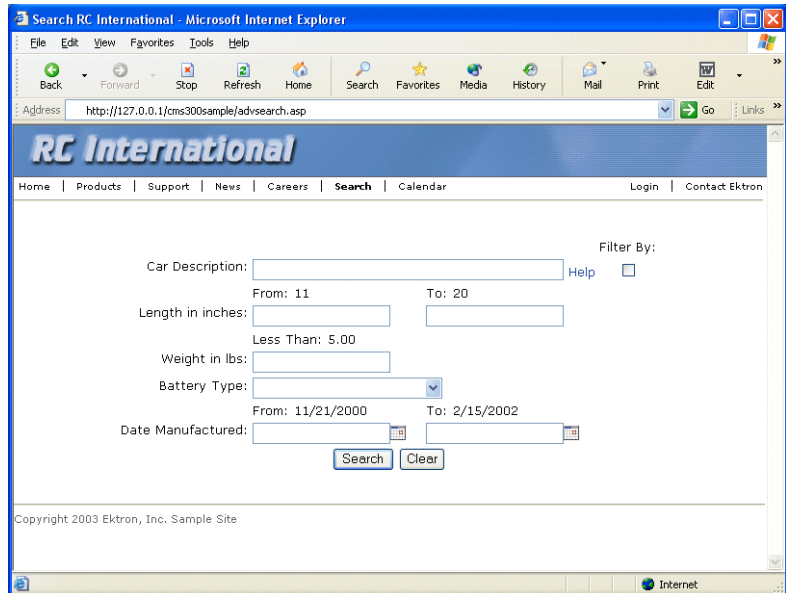
The following example places a text field and a search button, which allows you to search XML data, in a table cell inside an ASP template .

```
<tr>
  <td>
    <% ecmFTSearchAdvanced "ftsearch-display.asp",
      "", "Search", "", "Clear", 1 %>
  </td>
</tr>
```

The previous example has the following properties:

- The target page is **"ftsearch-display.asp"**.
- No image is used for the submit button, instead a standard button is used.
- The text on the submit button is "Search".
- No image is used for the reset button, instead a standard button is used.
- The text on the reset button is "Clear".
- The advanced search page displays search options set in the XML configuration ID=1.

When added to a template, and visited, the advanced XML search tag will appear similar to the following:



Search Tag Variables

Listed below are all the different variables and their descriptions.

Post Page

The target page is the page where the search results will be displayed. You may enter any form of a URL for the Web page you will have the search result displayed to.

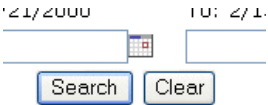
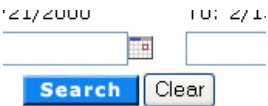
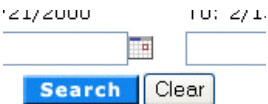
You can enter any Web page that has the **FTSearchDisplay** tag inserted in it. If you want to use the same page, you can enter “#” into the field.

NOTE

Target page **MUST** have the **FTSearchDisplay** function inserted in it for the search results to be displayed.

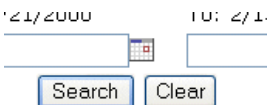
Submit Button Image Source

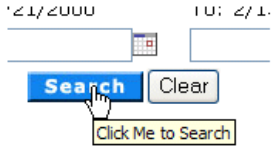
If you would like to use an image for the submit button instead of the default, you may enter the filename and path of the image.

Button Image Source	Web Page View
""	
"search.jpg"	
"http://www.ektron.com/images/search.jpg"	

Submit Button Text

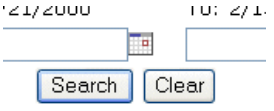
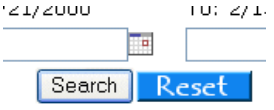
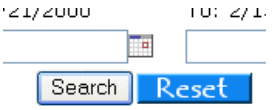
If no submit button image was specified, then the text that is entered in this field will be the text that appears on the button. If an image is being used for the submit button, then this text will be the alt text for the button.

Button Text	Web Page View
"Search"	

Button Text	Web Page View
"Click Me to Search"	

Reset Button Image Source

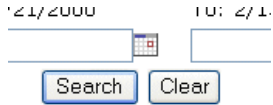
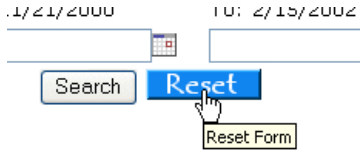
If you would like to use an image for the reset button instead of the default, you may enter the filename and path of the image.

Button Image Source	Web Page View
""	
"reset.jpg"	
"http://www.ektron.com/images/reset.jpg"	

Reset Button Text

If no reset button image was specified, then the text that is entered in this field will be the text that appears on the button. If an image is

being used for the reset button, then this text will be the alt text for the button.

Button Text	Web Page View
"Clear"	 A screenshot of a web page showing a search form. It includes two input fields with date ranges: '21/2000' and '10: 2/1'. Below the fields are two buttons: 'Search' and 'Clear'.
"Reset Form"	 A screenshot of a web page showing a search form. It includes two input fields with date ranges: '1/21/2000' and '10: 2/15/2002'. Below the fields are two buttons: 'Search' and 'Reset'. A mouse cursor is clicking the 'Reset' button, and a tooltip labeled 'Reset Form' is visible.

Template

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

Specify the ID of the XML configuration you wish to display the advanced search options for. The ID can be found by accessing the XML Configurations page in the Administrator Workarea.

Also, to display all search options for all indexed XML configurations, use "0" as your template ID.

XML Search Display

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

The XML search display custom tag creates the area where all the search results show after an XML search has been executed.

Shown here is the format of the ecmSearchDisplay tag:

```
<% ecmFTSearchDisplay      RequestForm,
                           MaxNumber,
                           "StyleInfo",
                           ShowDate           %>
```

Below is a description of each of the values of the tag:

Attribute	Description
RequestForm	Needed to pass search form back to the search display function.
Max Number	Maximum number of content blocks returned. (0=Unlimited)
Style Info	An HTML style string used within the HTML "hyperlink" generated by the search output. Can control font, color, size, etc...
ShowDate	Displays the last modified date of the content block. 0=No 1=Yes

XML Search Display Example

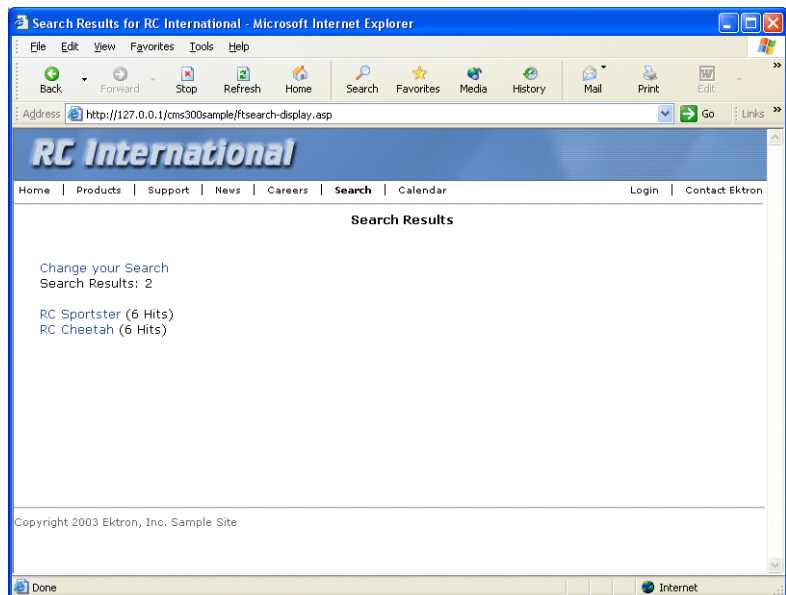
The following must be inserted on the page where you would like to display the search results. You may place it on the same page as the search tag, or insert it on a separate page.

```
<tr>
  <td>
    <% ecmFTSearchDisplay Request.Form, 25, "", True %>
  </td>
</tr>
```

The previous search display example has the following properties:

- Returns 25 results when the search is performed.
- Contains no style information.
- Shows the last modified date of the content.

When inserted into a template, and an XML search has been performed, the results will appear similar to the following.



Search Function

The search tag is used to display a text box, with a search button on the template. When seen on the template of a Web page, users will be allowed to enter text and click on the search button to search through all the content on the Web site.

Shown here is the format for the ecmSearch tag:

```
<% ecmSearch    "Starting Folder",  
                  Recursive,  
                  "Target Page",  
                  Text Box Size,  
                  Maximum  
                  Characters,  
                  "Button Image  
                  Source",  
                  "Button Text",  
                  "Font Face",  
                  "Font Color",  
                  "Font Size",  
                  Horizontal,  
                  "Spare"  
                  %>
```

The following table provides information about all the variables for the search tag.

Attribute	Description	Required?
Starting Folder	The folder you wish to search.	Yes
Recursive	1 or 0 (zero). Whether to search sub-folders. (0=No)	Yes

Attribute	Description	Required?
Target Page	The Web page to output the search results.	Yes
Text Box	The size of the display text box for user input.	Yes
Maximum Characters	The maximum characters the input text box should accept.	No
Button Image Source	The image location for the search button. Blank ("") = standard submit button.	No
Button Text	The text used for the button if the standard submit button is chosen, or alternative text for the button if an image is chosen.	No
Font Face	The font used for the text in the search tag.	No
Font Color	The font color used for the text in the search tag. Blank ("") = Black.	No
Font Size	The font size for the text in the search tag.	No
Horizontal	Determines if the search options are in a row or a column.	Yes
Spare1	Currently not used.	Must Be ""

Search Tag Example

The following example places a text field and a search button in a table cell inside an ASP template.

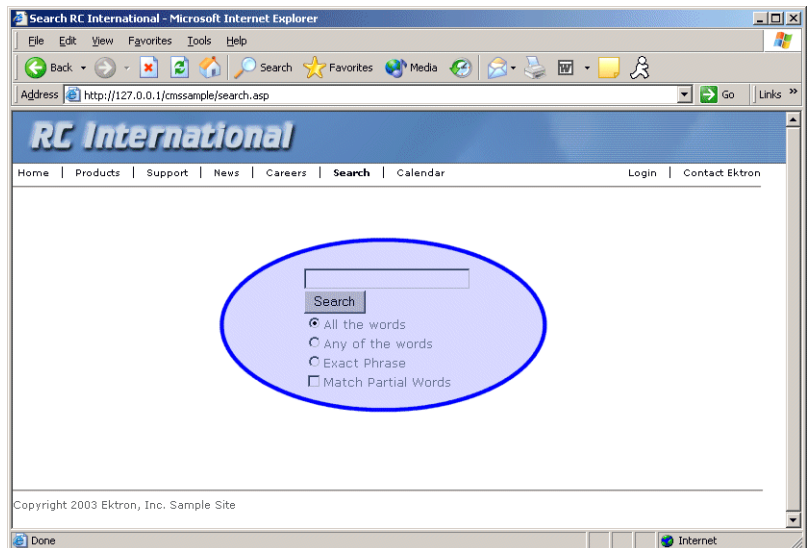
```
<tr>
  <td>
    <% ecmSearch "\marketing",1,"search-
display.asp",25,200,"","Search","Verdana",
"#808080","2",0,"" %>
  </td>
</tr>
```

The previous example has the following properties:

- Searches content in the \content\marketing folder.

- Search is recursive.
- The target page is “search-display.asp”.
- The size of the text box is 25.
- The search will allow a maximum of 200 characters.
- No image is used for a button, instead a standard submit button is used.
- The text on the submit button is “Search”.
- The search options appear in Verdana with a color of #808080 and a size of 2pt.
- The search options are arranged vertically.

When added to a template, and visited, the search tag will appear similar to the following:



Search Tag Variables

Listed below are all the different variables and their descriptions.

Starting Folder

In this variable, you will need to specify the content folder you want to begin the search at. There are two options for this variable:

Value	Description
"\"	Search will begin the search at the root folder.
"folder path"	Search will begin from the folder specified.

Recursive

When you setup the search to be recursive, you are allowing the search to search all the sub-folders associated with the starting folder you had specified. You have two options for this variable:

Value	Description
1	Recursive search (will search sub-folders of the starting folder).
0	Non-Recursive search (will only search starting folder).

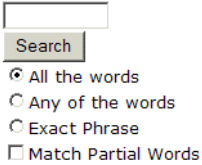
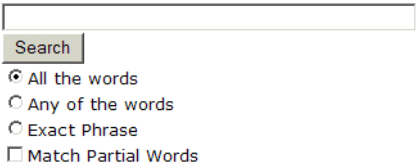
Target Page

The target page is the page where the search results will be displayed. You may enter any form of a URL for the Web page you will have the search result displayed to. You can enter any Web page that has the searchdisplay tag inserted in it. If you want to use the same page, you can enter "#" into the field.

NOTE Target page **MUST** have the searchdisplay function inserted in it for the search results to be displayed.

Text Box

This variable will set the size of the text box that is displayed. This can be any number depending on how big you would like the search text box to be.

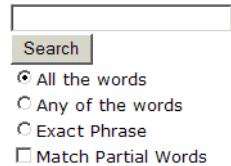
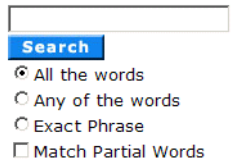
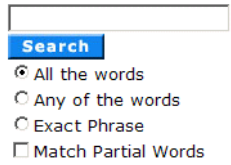
Text Box Size	Web Page View
10	
50	

Maximum Characters

The value in this attribute determines the amount of characters that can be displayed.

Button Image Source

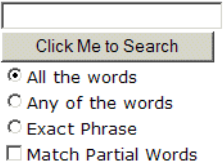
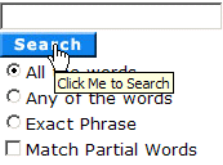
If you would like to use an image for the search button instead of the default, you may enter the filename and path of the image.

Button Image Source	Web Page View
""	
"search.jpg"	
"http://www.ektron.com/images/search.jpg"	

Button Text

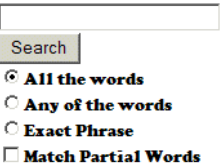
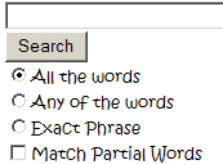
If no search button image was specified, then the text that is entered in this field will be the text that appears on the button. If an

image is being used for the search, then this text will be the alt text for the button.

Button Text	Web Page View
"Click Me to Search"	
"Click Me to Search"	

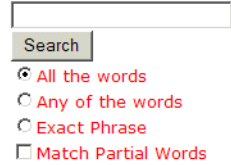
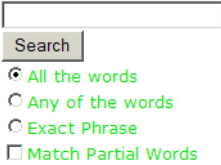
Font Face

This is the font that is used for the search options that are displayed on the Web page.

Font Face	Web Page View
Cooper Black	
Kristen ITC	

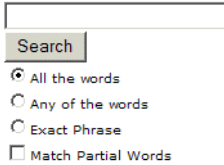
Font Color

This is the color of the text in the search tag. You may enter the color name or the hexadecimal value of the color.

Font Color	Web Page View
Red	
#14DA14	

Font Size

This is the size of the text that is in the search tag.

Font Size	Web Page View
1	

Font Size	Web Page View
5	

Horizontal

This variable determines whether the search options are displayed vertically or horizontally.

Attribute	Web Page View
0	
1	

Spare 1

At the end of the Search tag, there is a variable called Spare 1. This variable has not yet been assigned an attribute. As of right now, when you enter a search tag, you must pass this variable "".

Search Display Function

The search display custom tag created the area where all the search results show after the search has been executed.

Shown here is the format of the ecmSearchDisplay tag:

```
<% ecmSearchDisplay      "MaxNumberReturn",
                          "StyleInfo",
                          ShowDate      %>
```

Below is a description of each of the values of the tag:

Attribute	Description	Required
Max Number of content blocks Returned	Maximum number of content blocks returned. (0=Unlimited)	Yes
Style Info	An HTML style string used within the HTML "hyperlink" generated by the search output. Can control font, color, size, etc...	No
ShowDate	Displays the last modified date of the content block. 0=No 1=Yes	No

Search Display Example

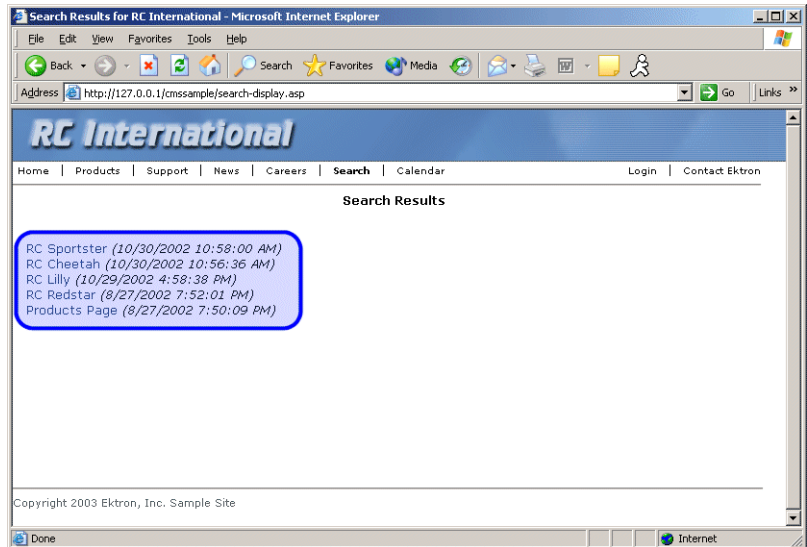
The following must be inserted on the page where you would like to display the search results. You may place it on the same page as the search tag, or insert it on a separate page.

```
<tr>
  <td>
    <% ecmSearchDisplay 0,"","1" %>
  </td>
</tr>
```

The previous search display example has the following properties:

- Returns unlimited results when the search is performed.
- Contains no style information.
- Shows the last modified date of the content.

When inserted into a template, and a search has been performed, the results will appear similar to the following.



Search Display with Default Content

You can also set up your search display tag to display a default content block on the page if a search is not performed. Then, if a search is performed, the results will take the place of the content block. The sample code shown below can do that:

```
<tr>
  <% if (Request.Form("ecmsearchtype") <> "") then %>
    <td>
      <% ecmSearchDisplay 0,"" %>
    </td>
  <% else %>
    <td>
      <% ecmContentBlock(1) %>
    </td>
  <% end if %>
</tr>
```

Search Display Variables

Listed below are all the variables for the SearchDisplay tag.

Max Number of Content Blocks Returned

This variable determines the amount of search results that will be displayed on the page. You may enter any integer into this variable. If you would like to display an unlimited amount, enter 0 (zero).

Max Returns	Web Page View
0	Search Results 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
3	Search Results RC Sportster RC Redstar RC Lilly

Style Info

An HTML style string used within the HTML “hyperlink” generated by the search output. Can control font, color, size, etc.

Style Info	Web Page View
""	<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>
"font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:50%;"	<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>

Show Date

Here, you can determine if you would like to include the Last Modified Date of the content block next to the result. This is useful

to determine which is the most recent content block in the results.
You have two choices for this:

Attribute	Web Page View
0	<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>
1	<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>

List Summary Function

The list summary function is used to display the list of content summaries on a Web page.

Shown here is the format of the list summary function:

```
<% ecmListSummary  "StartingFolder",
Recursive,
ShowSummary,
"StyleInfo",
"OrderBy, Asc/Desc",
"ShowInfo",
MaxNumber,
"Spare1",
"Spare2"
%>
```

A list and description of the variables are shown in this table:

Attribute	Description	Options
Folder	This is the folder the summary information will come from.	Any folder
Recursive	Allow the tag to get summary info from the child folders.	0 = Not Recursive 1 = Recursive
ShowSummary	Do you want the content summary to be displayed?	0 = No 1 = Yes
StyleInfo	Inline style information for the generated text hyperlink.	Any string

Attribute	Description	Options
OrderBy, ASC/DESC	Order the hyperlinks by:	"Title" "DateModified" "DateCreated" "Startdate" "AuthorFname" "AuthorLname"
	Order the hyperlinks in ascending or descending order.	"Title,asc" "Title,desc"
ShowInfo	Information that will appear to the right of the hyperlink.	"DateModified" "DateCreated" "AuthorFname" "AuthorLname"
MaxNumber	Maximum number of summaries returned.	0 = Unlimited
Spare1	Not yet implemented	MUST BE ""
Spare2	Not yet implemented	MUST BE ""

List Summary Example

Shown below is an example of the ListSummary function in an ASP Web page.

```

<TR>
  <TD>
    <% ecmListSummary "\Marketing\News",0,1,"",
      "Title,asc","DateModified",0,"","" %>
  </TD>
</TR>

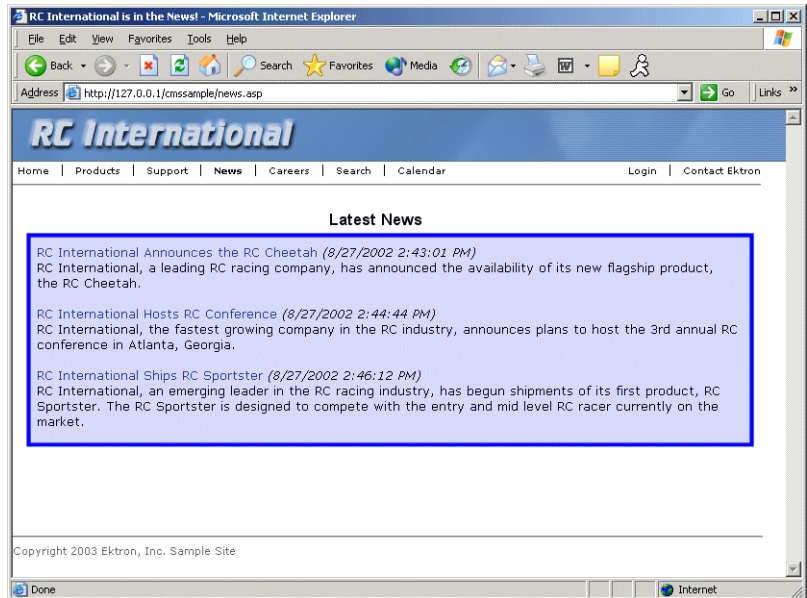
```

The previous example has the following properties:

- Displays summaries from the \Marketing\News folder.
- Not recursive.
- The summaries will be shown.
- No style was applied to the hyperlinks.

- The results will be displayed in ascending order by title.
- The date modified of each content block will be displayed.
- Unlimited results will be shown.

Here is how the summaries will appear on the web page.



List Summary Variables

Listed below are all the different variables and descriptions about them.

Starting Folder

In this variable, you will need to specify which content folder you will want to get all the summaries from. There are two options for this variable:

Value	Description
""	By leaving the value blank, all the summaries from the root content folder will be displayed.
"Folder Path"	You can specify a specific folder to display the summaries from by inserting the folder path

Recursive

When you setup the search to be recursive, you are allowing the search to search all the sub-folders associated with the starting folder you had specified. You have two options for this variable:

Value	Description
1	Recursive - summaries in the starting folder's sub-folders will also be displayed.
0	Non-recursive - only summaries in the starting folder will be displayed.

Show Summary

The show summary attribute will allow or disallow the summary to be displayed in the summary list. You have two options for this variable:

Attribute	Web Page View
0	<p>RC International Ships RC Sportster</p> <p>RC International Hosts RC Conference</p> <p>RC International Announces the RC Cheetah</p>
1	<p>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.</p> <p>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.</p> <p>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.</p>

Style Info

In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

Style Info	Web Page View
""	<p>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.</p> <p>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.</p> <p>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.</p>
"font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:100%;"	<p>RC International Announces the RC Cheetah RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p> <p>RC International Hosts RC Conference RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Ships RC Sportster 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.</p>

Order By

This variable will sort the summary results by different options. There are five different ways to sort the summaries:

Value	Summaries sorted by
Title	title
DateModified	date modified
Date Created	date created
Startdate	The Start Date assigned to the content block when created.

Value	Summaries sorted by
AuthorFname	author's first names
AuthorLname	author's last name

Ascending/Descending

This variable is an addition to the previous OrderBy variable. By default, the list of the summaries are descending, meaning they will follow the path of A-B-C, or 1-2-3. When set to ascending, they will order by Z-Y-X or 3-2-1. You will have the option to re arrange these to display either in ascending or descending order.

To do this, add the following to the OrderBy attribute:

Value	Description
asc, 1	This will order the Summaries in an ascending order
desc, 0	This will order the Summaries in a descending order. This is the default setting.

By using the ascending/descending option, you will have the option to set the order of content blocks from most recent to oldest, or vice versa. As well as ordering the list by the user's first names or last names in ascending or descending order.

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are four different options that you may choose from:

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.	d level RC racer currently on the mark arence (4/25/2002 4:42:55 PM) owing company in the RC st the 3rd annual RC conference in A
AuthorFname	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
AuthorLname	Displays the last name of the user who last edited the content block.	ct, RC Sportster. The RC Sp d mid level RC racer currentl Conference (Doe) st growing company in the R to host the 3rd annual RC cc

Max Number

In this variable, you will determine how many summaries will be displayed on the page. You may enter any number into this field,

and that number will be the amount of summaries that will be displayed. If you want all the summaries to be displayed, enter 0 (zero) and you will display unlimited results.

Max Number	Web Page View
0	<p>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.</p> <p>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.</p> <p>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.</p>
1	<p>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.</p>

Spare 1 & 2

There are two variable fields at the end that are currently not used. They are there for future scalability. When you insert a ListSummary tag into your template, you must leave those variables as ""

Single Summary Function

The single summary tag is used to display the summary of one content block in the Ektron CMS200/300 Web site, instead of a list of all the summaries in a content folder. Seen below is the single summary tag:

```
<% ecmSingleSummary "ContentID",
                    ShowSummary,
                    "StyleInfo",
                    "ShowInfo",
                    "Spare1" %>
```

You can look at the single summary tag as just a simpler form of the List summary tag. The format for each of them is almost identical, but the single summary tag does not require as many variables to be passed.

A list and description of the variables are shown in this table:

Attribute	Description	Required?
Content ID	This is the ID number of the content block the summary information will come from.	Yes
ShowSummary	Do you want the content summary to be displayed?	Yes
StyleInfo	Inline style information for the generated text hyperlink.	No
ShowInfo	Information that will appear to the right of the hyperlink.	No
Spare1	Not yet implemented	MUST BE ""

Single Summary Example

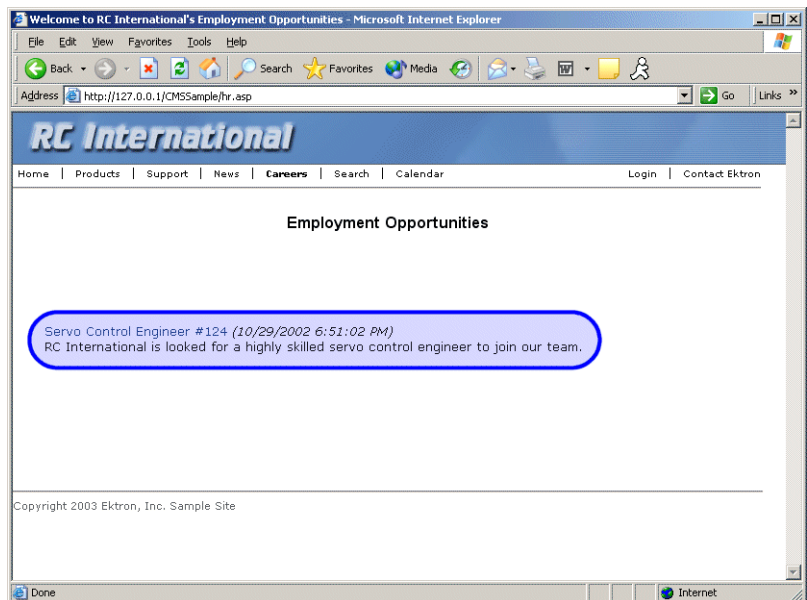
Shown below is an example of the single summary function in an ASP Web page.

```
<TR>
  <TD>
    <% ecmSingleSummary "14",1,"","DateModified","" %>
  </TD>
</TR>
```

The previous example had the following properties:

- Displays the summary for the content block with an ID=14
- The summary will be shown
- No style was applied to the hyperlinks
- The last date the content was modified will be displayed

Here is how the summary will appear on the web page.



Single Summary Variables

Listed below are all the different variables and descriptions about them.

Content ID

In this variable, you will need to specify which content block you will want the summary to be displayed from.

Show Summary

The show summary attribute will allow or disallow the summary to be displayed in the summary list. You have two options for this variable:

Value	Web Page View
0	Servo Control Engineer #124
1	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.

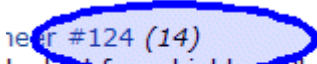
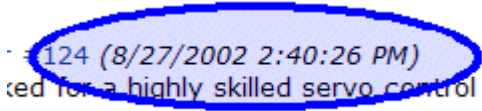
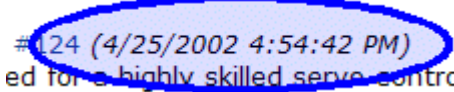
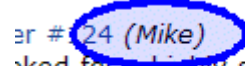
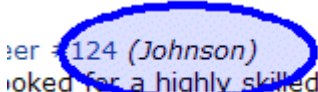
Style Info

In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

Style Info	Web Page View
""	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.
font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:100%;	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are five different options that you may choose from:

Value	Displays the...	Web Page View
ID	ID number for the content block.	
DateModified	date the content was last modified.	
DateCreated	date when the content was first created.	
AuthorFname	first name of the user who last edited the content block.	
AuthorLname	last name of the user who last edited the content block	

Spare 1

There is one variable fields at the end that are currently not used. This is here for future scalability. When you insert a single summary tag into your template, you must leave this variables as ""

Collection Function

The collection function is used to display the list of links that were created as a collection in the Ektron CMS200/300 Workarea. The ecmCollection function is highly customizable, allowing you to easily define how the collection data will be displayed on the Web page.

Shown here is the format of the ecmCollection function.

```
<% ecmCollection      "id",  
                      "Display Function"  %>
```

Each of the attributes in the ecmCollection function are explained in the following table.

Attribute	Description
id	The id of the collection that you would like to be displayed is defined here.
displayFunction	Specify a function that is defined in the API that will define how the collection data is displayed on the Web page. The display function must also be defined in the ecmCollection function.

Collection Examples

The following are examples of the collection function in the Ektron CMS200/300 sample Web site. One example uses the "ecmNavigation" display function and the other uses the "ecmTeaser" display function.

The ecmNavigation and ecmTeaser display functions are sample display functions included with Ektron CMS200/300. For information about creating your own custom display functions, see ["Creating Custom Display Functions" on page 105](#).

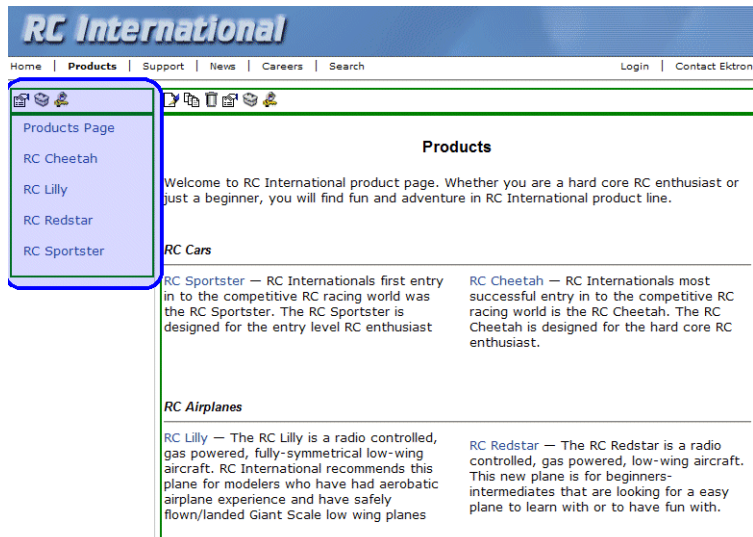
Example Using the ecmNavigation Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS200/300 sample Web site.

```
<% ecmCollection 1, "ecmNavigation" %>
```

In this example, the collection with an ID=1 is displayed, and the function “ecmNavigation” is used to define how the collection data is displayed on the Web page.

Shown here is how the collection would appear on the Web page.



As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used.

Shown below is the source code for the display function "ecmNavigation"

```
Function ecmNavigation(cInfo)
    dim html, info
    html = "<table border=""0"" cellspacing=""0"" cellpadding=""0""
           width=""100%"">"
    for each info in cInfo
        html = html & "<tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<a href=""
        html = html & info("ContentLinks")
        html = html & "">"
        html = html & Cstr(info("ContentTitle"))
        html = html & "</a></td></tr><tr><td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td></tr>"
    next
    html = html & "</table>"
    ecmNavigation = html
end Function
```

As you can see, the ecmNavigation function is a simple ASP function that creates a table of the links that belong to the collection specified.

The ecmNavigation function also displays the title of the content blocks by using the "info("ContentTitle")) building block. See ["Creating Custom Display Functions" on page 105](#) for more information.

Example Using the ecmTeaser Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS200/300 sample Web site.

```
<% ecmCollection 2, "ecmTeaser" %>
```

In this example, the collection with an ID=2 is displayed, and the function "ecmTeaser" is used to define how the collection data is displayed on the Web page.

Shown here is how the collection would appear on the Web page.



As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used.

Shown below is the source code for the display function "ecmTeaser"

```
Function ecmTeaser(cInfo)
    dim html, info
    html = "<table border=""0"" cellspacing=""0"" cellpadding=""0""
        width=""100%"">"
    for each info in cInfo
        html = html & "<tr><td><a href=""
        html = html & CStr(info("ContentLinks"))
        html = html & "" "" >"
        html = html & CStr(info("ContentTitle"))
        html = html & "</a>&nbsp;"
        html = html & info("DateModified")
        html = html & "</td></tr><tr><td>"
        html = html & CStr(info("ContentTeaser"))
        html = html & "</td></tr><tr><td>&nbsp;</td></tr>"
    next
    html = html & "</table>"
    ecmTeaser = html
end Function
```

As you can see, the ecmTeaser function is a simple ASP function that creates a table of the links that belong to the collection specified.

The ecmTeaser function also displays the:

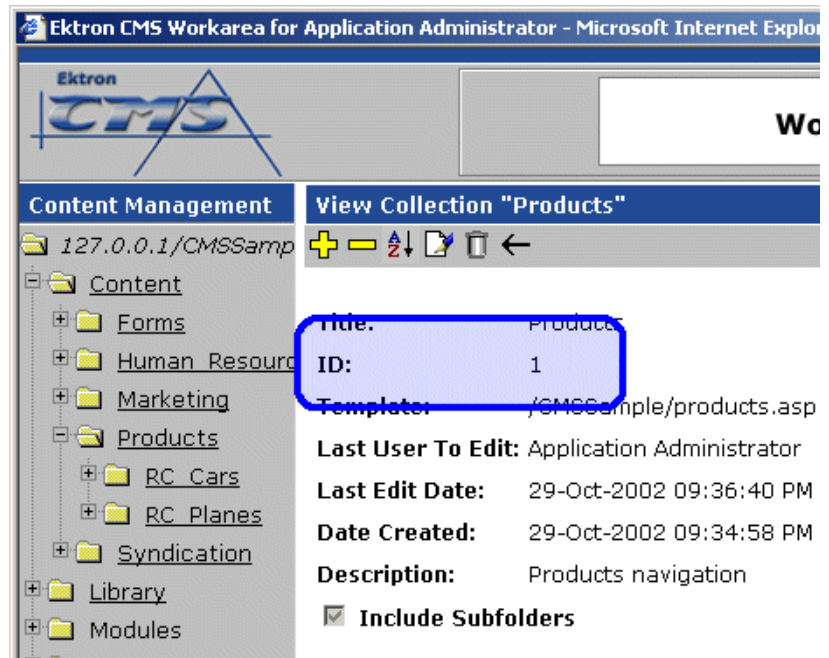
- Content Title
- Date Modified
- Content Teaser

See ["Creating Custom Display Functions" on page 105](#) for more information.

Collection Variables

ID

The collection ID is the ID number that is assigned to the collection when it is created in the workarea.



DisplayFunction

As explained earlier, the display function is a function that you create that will define how the collection data will be displayed on the Web page.

By using simple ASP scripting, you can create your own display functions, or use the three included functions.

The display function must be defined in two places.

- In the ecmCollection function as a case
- In the API file as a function.

Creating Custom Display Functions

The following section explains the basics behind creating custom display functions for your ecmCollection function.

Function Structure

As shown in the examples above, the display functions use a similar structure:

```
Function yourCollectionDisplayFunction(cInfo)
    dim html, info
    html = html & "<b>List of Content Titles:</b><br>"
    for each info in cInfo
        html = html & info("ContentTitle") & "<br>"
    next
    yourCollectionDisplayFunction = html
end Function
```

Creating a Basic Display Function

Explained below is a how to create a basic display function.




1. In the API file, begin by defining the display function you are creating	<pre>Function MyDisplayFunction(cInfo) end Function</pre>
2. Declare any necessary variables	<pre>Function MyDisplayFunction(cInfo) dim html, info end Function</pre>





<p>3. Add any html or text that you would like to apply to the collection data as a whole, not individually (for example, add a opening and closing html table tag)</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" html = html & "" end Function</pre>
<p>4. Add a For loop that will be used to cycle through each of the collection items, and apply the information defined.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo next html = html & "" end Function</pre>
<p>5. Inside the for loop, create the string that will be applied to each of the items generated by the collection data.</p> <p>For information about the building blocks, see "Building Blocks" on page 108.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo next html = html & "" end Function</pre>
<p>This will place a at the beginning and the end of each of the items displayed.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" next html = html & "" end Function</pre>




<p>This will create the hyperlink for the collection item. When the collection item is generated on the Web page, the hyperlink will send the visitor to the corresponding content block.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" & Cstr(info("ContentTitle")) & "" html = html & "" next html = html & "" end Function</pre>
<p>This will display the summary for the content block that is displayed with the collection item.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" & Cstr(info("ContentTitle")) & "" html = html & Cstr(info("ContentTeaser")) html = html & "" next html = html & "" end Function</pre>
<p>6. Finally, add this line at the end of the function to return the HTML to be outputted to the ecmCollection function.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" & Cstr(info("ContentTitle")) & "" html = html & Cstr(info("ContentTeaser")) html = html & "" next html = html & "" MyDisplayFunction = html end Function</pre>


Building Blocks

The sample display function that we just created only uses two building blocks. This section explains each of the possible building blocks that you may use when creating a custom display function.

Name	Description	Web Page View
<code>info("ContentTeaser")</code>	Displays the summary that has been created for the content block that is displayed in the collection data.	 <ul style="list-style-type: none"> • Plastic Molder #123 RC International is looking for an experienced plastics molder. • Servo Control Engineer #124 RC International is looking for a highly skilled servo control engineer to join our team.
<code>info("Comment")</code>	Displays the comment added to the content block.	 <ul style="list-style-type: none"> • Plastic Molder #123 This is the historical comment for Plastics Molder • Servo Control Engineer #124 This is the historical comment for Servo Control
<code>info("ContentID")</code>	Displays the ID number of the content block that is assigned to it by Ektron CMS200/300.	 <ul style="list-style-type: none"> • Plastic Molder #123 ID= 13 • Servo Control Engineer #124 ID= 14

Name	Description	Web Page View
<code>info("ContentLinks")</code>	Inserts the path of the content block with respect to your site root.	 <ul style="list-style-type: none"> • Plastic Molder #123 /CMS200Sample/hr.asp?id=13 • Servo Control Engineer #124 /CMS200Sample/hr.asp?id=14
<code>info("ContentTitle")</code>	Displays the title of the content block that is displayed with the collection data.	 <ul style="list-style-type: none"> • Plastic Molder #123 /CMS200Sample/hr.asp?id=13 • Servo Control Engineer #124 /CMS200Sample/hr.asp?id=14
<code>info("DateModified")</code>	Displays the date and time the content block was last modified.	 <ul style="list-style-type: none"> • Plastic Molder #123 8/27/2002 2:38:34 PM • Servo Control Engineer #124 8/27/2002 2:40:26 PM
<code>info("DisplayDateModified")</code>	Displays the date and time the content block was last modified using the date/time format specified in Setup.asp.	 <ul style="list-style-type: none"> • Plastic Molder #123 27-Aug-2002 02:38:34 PM • Servo Control Engineer #124 27-Aug-2002 02:40:26 PM

Name	Description	Web Page View
<code>info("DisplayEndDate")</code>	Displays the end date and time for the content block using the date/time format specified in Setup.asp.	
<code>info("DisplayGoLive")</code>	Displays the start date and time for the content block using the date/time format specified in Setup.asp.	 <ul style="list-style-type: none"> • Plastic Molder #123 This content went live on: 30-Oct-2002 10:5 • Servo Control Engineer #124 This content went live on: 29-Oct-2002 06:5
<code>info("EditorFname")</code>	Displays the first name of the user who last edited the content block.	 <ul style="list-style-type: none"> • Plastic Molder #123 Application • Servo Control Engineer #124 Application
<code>info("EditorLname")</code>	Displays the last name of the editor who last edited the content block.	 <ul style="list-style-type: none"> • Plastic Molder #123 Administrator • Servo Control Engineer #124 Administrator
<code>info("EndDate")</code>	Displays the end date and time for the content block	

Name	Description	Web Page View
info("GoLive")	Displays the start date and time for the content block.	 <p>The screenshot shows a web page with a green header bar containing icons for a folder, a document, a box, and a person. Below the header, there is a list of items:</p> <ul style="list-style-type: none"> • Plastic Molder #123 This content went live on: 10/30/2002 10:00 • Servo Control Engineer #124 This content went live on: 10/29/2002 6:50

Defining Display Functions

Once a display function has been created in the API, you need to define it in the ecmCollection function as a new case.

To define a display function as a new case:

1. Locate the ecmCollection function in the API file.
2. Find the line that contains "Select Case displayFunction"
3. Under that line, type in the following substituting "MyDisplayFunction" with the actual name of the function that you created:

```
Case "MyDisplayFunction"
    html = MyDisplayFunction(cInfo)
```

4. Save and close the file

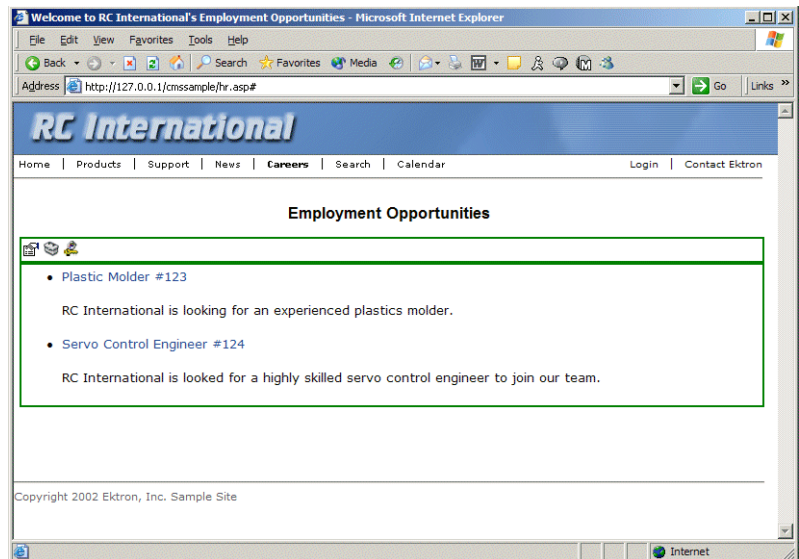
With the display function defined, and added to the ecmCollection function as a new case, you are ready to use the display function to define how collections look.

Using a Custom Display Function

To use the custom display function that you have created, simply add the ecmCollection function to a template, specify the ID of the collection that will be displayed, and add the display function that will define how the collection is displayed.

```
<% ecmCollection "2", "MyDisplayFunction" %>
```

When displayed on the Web site, the collection will look like the following.



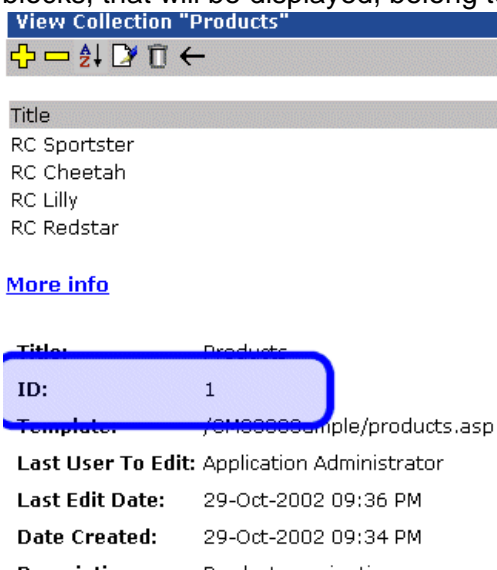
Random Content

The random content function is used to randomly display the content of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

<% ShowRandomContent "id", XSLTid %>

Where the “id” represents the ID number of the collection the content blocks, that will be displayed, belong to.



And “XSLTid” is the ID number of the display XSLT that will be applied to the content blocks in the collections.

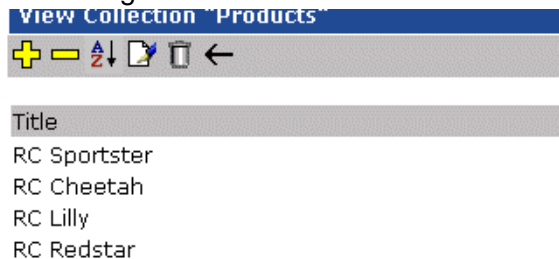
Random Summary

Similar to the random content function, the random summary function randomly displays a content block summary that belongs in a specified collection.

Here is the format for the random content function

```
<% ShowRandomSummary "id" %>
```

The "id" represents ID number of the collection to which the content blocks belong.



[More info](#)

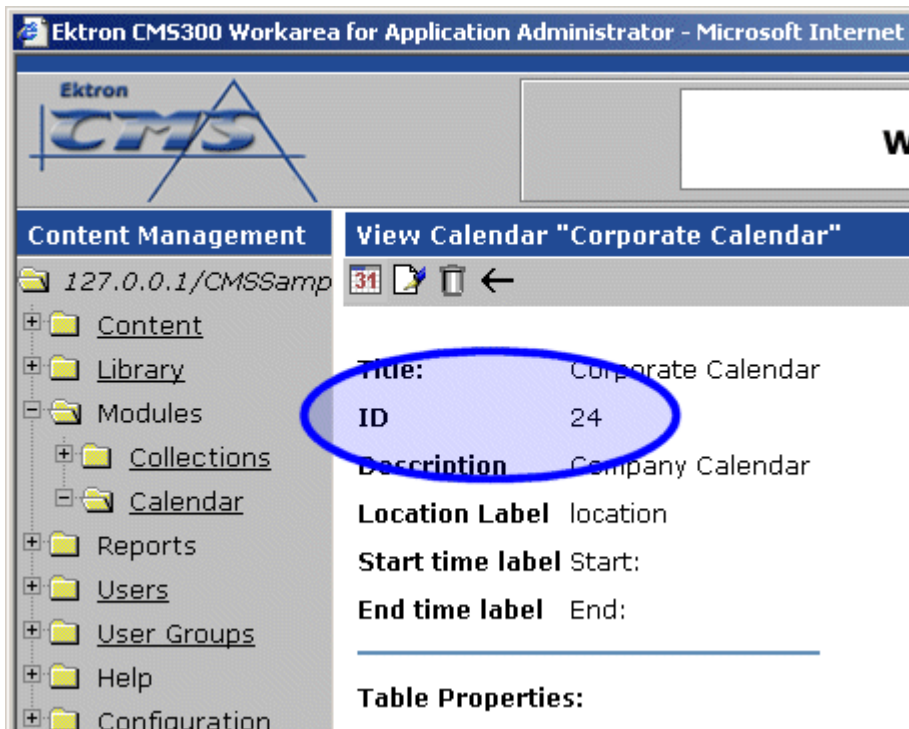
Title:	Products
ID:	1
Template:	/CMS00000/sample/products.asp
Last User To Edit:	Application Administrator
Last Edit Date:	29-Oct-2002 09:36 PM
Date Created:	29-Oct-2002 09:34 PM
...	...

Event Calendar Function

The ecmEvtCalendar function displays an event calendar that has been created in Ektron CMS200/300. Here is the format for the event calendar custom function.

```
<% ecmEvtCalendar "id" %>
```

The “**id**” argument represents the ID number of the event calendar in the Workarea.



Event Calendar Style Sheet

Before you insert an event Calendar into your Ektron CMS200/300 template, you should make a reference to the style sheet,

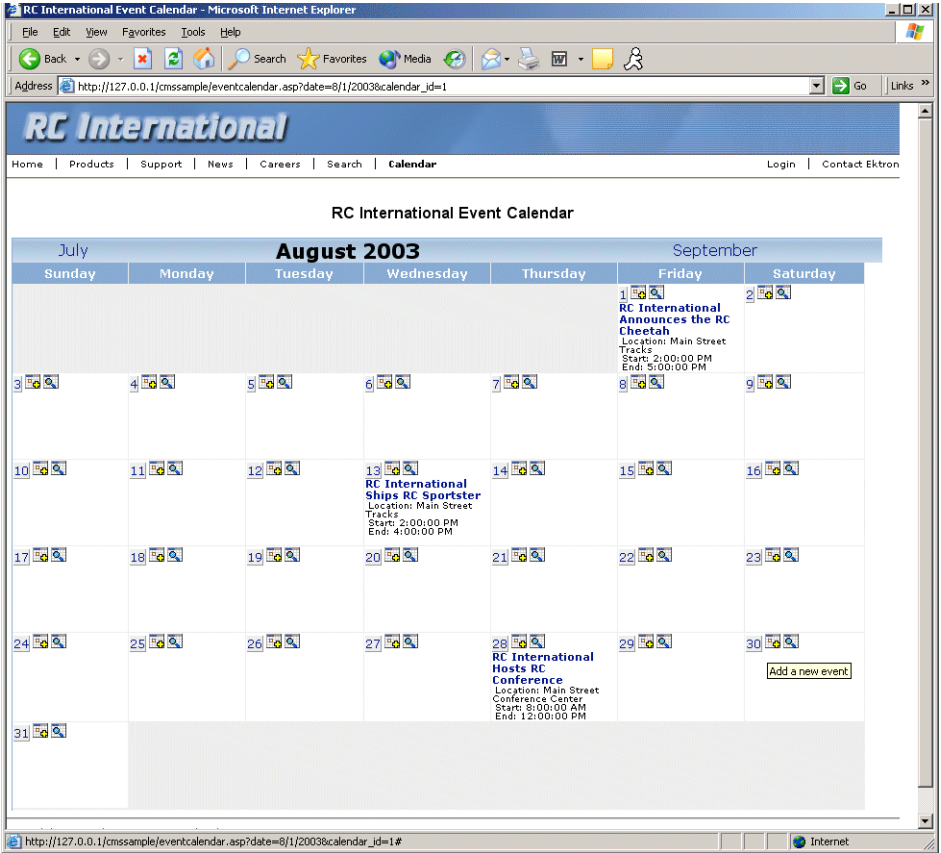
\CMS300Sample\ekCalendar.css, in your template. This style sheet contains styles that define the Calendar.

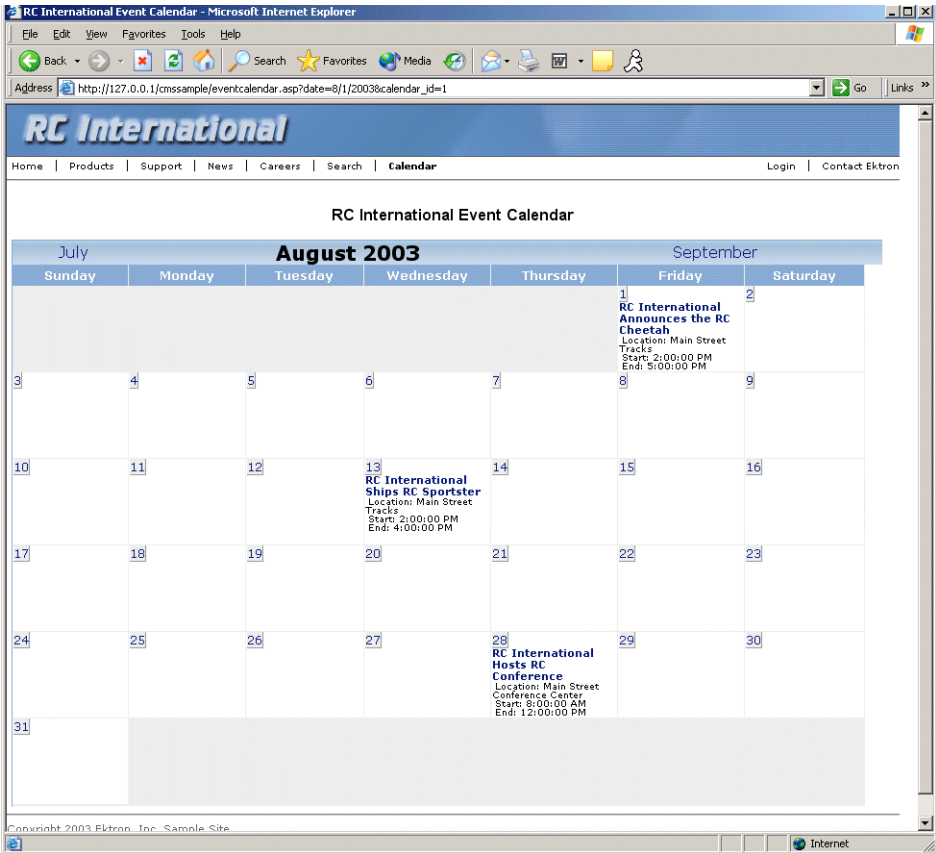
Event Calendar Sample

Here is an example of an event Calendar in an Ektron CMS200/300 template.

```
<tr>
  <td>
    <% ecmEvtCalendar "1" %>
  </td>
</tr>
```

When this code is displayed in a browser, the Calendar with an ID-1 is displayed. Here is the calendar displayed on a Web Page

If...	Calendar Display
Logged in	 <p>The screenshot shows a web browser window titled "RC International Event Calendar - Microsoft Internet Explorer". The address bar displays the URL: <code>http://127.0.0.1/cmssample/eventcalendar.asp?date=8/1/2003&calendar_id=1</code>. The page content features the "RC International" logo and a navigation menu with links: Home, Products, Support, News, Careers, Search, and Calendar. The main content area is titled "RC International Event Calendar" and displays a monthly calendar for August 2003. The calendar includes events for August 1st, 13th, and 28th, each with details like location, start/end times, and a "Add a new event" button is visible on August 30th.</p>

If...	Calendar Display
Logged Out	

As you can see, when you are logged into the Web site, you can add and view Calendar events. When logged out, you can only view calendar events.

Abbreviating Day Names

To further customize the event calendar function, you may specify in ApplicationAPI.asp abbreviated day names (for example, Mon instead of Monday, Wed instead of Wednesday).

Here is an example of the ecmEvtCalendar using abbreviated day names.

November		December 2003			January
Mon	Tue	Wed	Thu	Fri	
1	2	3	4	5	
8	9	10	11	12	

Configuring ApplicationAPI.asp

To configure the ecmEvtCalendar function to display abbreviated day names, perform the following steps.

NOTE

Before editing ApplicationAPI.asp, Ektron recommends copying it, referencing it in site_scripts_path.asp, and making changes to the new file. See "Custom API" on page 132 for additional information.

1. Open the file that contains the ecmEvtCalendar function you want to edit.
2. Locate the ecmEvtCalendar function in the file.

```

573 Function ecmEvtCalendar(byVal cID)
574     'check to see if we are logged in if we need some script
575     if request.cookies("ecm")("user_id") <> "" then
576         if (PopUpLoaded = 0) then
577             Response.Write(ecmGetPopupScript())
578             PopUpLoaded = 1
579         end if
580     end if
581     dim CalendarObj,Arg,CalView
582     Set CalendarObj = Server.CreateObject(MODULE_OBJ)
583     Set Arg = Server.CreateObject("Scripting.Dictionary")
584     if((request.QueryString("ekM") <> "") And (request.QueryString("ekY") <> ""))
585         Arg.Add "Month" , request.QueryString("ekM")
586         Arg.Add "Year", request.QueryString("ekY")
587     else
588         Arg.Add "Month" , ""
589         Arg.Add "Year", ""
590     end if
591     Arg.Add "ImagePath",AppImgPath
592     Arg.Add "AppPath",AppPath
593     Arg.Add "CallerID",request.cookies("ecm")("user_id")
594     Arg.Add "Site",request.cookies("ecm")("site_id")
595     Arg.Add "CalendarID",cID
596     Arg.Add "WorkSpace",""
597     Arg.Add "QueryString",Request.ServerVariables("Query_String")
598     Arg.Add "URLfile",Request.ServerVariables("URL")
599     Arg.Add "PathInfo",Request.ServerVariables("path_info")

```

3. Locate the line that reads
Arg.Add "AbbreviateDayNames", true"
4. On that line, change **false** to **true**. This causes the ecmEvtCalendar to use abbreviated day names.
5. Save and close the file.
6. View the Web page that contains the updated function.

Form Function

The ecmFormBlock function displays an HTML form content block. For more information about HTML forms, see the Ektron CMS200/300 Administrator manual chapter “Working with HTML Forms.”

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 CMS200/300 Administrator manual chapter “Managing Library Assets”, section “Updating Default Template for Multiple Quicklinks.”

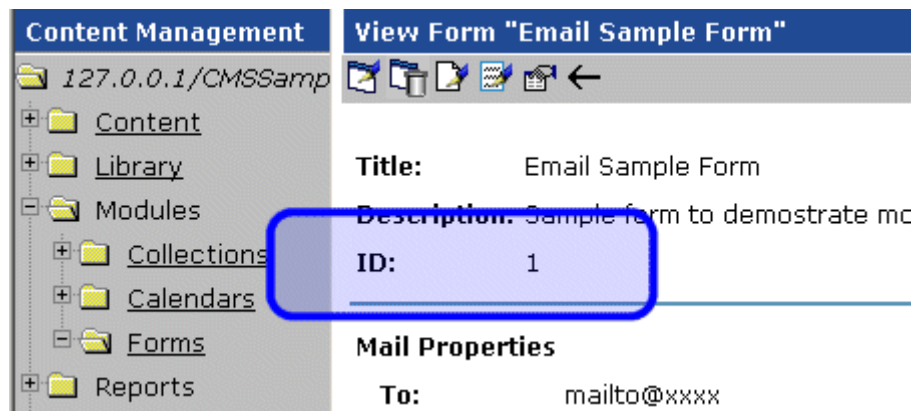
Here is the format for the form custom function.

```
<% ecmFormBlock "id" %>
```

NOTE

As an alternative, you can use the multipurpose function, which can display either a content block or a content block associated with a form. For more information, see “MultiPurpose Function” on page 123.

The “id” argument represents the form’s ID number, visible when viewing the form from the View Form screen.



Here is an example of a form content block in an Ektron CMS200/300 template.

```
<tr>
  <td>
```

```
<% ecmFormBlock "1" %>
</td>
</tr>
```

When this code is read by a browser, the following is displayed.



The screenshot displays the RC International website's contact form. The header features the "RC International" logo in a blue banner. Below the logo is a navigation menu with links: Home, Products, Support, News, Careers, Search, and Calendar. The contact form itself includes input fields for "First Name:", "Last Name:", and "Email:". There is a dropdown menu for "Ektron Product:" with "CMS300" selected. A large text area for "Comments:" is positioned below the product dropdown. At the bottom of the form is a "Send Mail" button.

When a visitor to your site submits a form, the form uses the form information to process the data.

MultiPurpose Function

The ecmMultiPurpose function displays either a content block or an content block associated with a form. As a result, you can use one template to display both types of content block.

Here is the format for the Multipurpose function:

`<% ecmMultiPurpose "id" %>`

To pass in a standard content block, pass the id value in the url parameter, as illustrated below.

```
http://localhost/cms300sample/index.asp?id=2
```

To pass in a Form content block, pass the ekfrm value in the url parameter, as illustrated below.

```
http://localhost/cms300sample/index.asp?ekfrm=2
```

NOTE

For more information about passing the id value as a url parameter, see ["Dynamic Content Block" on page 52.](#)

If the id value or ekfrm value does not exist, a blank template appears.

If the url parameter is not passed in or recognized, the function calls ecmContentBlock and passes the DefaultID value as the id argument. If the DefaultID value is zero (0), ecmContentBlock is not called.

Metadata Function

The metadata tag is used to display the metadata for one or more content blocks in the Ektron CMS200/300 Web site. This custom tag lets you generate the list of metadata edited by the authors and insert it in the source of the Web page. Below is an example of the metadata custom tag.

```
<% ecmMetaData "ItemList","Spare1" %>
```

A list and description of the variables are shown in this table:

Attribute	Description	Required?
ItemList	List of the content block ID numbers, and exclusions	No
Spare1	Not yet implemented	MUST BE ""

Creating an Item List

The ItemList for the ecmMetadata tag has the following format:

```
"[content block id; Exclusion list]...[content block id; Exclusion list]"
```

Attribute	Description	Required?
[(left bracket)	Delineates the beginning of a content block id/Exclusion List pair.	Yes
Content Block ID	A number that represents the content block you wish to retrieve the metadata from	Yes
; (semicolon)	Separator: Separates the content block id from the exclusion list	No (Yes if including an exclusion list)
Exclusion List	A comma delimited list of metadata you do not want to be retrieved for this content block.	No

Attribute	Description	Required?
] (right bracket)	Delineates the end of a content block id/Exclusion List pair.	Yes

ecmMetadata Example

For this example, our administrator has defined the following metadata types for Ektron CMS200/300.

Name	Style	Type	Editable	Required	Remove Dups	Case Sensitive
Title	n/a	HTML	Yes	Yes	No	n/a
Keywords	Name	Meta	Yes	Yes	Yes	No
Content-Type	http-equiv	Meta	No	Yes	Yes	n/a

Our example Web page has two content blocks displayed. Content block 1, the company's home page introduction text, and content block 2, a sidebar containing the lead story for the week. The metadata entered by your content editors for these two content blocks are:

Content Block 1:

Title – "Welcome to Our Home Page"

Keywords – "Home; page; company; Welcome"

Content-Type -- "text/html; charset=iso-8859-1"

Content Block 2:

Title – "Announcing our New Product!"

Keywords – "Announce; company; product; widget"

Content-Type -- "text/html; charset=iso-8859-1"

To include both content blocks metadata in a Web page, the ecmMetadata call looks like this.

```
<html>
  <head>
    <% ecmMetadata "[1][2;Title,Content-Type]", " " %>
    (rest of head)
  </head>
  <body>
```

```
        (rest of web page document)
    </body>
</html>
```

In the above example, the ecmMetadata call is contained in the <head> of the HTML document (since all HTML metadata is required to exist there). When the Web page is viewed, the Web page source contains the following information:

```
<html>
  <head>
    <Title>Welcome to Our Home Page</Title>
    <meta name="Keywords" content= Home; page; company; Welcome; Announce;
    product; widget">
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    (rest of head)
  </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

The ecmMetadata constructed these three HTML tags based on the administrator defined parameters and the ecmMetadata call.

The metadata tag was constructed in the page because:

- **[1]** - Displays all metadata content for the content block ID=1. In this example, the title, keywords, and content-type were defined, so they were displayed as:

```
<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome;
Announce; product; widget">
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
```

- **[2;title,content-type]** - Displays all metadata content for content block ID=2, *except* for its title and content-type definitions.

```
<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome;
Announce; product; widget">
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
```

The next section provides more examples.

More ecmMetadata Examples

Shown below is an example of the metadata function in an ASP Web page.

```
<TR>
  <TD>
    <% ecmMetaData "[1][2;title][14;keywords]", "" %>
  </TD>
</TR>
```

The previous example has the following properties:

- All metadata for content block ID=1 is inserted into the page
- All metadata for the content block ID=2 except for its title is inserted into the page
- All metadata for the content block ID=14 except for its keywords is inserted into the page

Using a Dynamic Metadata Tag

You can use a metadata tag in a template to define the metadata for a specific content block. However, you can also use a dynamic metadata tag combined with a content block tag to show the metadata for all content blocks that are displayed on the Web page. This allows you to create a template and not have to keep updating the tag by adding the new ID and exclusion list. Shown below is the sample code for this.

```
<!-- #include virtual="/CMS200/300sample/CMS200/300scripts/
ektronAPI.asp" -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <% Dim id
      if (request.QueryString("id") <> "") then
        id = request.QueryString("id")
      else
        id = 1
      end if
      ecmMetadata "[" & id & "]", ""
    %>
  </head>
  <body>
    <% ecmContentBlock(request.QueryString("id")) %>
  </body>
</html>
```

In this example, if you specify a content block in the URL and the page is displayed, the metadata takes that ID and displays the proper metadata in the source for the content block.

Listed below in the next section is a description of each variable.

ecmMetadata Variables

Below are the variables and their descriptions.

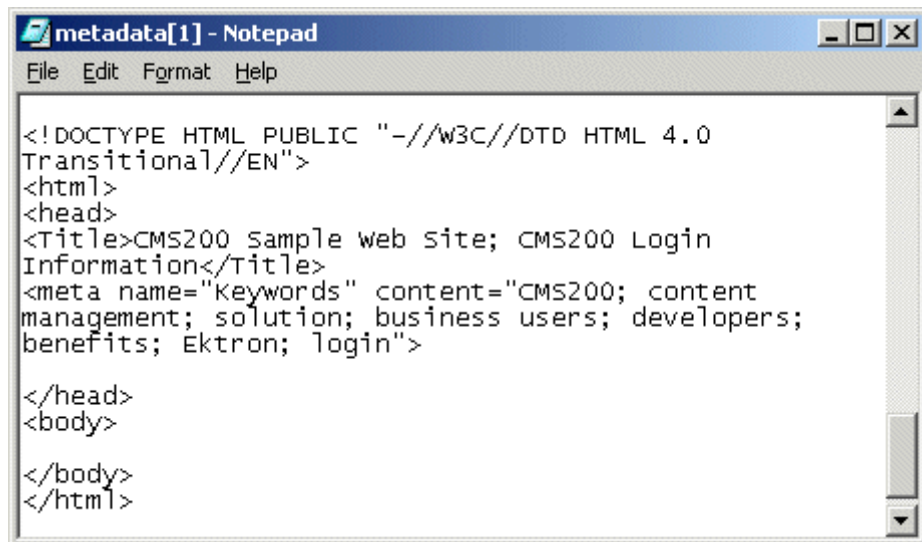
Item List

In this variable of the ecmMetadata tag, you need to specify two values:

Value	Description
ID	Specify the content block ID to specify which content block will have the metadata displayed.
Exclusion List	After you specify the ID for the content block you want to display the metadata, you can also specify the metadata to be excluded.

Some examples might be:

```
<% ecmMetaData "[1][2]", "" %>
```



```

metadata[1] - Notepad
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 sample web site; CMS200 Login
Information</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits; Ektron; login">

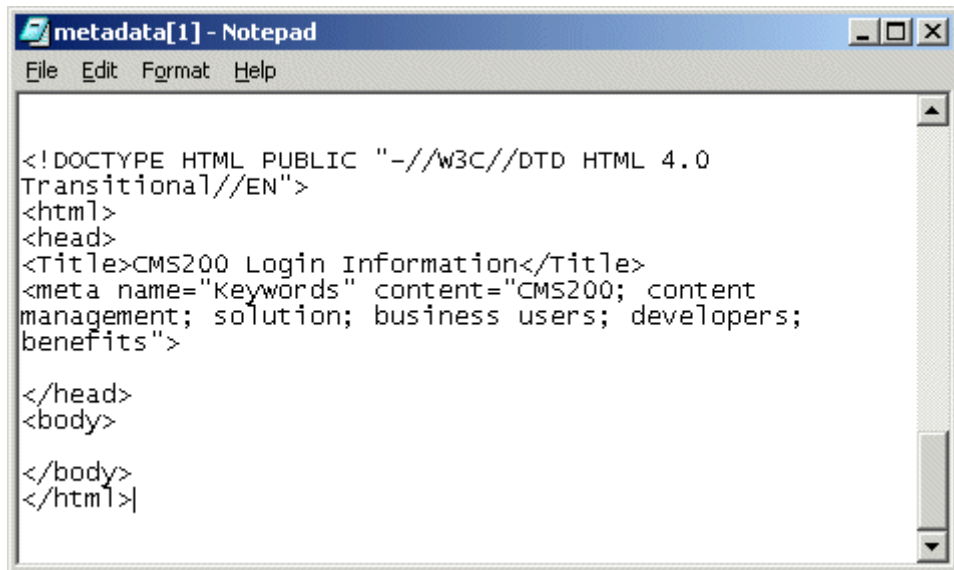
</head>
<body>

</body>
</html>

```


In this example, all the metadata from the content block ID=1 and 2 is displayed.

```
<% ecmMetaData "[1;title][2;keywords]", "" %>
```



```

metadata[1] - Notepad
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Login Information</Title>
<meta name="keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

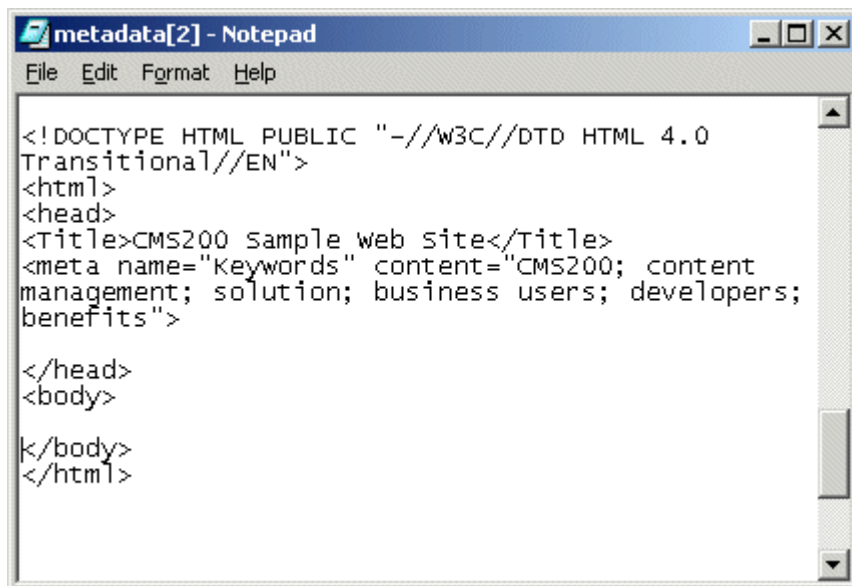
</head>
<body>

</body>
</html>

```

In this example, all the metadata from content block ID=1 will be displayed *except* for the title. Also, all the metadata for content block ID=2 is displayed *except* its keywords.

```
<% ecmMetaData "[1][2;title,keywords]", "" %>
```



```

metadata[2] - Notepad
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Sample web site</Title>
<meta name="keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

</head>
<body>

</body>
</html>

```

In this example, all metadata from content block ID=1 is displayed, and all metadata from content block ID=2 is displayed except its title and keywords.

Spare 1

There is one variable field at the end that is currently not used. It is for future scalability. When you insert a single metadata tag into a template, you must leave this variables as "".

Show Site Language Function

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

Site Language: English (US)

- Danish
- English (US)
- French (Standard)
- German (Standard)

(For more information, see the “Multi-Language Support “ chapter of the **Ektron CMS200/300** Administrator manual.)

Below is an example of the Show Site Language custom tag.

```
<% ecmShowSiteLanguage(FieldName)%>
```

You can place this function in any location of any page on your site. By default, it appears in the lower right corner of the home page.

The variable is described in this table.

Variable	Description
FieldName	<p>An optional argument. You can leave it blank and allow the drop-down list to be populated by the available languages for your Web site. For example:</p> <pre><%=ecmShowSiteLanguage('')%></pre> <p>Or you can pass in an argument such as:</p> <pre><%=ecmShowSiteLanguage('mysitelanguage')%></pre>

Custom API

Formatting for the custom ASP functions used in Ektron CMS200/300 for ASP is defined in the file `ApplicationAPI.asp`. As a developer, you may change this file to enhance the appearance of the custom functions on your Web site.

However, Ektron recommends creating a copy of the `ApplicationAPI.asp` file and renaming it, for example `MyCustomAPI.asp`. Then, include that file when creating Ektron CMS200/300 templates. This renaming ensures that your custom API is not overwritten when you upgrade to another version of Ektron CMS200/300.

Multi-Level Collection

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

Ektron CMS200/300's Menu feature lets users create and maintain a dropdown menu system for your Web site. Also known as a Multi-level Collection, the Menu feature lets you display menu options that link to content blocks, library assets, external hyperlink and sub-menus.

To learn more about menus, please refer to the "Menu Feature" chapter of the **Ektron CMS200/300** User Manual.

Customizing the Menu

Before you display the menu on a web page, you can customize it. Customizing a menu gives you the flexibility of displaying the menu as per your liking.

There are two ways of customizing a menu in **Ektron CMS200/300**. This can be done by making changes to:

- `ek_menu.js`
- `ApplicationAPI`

This section explains to customize both files.

`ek_Menu.js`

The first way to change the look and feel of the menu is by making changes to `ek_menu.js` file. If you used the default settings when installing **Ektron CMS200/300**, this file can be found under `c:\inetpub\wwwroot\CMS300Sample\CMS300Scripts\Java\`.

Using the following table as a reference, you can make changes to the menu attributes.

Attribute Name	Description	Default Value	See Also
version	The Ektron CMS200/300 version.		"Version" on page 135
type	Specify the type of resource which, in this case, is Menu.		"Type" on page 136
fontSize	Specify the font size for the menu.	14	"FontSize" on page 136
fontWeight	Specify the thickness of the font that is displaying the menu. FontWeight can be set based on name or value.	Plain	"FontWeight" on page 136
fontFamily	Specify the name of font that will be used to display the text for the menu.	arial, helvetica, espy, sans-serif	"FontFamily" on page 137
fontColor	Specify the font color that will be used to display the text for the menu and sub menus under it.	#000000	"FontColor" on page 138
fontColorHiLite	Specify the font color that will be used to display the text for the menu when you highlight the menu item.	#ffffff	"FontColorHilite" on page 138
bgColor	Specify the background color of the table cell that displays each menu item.	#555555	"bgColor" on page 139
menuBorder	Specify the width of the border that will be displayed around the main menu frame.	1	"menuBorder" on page 140
menuItemBorder	Specify the width of the border that will be displayed around each menu item cell.	1	"menuItemBorder" on page 140
menuItemBgColor	Specify the background color that will be displayed around the menu item.	#6699ff	"menuItemBgColor" on page 141

Attribute Name	Description	Default Value	See Also
menuLiteBgColor	Specify the width of the background color of the border that surrounds the main menu frame.	#ffffff	"menuLiteBgColor" on page 142
menuBorderBgColor	Specify the width of the background color that will be displayed for the menu border.	#6699ff	"menuBorderBgColor" on page 142
menuHiLiteBgColor	Specify the background color that will be displayed when you highlight a menu item.to	#000084	"menuHiLiteBgColor" on page 143
menuContainerBgColor	Specify the background color that will be displayed for the menu container.	#6699ff	"menuContainerBgColor" on page 143
childMenuIcon	Specify the path for the submenu icon.	./ cms300scripts/ images/ application/ arrows.gif	"childMenuIcon" on page 144
childMenuIconHiLite	Specify the path for the icon that changes when you hover the mouse over the submenu.o	./ cms300scripts/ images/ application/ arrows2.gif	"childMenuIconHiLite" on page 144

Below is a brief description of each menu attribute.

Version

The version attribute is used to specify the version for **Ektron CMS200/300**. The version can be set as anything.

Example:

To change the version of the menu to something other than the default, say CMS300 v45, change the following line:

```
this.version = "CMS300 v45";
```

Type

The type attribute is used to specify the resource type. The type can be set as anything.

Example:

To change name of the Type attribute, to say Multi-Level Collection, make changes to the following line:

```
this.type = "Multi-Level Collection";
```

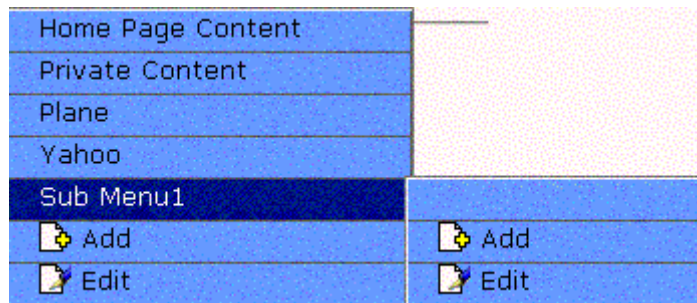
FontSize

The fontSize attribute is used to specify the size of the font that will be used to display the text of the menu. The size of the font can be set as any value.

Example:

To change the fontSize of the menu to a different size, say 16, make changes to the following line:

```
this.fontSize = 16;
```



FontWeight

The fontWeight attribute is used to specify the thickness of the font to display the text of the menu. This attribute can take the value from 0-1000 to display the thickness (weight) of the font. To set the FontWeight, you can either set it by name such as Bold or Plain, or by its value.

Example:

To change the fontWeight of the menu to a different style, say Bold, make changes to the following line:


```
this.fontWeight = Bold;  
or  
this.fontWeight = 700;
```



NOTE: Both values display the same result.

FontFamily

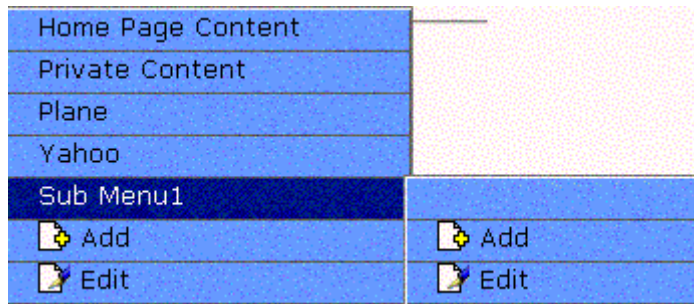
The `fontFamily` attribute is used to specify the name of font that will be used to display the text for the menu. The name of the font that is displayed is dependent on the one that is supported by the browser. This property can be set to multiple comma-separated values.

NOTE In order for the menu to display the `fontfamily` other than the default, you need to specify only the font name and no other comma-separated values. However, if you do specify more than one value for the `fontfamily` as mentioned in the example below, the application uses the default IE settings.

Example:

To change the `fontFamily` of the menu to a different name, say `arial, helvetica, espy, sans-serif`, make changes to the following line:

```
this.fontFamily = "arial, helvetica, espy, sans-serif";
```



FontColor

The fontColor attribute is used to specify the font color that will be used to display the text for the menu and sub menus under it.

Example:

To change the fontColor of the menu to a different color, say (aquamarine) #7FFFD4, make changes to the following line:

```
this.fontColor = "#7FFFD4";  
or  
this.fontColor = "aquamarine"
```

NOTE: To make changes to the color, you can either specify the color name or the color code.



FontColorHilite

The fontColorHilite attribute is used to specify the font color that will be used to display the text for the menu when you highlight the menu item.

Example:

To change the `fontColorHilite` of the menu to a different color, say (crimson)#DC143C, make changes to the following line:

```
this.fontColorHilite = "#DC143C";  
or  
this.fontColorHilite = "crimson";
```

NOTE: To make changes to the color, you can either specify the color name or the color code.



bgColor

The `bgColor` attribute is used to specify the background color of the table cell that displays each menu items.

Example:

To change the `bgColor` of the menu to a different color, say (lightsalmon)#FFA07A, make changes to the following line:

```
this.bgColor = "#FFA07A";  
or  
this.bgColor = "lightsalmon";
```

NOTE: To make changes to the color, you can either specify the color name or the color code.



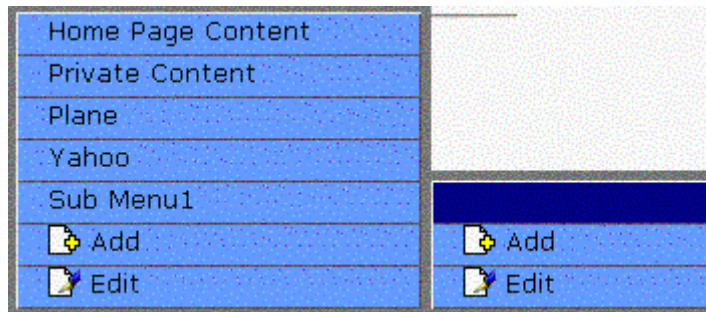
menuBorder

The menuBorder attribute is used to specify the width of the border that will be displayed around the menu frame. This includes the main menu as well as the sub-menu frame.

Example:

To change the menuBorder to a different value, say 5, make changes to the following line:

```
this.menuBorder = 5;
```



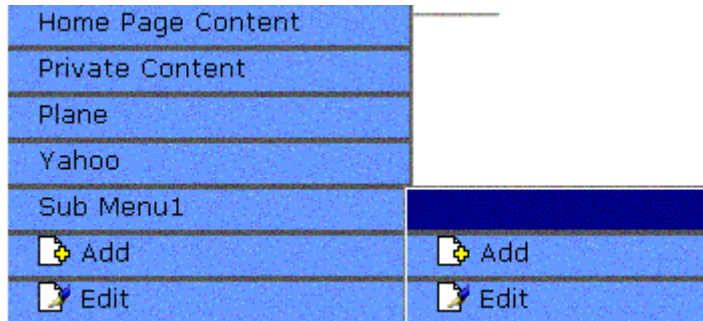
menuItemBorder

The menuItemBorder attribute is used to specify the width of the border that will be displayed around each menu item cell. This includes the main menu cell as well as the sub-menu cell.

Example:

To change the menuItemBorder to a different value, say 3, make changes to the following line:

```
this.menuItemBorder = 3;
```



menuItemBgColor

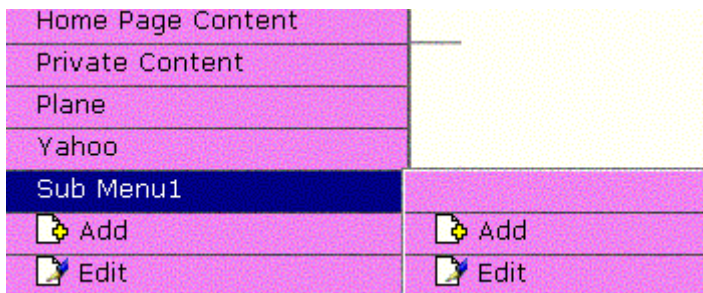
The menuItemBgColor attribute is used to specify the background color that will be displayed around the menu item. This includes the main menu as well as the sub-menu.

Example:

To change the menuItemBgColor to a different color, say (violet) #EE82EE, make changes to the following line:

```
this.menuItemBgColor = "#EE82EE";  
or  
this.menuItemBgColor = "violet";
```

NOTE: To make changes to the color, you can either specify the color name or the color code.



menuLiteBgColor

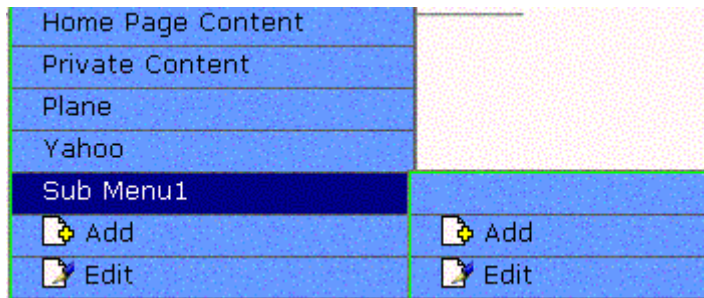
The menuLiteBgColor attribute is used to specify the width of the background color of the border that surrounds the main menu as well as the sub-menu frame.

Example:

To change the menuLiteBgColor to a different color, say (lime)#00FF00, make changes to the following line:

```
this.menuItemBgColor = "#00FF00";  
or  
this.menuItemBgColor = "lime";
```

NOTE: To make changes to the color, you can either specify the color name or the color code.



menuBorderBgColor

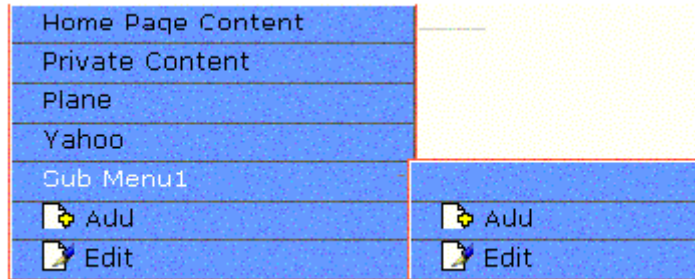
The menuBorderBgColor attribute is used to specify the width of the background color that will be displayed for the menu border. This includes the main menu as well as the sub-menu.

Example:

To change the menuBorderBgColor to a different color, say (tomato)#FF6347, make changes to the following line:

```
this.menuBorderBgColor = "#FF6347";  
or  
this.menuBorderBgColor = "tomato";
```

NOTE: To make changes to the color, you can either specify the color name or the color code.



menuHiLiteBgColor

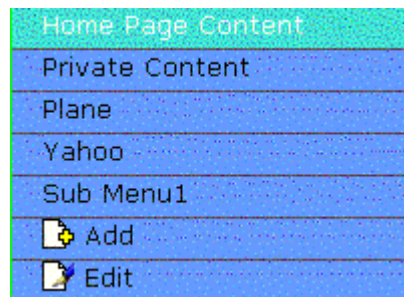
The menuHiLiteBgColor attribute is used to specify the background color that will be displayed when you highlight a menu. This includes the main menu as well as the sub-menu.

Example:

To change the menuHiLiteBgColor to a different color, say (turquoise) #40E0D0, make changes to the following line:

```
this.menuHiliteBgColor = "#40E0D0";
or
this.menuHiliteBgColor = "turquoise";
```

NOTE: To make changes to the color, you can either specify the color name or the color code.



menuContainerBgColor

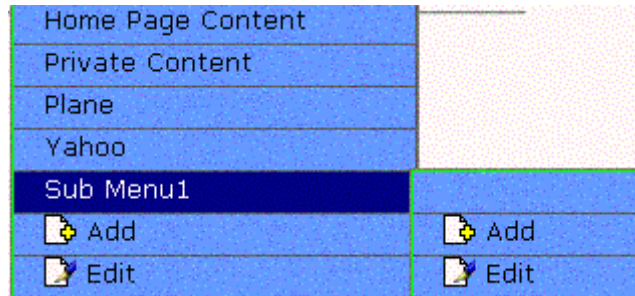
The menuContainerBgColor attribute is used to specify the background color that will be displayed for the container.

The container can be considered as a <div> tag that displays the menu. When you set the container color, it only gets displayed at the top and left side of the menu frame.

Example:

To change the menuContainerBgColor to a different color, say (lime)#00FF00, make changes to the following line:

```
this.menuContainerBgColor = "#00FF00";
```



childMenuIcon

The childMenuIcon attribute is used to specify the path to display an image if an item is a submenu.

To display a different image, you can either provide an external path for the image or upload the image into **Ektron CMS200/300** and then provide its path.

Example:

To change the path of the childMenuIcon, make changes to the following line in ek_menu.js file.

```
this.childMenuIcon = "../cms300scripts/images/application/arrows.gif";
```

childMenuIconHilite

The childMenuIconHilite attribute is used to specify the path for the image that is displayed when you hover the mouse over the submenu.

To display a different image, you can either provide an external path for the image or upload the image into **Ektron CMS200/300** and then provide its path.

Example:

To change the path of the childMenuIconHilite, make changes to the following line:

```
this.childMenuIconHilite = "../cms300scripts/images/application/arrows2.gif";
```

Application API

The second way of customizing a menu, is by making changes to the ApplicationAPI.

This section will talk about how to customize the menu by making changes to ApplicationAPI.

Customizing ApplicationAPI

Begin by navigating to ApplicationAPI file. If you used the default settings when installing **Ektron CMS200/300**, this file will be installed under C:/inetpub/wwwroot/CMS300Sample/CMS300Scripts/

.

There are two places from where you can customize the menu in ApplicationAPI.

- Sub-Menu
- Main Menu

We will talk about both sections.

Sub-Menu

After you open ApplicationAPI file, scroll through the section on Function AddSubMenu.

Under this function, there are a few lines of code as mentioned below that will allow you to change the look and feel of the sub-menu.

```
'html = html & "Menu" & Menu_ID & ".menuItemBgColor = 'red';" & VbCrLf  
'html = html & "Menu" & Menu_ID & ".menuLiteBgColor = '#6699ff';" & VbCrLf  
'html = html & "Menu" & Menu_ID & ".menuBorderBgColor = 'red';" & VbCrLf  
'html = html & "Menu" & Menu_ID & ".menuHiliteBgColor = '#6699ff';" & VbCrLf  
'html = html & "Menu" & Menu_ID & ".menuContainerBgColor = 'red';" & VbCrLf
```

Uncomment these lines of code and make changes to the submenu attributes.

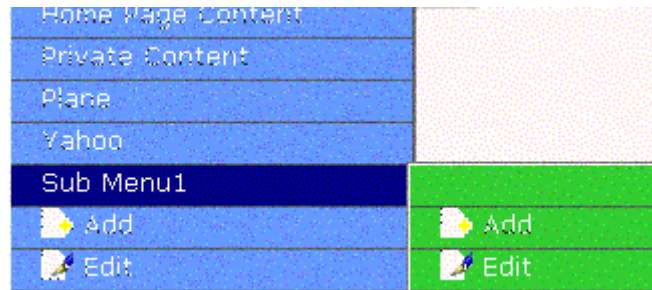
menuItemBgColor

The menuItemBgColor attribute is used to specify the background color that will be displayed around the sub-menu item.

For Example:

To change the default menuItemBgColor from red to lets say lime green, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuItemBgColor = 'limegreen';" & VbCrLf
or
html = html & "Menu" & Menu_ID & ".menuItemBgColor = '#00FF00';" & VbCrLf
```



NOTE

To change the color, you can specify the color name or the color code.

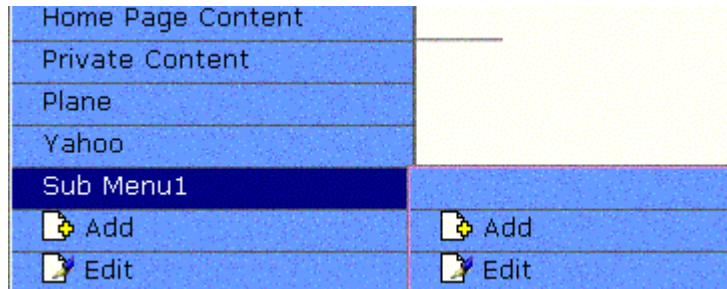
menuLiteBgColor

The menuLiteBgColor attribute is used to specify the width of the background color of the border that surrounds the sub-menu frame.

For Example:

To change the default menuLiteBgColor from blue to lets say violet, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuLiteBgColor = 'violet';" & VbCrLf
or
html = html & "Menu" & Menu_ID & ".menuLiteBgColor = '#EE82EE';" & VbCrLf
```



To make changes to the color, you can either specify the color name or the color code.

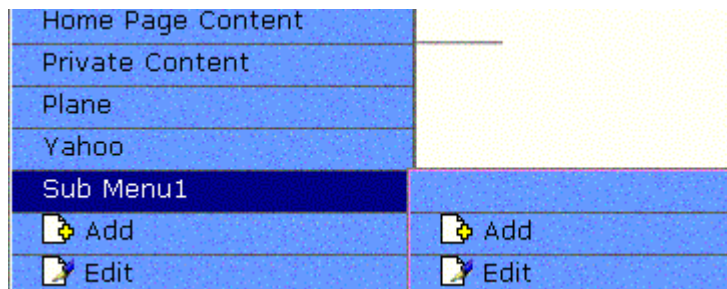
menuBorderBgColor

The menuBorderBgColor attribute is used to specify the width of the background color that will be displayed for the sub-menu border.

For Example:

To change the default menuBorderBgColor from red to lets say turquoise, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuBorderBgColor = 'turquoise';" & VbCrLf
or
html = html & "Menu" & Menu_ID & ".menuBorderBgColor = '#90EE90';" & VbCrLf
```



To make changes to the color, you can either specify the color name or the color code.

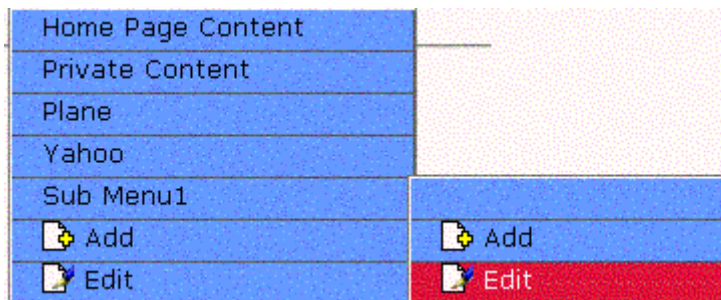
menuHiliteBgColor

The menuHiLiteBgColor attribute is used to specify the background color that will be displayed when you highlight a sub-menu item.

For Example:

To change the default menuHiliteBgColor from blue to lets say crimson, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuHiliteBgColor = 'crimson';" & VbCrLf
or
html = html & "Menu" & Menu_ID & ".menuHiltemBgColor = '#DC143C';" & VbCrLf
```



To make changes to the color, you can either specify the color name or the color code.

menuContainerBgColor

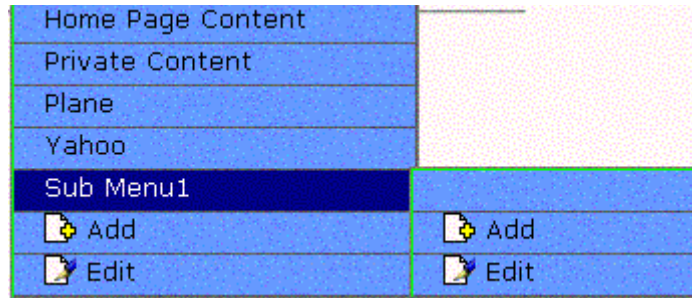
The menuContainerBgColor attribute is used to specify the width of the background color that will be displayed for the container.

The container can be considered as a <div> tag that displays the menu. When you set the container color, it only gets displayed at the top and left side of the menu frame.

For Example:

To change the default menuContainerBgColor from red to lets say lime, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuContainerBgColor = 'lime';" & VbCrLf
or
html = html & "Menu" & Menu_ID & ".menuContainerBgColor = '#00FF00';" & VbCrLf
```



To make changes to the color, you can either specify the color name or the color code.

Main Menu

After you open ApplicationAPI file, scroll through the section on Function `ecmDHTML_GenerateMenu`.

Under this function, there are a few lines of code as mentioned below that will allow you to change the look and feel of the menu.

NOTE Any changes that you make in the ApplicationAPI, get reflected in the `ek_menu.js` file as well.

```
'html = html & "Menu" & menu_id & ".menuItemBgColor = 'white';" & VbCrLf
'html = html & "Menu" & menu_id & ".menuLiteBgColor = '#6699ff';" & VbCrLf
'html = html & "Menu" & menu_id & ".menuBorderBgColor = 'white';" & VbCrLf
'html = html & "Menu" & menu_id & ".menuHiliteBgColor = '#6699ff';" & VbCrLf
'html = html & "Menu" & menu_id & ".menuContainerBgColor = 'white';" & VbCrLf
```

Uncomment these lines of code and make changes to the menu attributes.

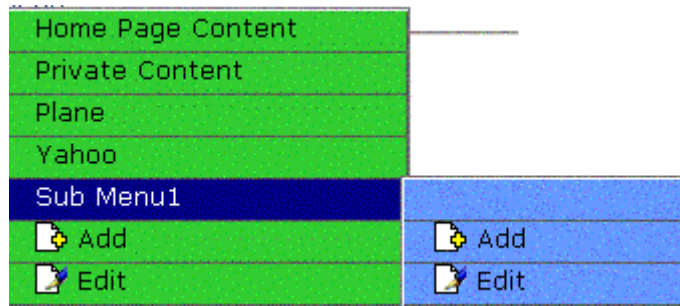
menuItemBgColor

The `menuItemBgColor` attribute is used to specify the background color that will be displayed around the main menu item.

For Example:

To change the default `menuItemBgColor` from white to lets say limegreen, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuItemBgColor = 'limegreen';" & VbCrLf
or
html = html & "Menu" & Menu_ID & ".menuItemBgColor = '#00FF00';" & VbCrLf
```



To make changes to the color, you can either specify the color name or the color code.

menuLiteBgColor

The menuLiteBgColor attribute is used to specify the width of the background color of the border that surrounds the main menu frame.

For Example:

To change the default menuLiteBgColor from light azure blue to lets say yellow, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuLiteBgColor = 'yellow';" & VbCrLf
or
html = html & "Menu" & Menu_ID & ".menuLiteBgColor = '#FFFF00';" & VbCrLf
```



To make changes to the color, you can either specify the color name or the color code.

menuBorderBgColor

The menuBorderBgColor attribute is used to specify the width of the background color that will be displayed for the main menu border.

For Example:

To change the default menuBorderBgColor from white to lets say gray, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuBorderBgColor = 'gray';" & VbCrLf
or
html = html & "Menu" & Menu_ID & ".menuBorderBgColor = '#333333';" & VbCrLf
```



To make changes to the color, you can either specify the color name or the color code.

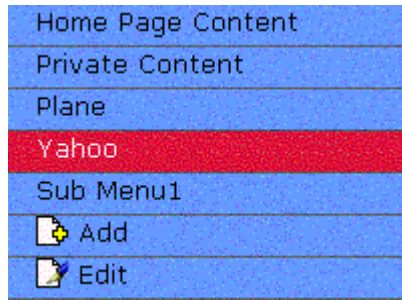
menuHiliteBgColor

The menuHiliteBgColor attribute is used to specify the background color that will be displayed when you highlight the main menu item.

For Example:

To change the default menuHiliteBgColor from black to lets say crimson, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuHiliteBgColor = 'crimson';" & VbCrLf
or
html = html & "Menu" & Menu_ID & ".menuHiltemBgColor = '#DC143C';" & VbCrLf
```



To make changes to the color, you can either specify the color name or the color code.

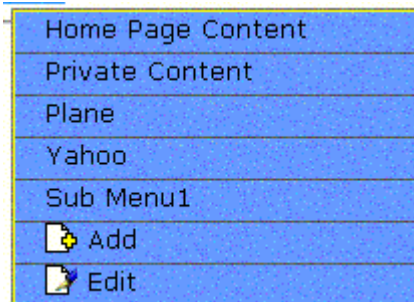
menuContainerBgColor

The menuContainerBgColor attribute is used to specify the background color that will be displayed for container.

For Example:

To change the default menuContainerBgColor from red to lets say aquamarine, make the following changes to this line of code:

```
html = html & "Menu" & Menu_ID & ".menuContainerBgColor = 'aquamarine';" & VbCrLf  
or  
html = html & "Menu" & Menu_ID & ".menuContainerBgColor = '#7FFFD4';" & VbCrLf
```



To make changes to the color, you can either specify the color name or the color code.

Customizing ek_menu.js versus ApplicationAPI

Though you can customize the menu via ek_menu.js or file, the way of customizing these files is very different.

When customizing the **ek_menu.js** file, changes to a menu attribute affect the appearance of the menu and its sub-menus.

When customizing **ApplicationAPI** file, you can choose to change the main menu, sub-menu, or both.

Custom ASP.Net Functions

Ektron CMS200/300 maintains and display content through the use of custom Ektron ASPX functions and ComObjects. On your template, you must include the following file in order to be able to use the custom functions:

```
<!-- #include file="site_scripts_path.aspx" -->
```

This file will call the file with the custom Ektron functions to allow the page to function properly and must appear at the top of the template file.

To customize you Ektron CMS200/300 application, you will need to be familiar with these tags:

Tag	Description	More Information
Add Content	Programmatically adds content block(s) to Ektron CMS200/300	"Add Content" on page 165
Collection	Displays the links generated by collections.	"Collection Function" on page 201
Content Block	Displays and manages a content block.	"Content Block Function" on page 159
Event Calendar	Displays an event calendar	"Event Calendar Function" on page 219
Form	Displays a content block that is associated with a form.	"Form Function" on page 225
List Summary	Displays the list of the content block summaries.	"List Summary Function" on page 186
Login	Paints the login or logout button which allows the user to login and use the Ektron CMS200/300 application.	"Login Function" on page 155

Tag	Description	More Information
Metadata	Displays the metadata in the source of the Web page the content is displayed on.	"Metadata Function" on page 229
MultiPurpose	Can display either a content block or a form.	"MultiPurpose Function" on page 227
Random Content	Randomly displays the body content of a collection item.	"Random Content" on page 217
Random Summary	Randomly displays the summary content of a collection item.	"Random Summary" on page 218
Search	Inserts a text box on the template with a search button.	"Search Function" on page 170
Search Display	Creates area where search results will be displayed.	"Search Display Function" on page 181
Single Summary	Displays the summary for a single content block.	"Single Summary Function" on page 195
Show Site Language Function	Lets user select language in which to view Web site.	"Show Site Language Function" on page 228



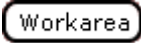


Login Function

The login function paints a login button, on the template the login function is inserted on, when displayed in a browser.

Displayed here is the format for this special login function:

```
<% ecmLogin %>
```

This tag is responsible for displaying the following buttons on the Web page:

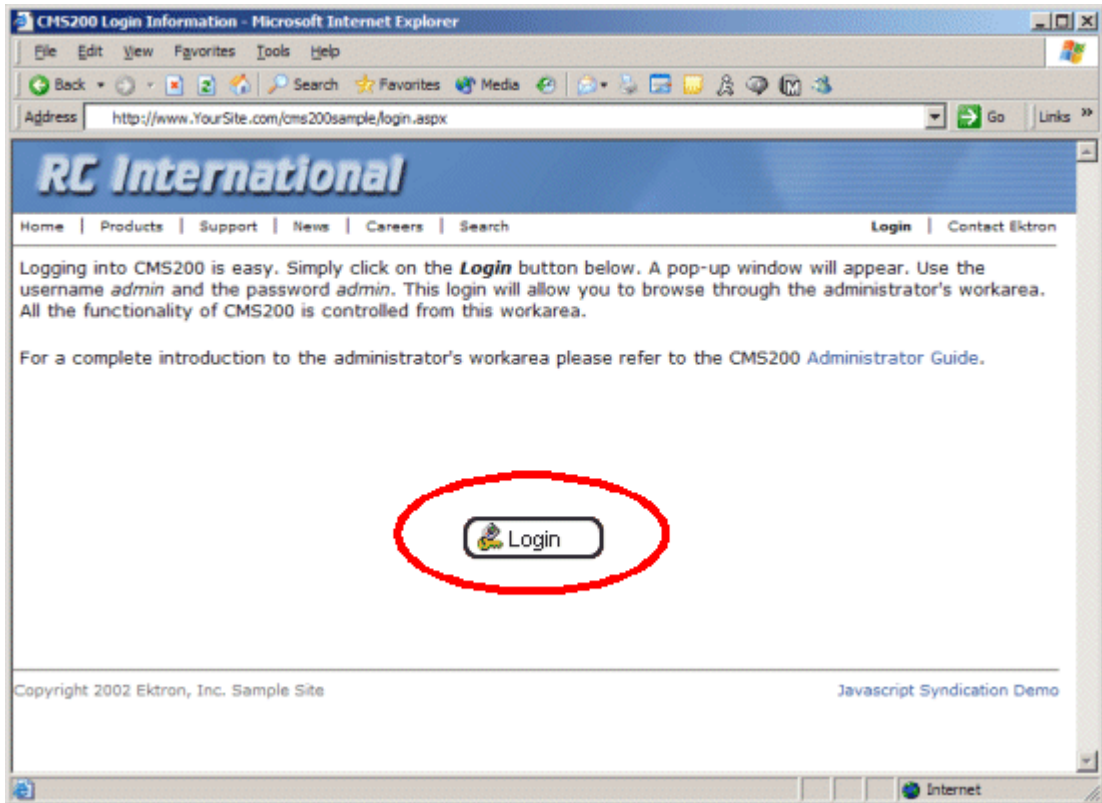
Button	Name	Description
	Login	When user is not logged into Ektron CMS200/300, this button will be displayed. Clicking this button will allow the user to log into the application.
	Logout	Once logged in, this button will replace the login button to allow the user to log out from the Ektron CMS200/300 Web site.
	Workarea	When logged in, this button will be displayed under the logout button allowing the user to access their workarea.
	Preview On	This button will allow the user to preview the entire Web site as if all checked-in content was published.
	Preview Off	Turns off the site preview mode.

Login Tag Example

The following example places a login button in a table cell inside a asp template.

```
<tr>
  <td>
    <% ecmLogin %>
  </td>
</tr>
```

When inserted properly, a login button will be placed on the Web page.



Clicking on the login tag will open the login window where a user will be able to enter their username and password. Upon successful completion, the user will then be logged into the Ektron CMS200/300 Web site.

Placing Your Login Button

You are able to add as many login buttons to a template as you would like. You might want to include a login button on each template, or have a special Web page called login.aspx where you can log in there without the public being able to access the login page.

Partner Login



Ektron CMS200/300 also includes a special login function that only displays a login or logout button on the template.

Displayed here is the format for this special login function:

```
<% ecmLoginPartner %>
```

As stated before, when inserted onto a template, only a login or logout button is displayed.

The following table displays how the function appears when the user is logged in or logged out.

Logged Out	Logged In
<p>administrator's workarea please refer to the</p> 	<p>administrator's workarea please refer to the CM:</p> 

Content Block Function

The content block tag is used to display a content block on an Ektron CMS200/300 Web page.

Here is the format for the content block function.

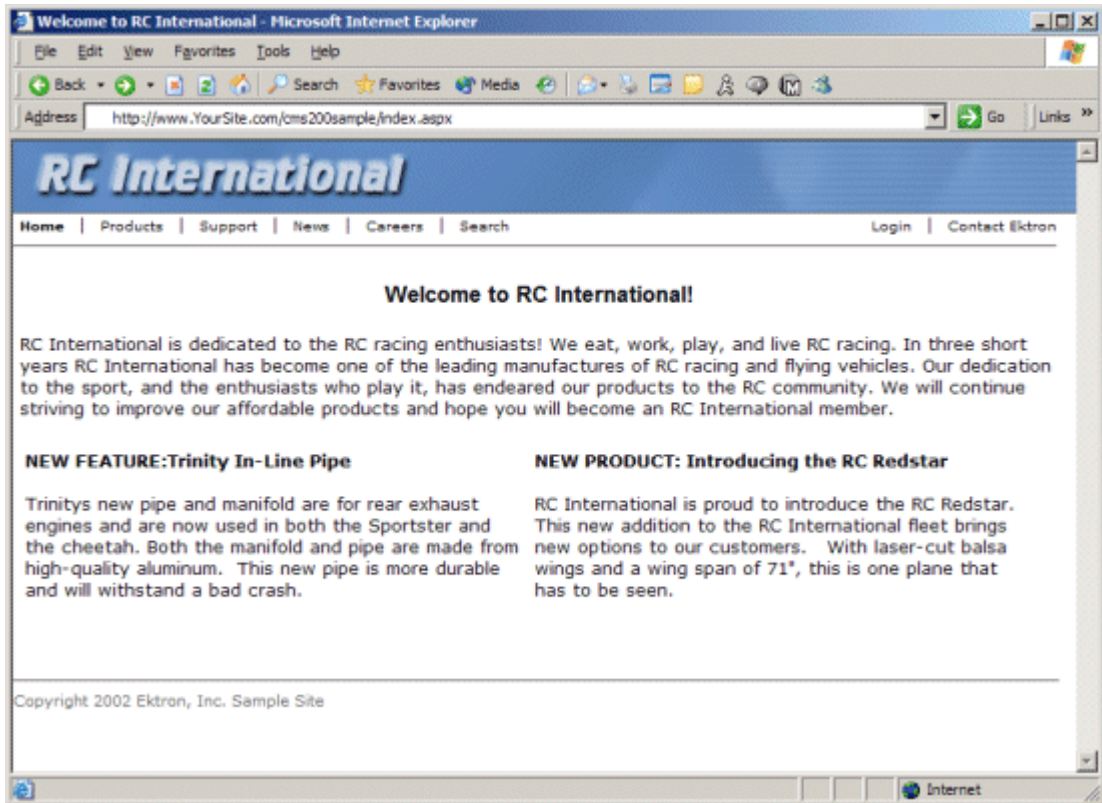
<% ecmContentBlockEx (ID, "XSLT", Override) %>

NOTE As an alternative, you can use the multipurpose function, which can display either a content block or a content block associated with a form. For more information, see ["MultiPurpose Function" on page 227](#).

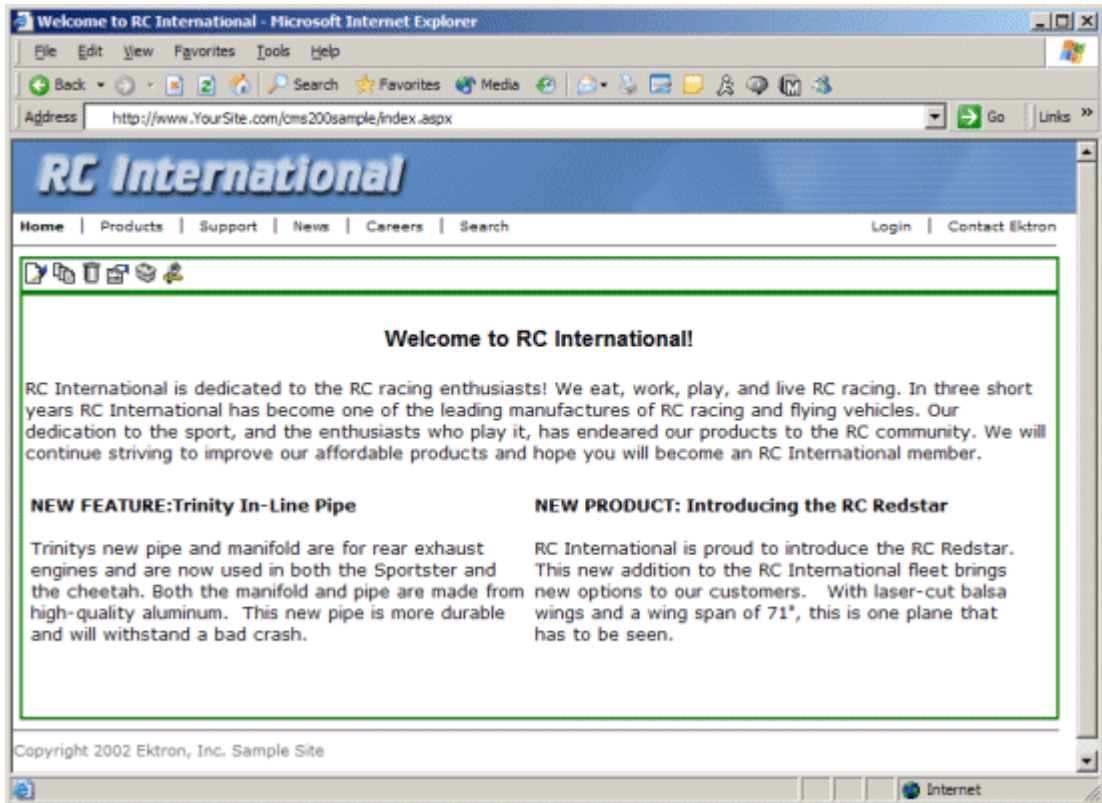
The attributes of the content block function are explained below.

Attribute	Description	Required
ID	ID number assigned to content block when created.	Yes
XSLT	File name and path to a valid XSLT to transform the XML content.	No
Override	If no XSLT is specified, enter 1 to use XSLT1, 2 for XSLT2, or 3 for XSLT3.	No

In normal view, this tag retrieves the content block from the database and returns it to the calling template.



If a user is logged into Ektron CMS200/300, the content block tag will paint a border around the content block and provide the menu for managing that piece of content.



Content Block Examples

In Ektron CMS200/300, there are three different types of content blocks:

- Static content blocks
- Dynamic content blocks
- XML Content Blocks

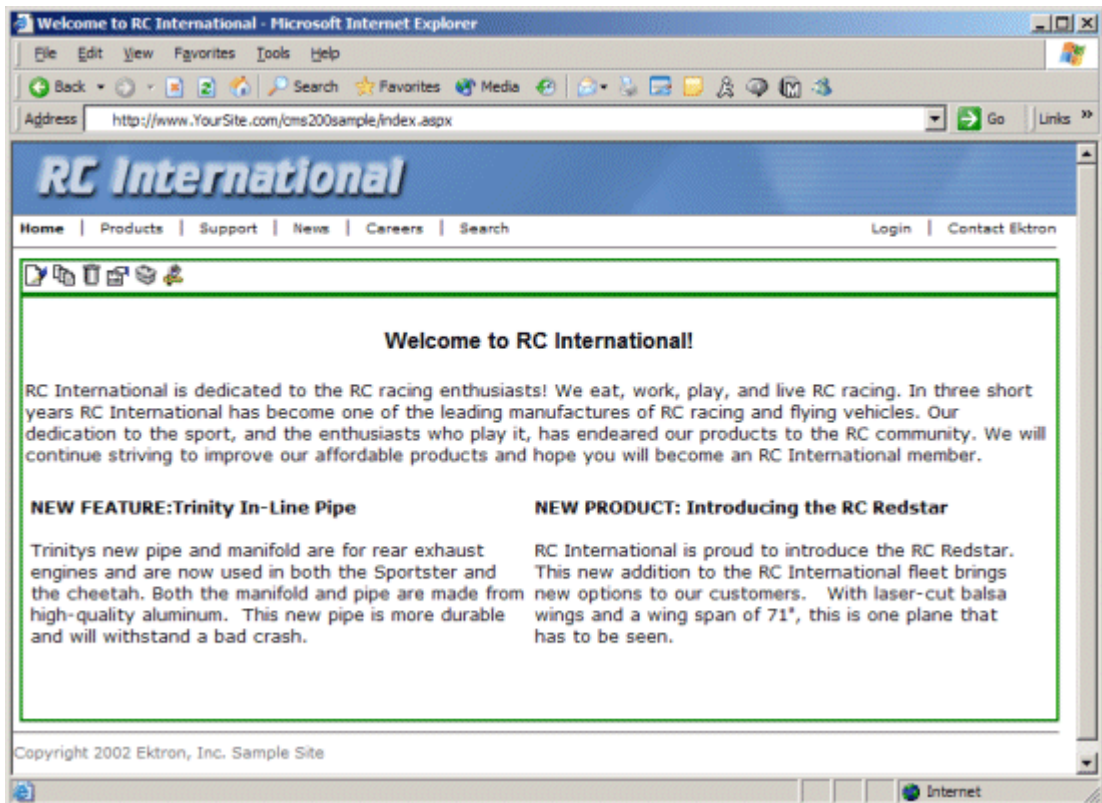
Static content blocks are used to display one specific content block on the Web page, where a dynamic content block is used to display the content block of the ID that is passed through the URL. XML content blocks are used to display XML content transformed by an XSLT specified in the function.

Static Content Block

The following example explains what to insert for a static content block on a template.

```
<tr>
  <td>
    <% ecmContentBlock(1) %>
  </td>
</tr>
```

In the example displayed above, the function will call the content block with an id of 1 from the database to the browser.



Dynamic Content Block

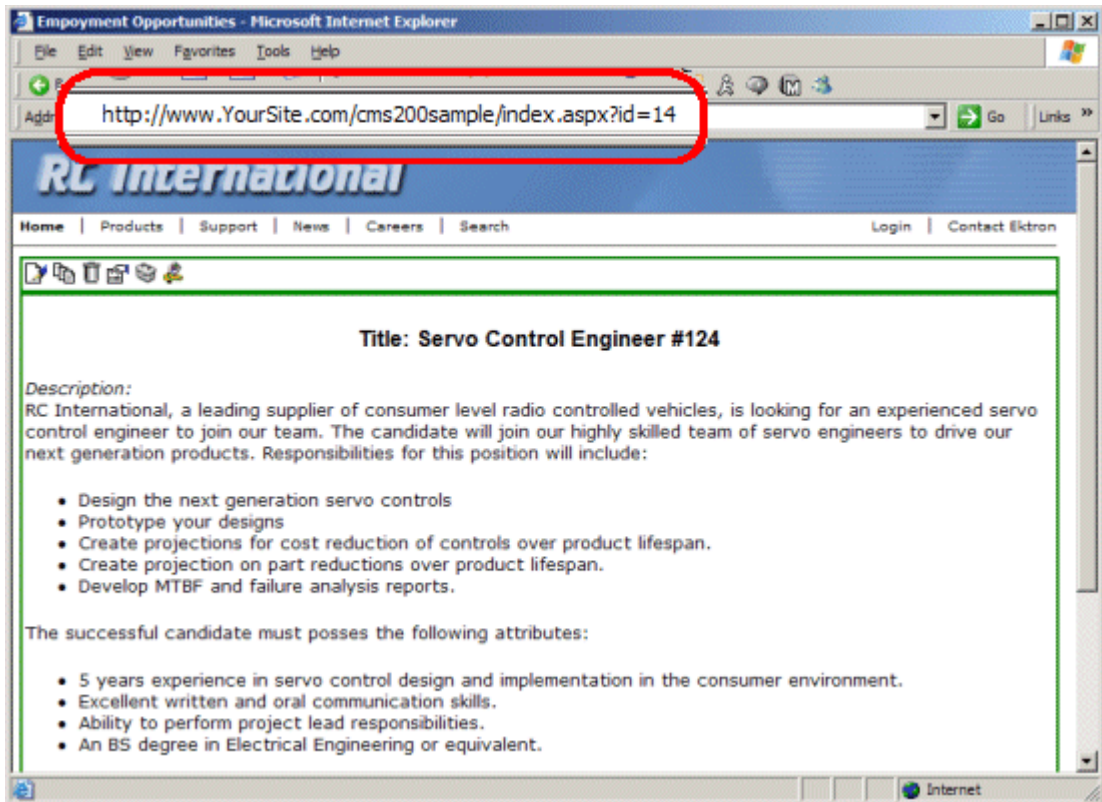
The following example shows how the content block tag can be used where the id is being passed in as a URL parameter. This is the tag which would be used on a dynamic template.

```

<tr>
  <td>
    <% ecmContentBlock(request.QueryString("id")) %>
  </td>
</tr>

```

When a user accesses this page and adds the ID number to the template this tag lives on, the content block with the corresponding ID will be displayed on the Web page in the template that is specified.



XML Content Block

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

Here is an example of displaying an XML content block. Note that you specify both a content block and an XSLT that determines how the content block is displayed.

```
<tr>
  <td>
    <% ecmContentBlockEx (13, "", 1) %>
  </td>
</tr>
```

The previous example would display the content block ID=13, and use the XSLT1 display XSLT instead of the default.

To learn about the function's arguments, see ["The Function's Arguments" on page 53](#).

Dynamic and Static Combination

Here is an example of how you can use the two different content tags in the same table cell by using an IF statement

```
<tr>
  <td>
    <%
      if request.QueryString("id") <> "" then
        ecmContentBlock(request.QueryString("id"))
      else
        ecmContentBlock(35)
      End if %>
  </td>
</tr>
```

In this sample code, when a person accesses this page without passing through the content's ID number, the content with an ID=14 will be displayed. If you pass a content id through the URL like **index.aspx?id=4**, the content block ID=4 will be displayed.

Add Content

The add content function allows you to programmatically add a new content block to Ektron CMS200/300 without the need of logging in.

This function can be for several applications including, but not limited to:

- Applied to a Web form allowing you to collect data from users about the content block.
- Creating a loop function that moves information from a database to an Ektron CMS200/300 database

Shown here is the format for the add content function:

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

Explained in the table below are the attributes for the add content block function.

Attribute	Description
UserName	The username of the user to be authenticated for permissions to see whether the user is able to add a content block.

Attribute	Description
Password	The password for the username being authenticated.
Domain	If using Active Directory, specify the domain needed to authenticate the user.
Content_title	Specify a title for the content block being added.
Content_comment	Specify a historical comment for the content being added.
ContentHTML	Actual content for the content block.
SummaryHTML	Summary information about the content block.
ContentLanguage	Language of content.
FolderID	ID number of the content folder where the content block will be added to.
GoLive	Date and time the content will go live. Must match date/time format set in setup.asp.
EndDate	Date and time the content expires Must match date/time format set in setup.asp.
MetalInfo	An metadata information
ErrString	String returned if the content fails to get added to Ektron CMS200/300.

Add Content Example

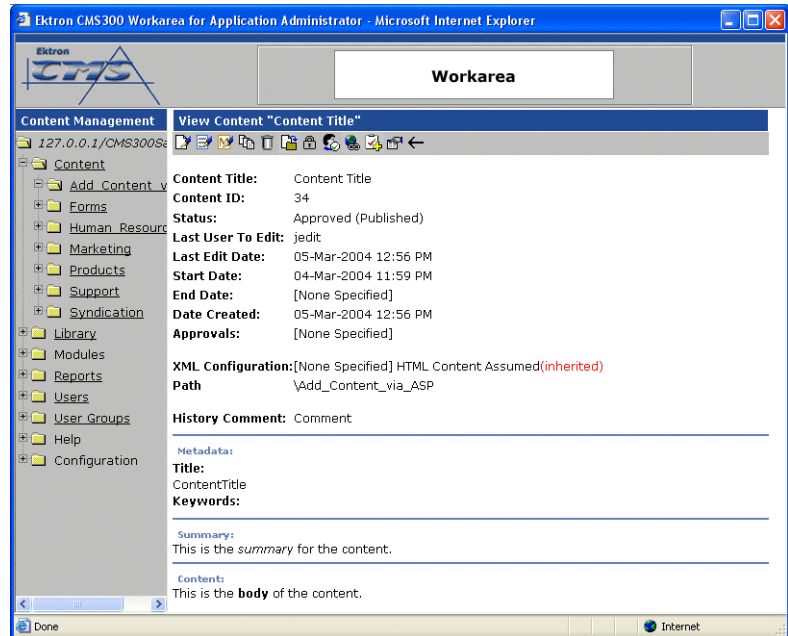
The following is an example of the add content function hard coded in a template:.

```
<tr>
  <td>
    <% AddContent ("jedit", "jedit", "", "Content Title", "Comment",
      "This is the <b>body</b> of the content.",
      "This is the <i>summary</i> for the content.", 1, 14,
      "03-04-2004 11:59 PM", "",
      "<metadata><meta id=\"3\">ContentTitle</meta></metadata>",
      "") %>
  </td>
</tr>
```

The previous example has the following properties:

- The user jedit is used to be authenticated
- 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 = 1, or English.
- The new content block is added to content folder ID=14
- The go live date is set at March 3, 2004 at 11:59 PM
- No end date is specified
- The metadata “Content title” is set for the metadata definition ID=3

When this function is executed, the content is added to Ektron CMS200/300. Displayed here is the record of the content in the workarea.



Add Content Web Form Example

Displayed here is an example of a Web for that is included with Ektron CMS200/300. This form can be found by default in the Ektron CMS200/300 root directory.

Add Content Programatically To CMS

Username:

Password:

Domain: (when using Active Directory)

Title: (as text)

Comment: (as text)

Content: (as HTML content)

Summary: (as HTML content)

Meta Data (Title): (as text)

Meta Data (Keywords): (separated by ;)

Language: (1 for English, 1031 for German, 1036 for French)

FolderID: (numeric, 0 for root)

Start Date: (as mm-dd-yy)

End Date: (as mm-dd-yy)

The form is set up to pass the data into the addcontent function.

Search Function

The search tag is used to display a text box, with a search button on the template. When seen on the template on the Web page, the users will be allowed to enter text and click on the search button to search through all the content on the Web site.

Shown here is the format for the ecmSearch tag:

```
<% ecmSearch( "Starting Folder",
              Recursive,
              "Target Page",
              Text Box Size,
              Maximum Characters,
              "Button Image Source",
              "Button Text",
              "Font Face",
              "Font Color",
              "Font Size",
              Horizontal,
              "Spare "
            ) %>
```

The following table provides information about all the variables for the search tag.

Attribute	Description	Required?
Starting Folder	The folder you wish to search	Yes
Recursive	1 or 0 (zero). Whether to search sub-folders. (0=No)	Yes
Target Page	The Web page to output the search results.	Yes
Text Box	The size of the display text box for user input.	Yes
Maximum Characters	The maximum characters the input text box should accept.	No

Attribute	Description	Required?
Button Image Source	The image location for the search button. Blank ("") = standard submit button.	No
Button Text	The text used for the button if the standard submit button is chosen, or alternative text for the button if an image is chosen	No
Font Face	The font used for the text in the search tag.	No
Font Color	The font color used for the text in the search tag. Blank ("") = Black.	No
Font Size	The font size for the text in the search tag.	No
Horizontal	Determines if the search options are in a row or a column	Yes
Spare1	Currently not used	Must Be ""

Search Tag Example

The following example places a text field and a search button in a table cell inside an ASP template.

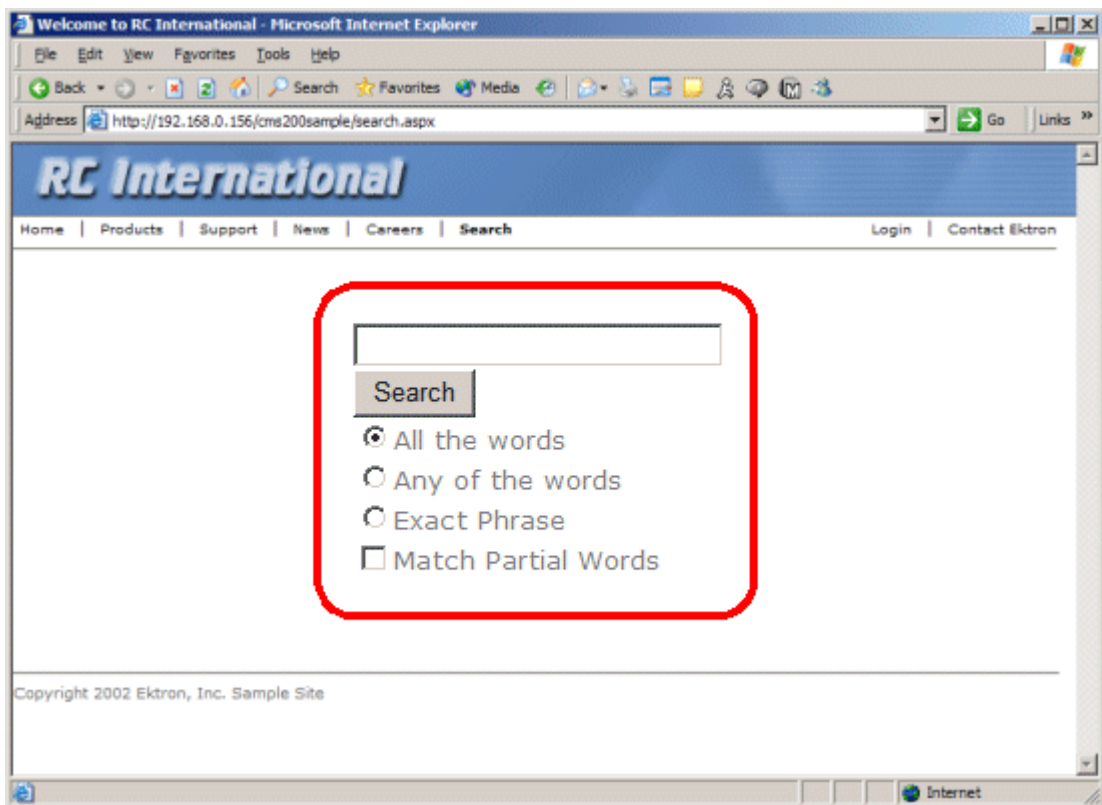
```
<tr>
  <td>
    <% ecmSearch ("\\marketing",1,"search-
display.aspx",25,200,"","Search","Verdana","#808080","2",0,"") %>
  </td>
</tr>
```

The previous example has the following properties:

- Searches content in the \\content\\marketing folder
- Search is recursive
- The target page is "search-display.aspx"
- The size of the text box is 25
- The search will allow a maximum of 200 characters

- No image is used for a button, instead a standard submit button is used.
- The text on the submit button is “Search”
- The search options appear in Verdana with a color of #808080 and a size of 2pt.
- The search options are arranged vertically

When added to a template, and visited, the search tag will appear similar to the following:



Search Tag Variables

Listed below are all the different variables and their descriptions about them.

Starting Folder

In this variable, you will need to specify which content folder you will want to begin the search. There are two options for this variable:

Value	Description
"\"	Search will begin the search at the root folder
"folder path"	Search will begin from the folder specified

Recursive

When you setup the search to be recursive, you are allowing the search to search all the sub-folders associated with the starting folder you had specified. You have two options for this variable:

Value	Description
1	Recursive search (will search sub-folders of the starting folder)
0	Non-Recursive search (will only search starting folder)

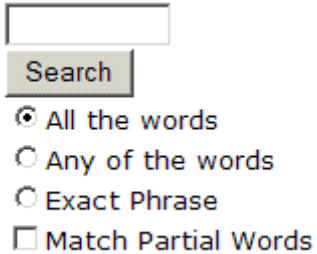
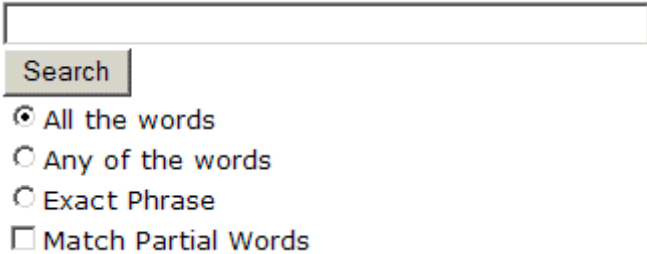
Target Page

The target page is the page where the search results will be displayed. You may enter any form of a URL for the Web page you will have the search result displayed to. You can enter any Web page that has the searchdisplay tag inserted in it. If you want to use the same page, you can enter "#" into the field.

NOTE Target page **MUST** have the searchdisplay function inserted in it for the search results to be displayed.

Text Box

This variable will set the size of the text box that is displayed. This can be any number depending on how big you would like the search text box to be.

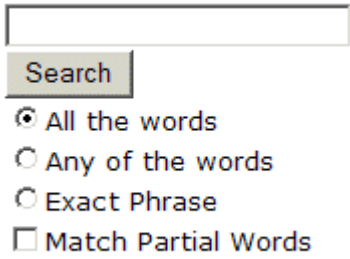
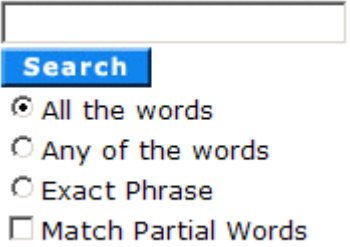
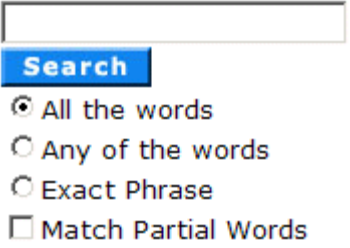
Text Box Size	Web Page View
10	 <p>The screenshot shows a search interface with a small text box, a 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.</p>
50	 <p>The screenshot shows a search interface with a larger text box, a 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.</p>

Maximum Characters

The value in this attribute determines the amount of characters

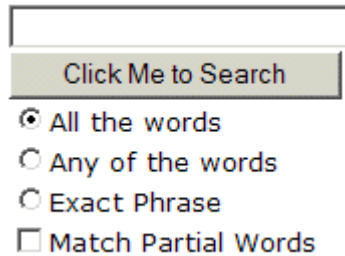
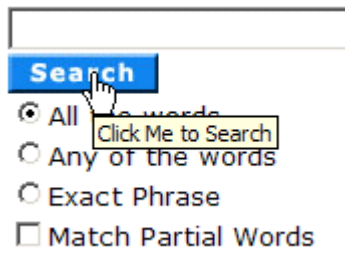
Button Image Source

If you would like to use an image for the search button instead of the default, you may enter the filename and path of the image.

Button Image Source	Web Page View
""	 <p>A screenshot of a web page showing a search form. It includes a text input field, a button labeled "Search", and four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words".</p>
"search.jpg"	 <p>A screenshot of a web page showing a search form. The button is a custom image labeled "Search". The radio button options are the same as in the first screenshot.</p>
"http://www.ektron.com/images/search.jpg"	 <p>A screenshot of a web page showing a search form. The button is a custom image labeled "Search" loaded from a URL. The radio button options are the same as in the previous screenshots.</p>

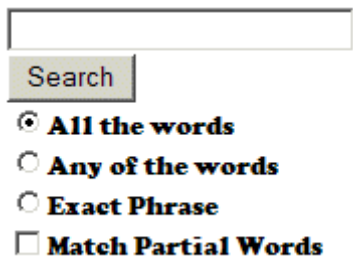
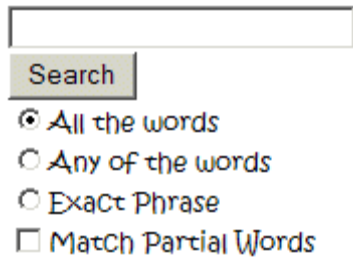
Button Text

If no search button image was specified, then the text that is entered in this field will be the text that appears on the button. If an image is being used for the search, then this text will be the alt text for the button.

Button Text	Web Page View
"Click Me to Search"	 <p>The screenshot shows a search interface. At the top is a text input field. Below it is a button with the text "Click Me to Search". Under the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words" (which is a checkbox).</p>
"Click Me to Search"	 <p>The screenshot shows the same search interface as above, but the button is blue and labeled "Search". A mouse cursor is clicking the button, and a tooltip with the text "Click Me to Search" is visible over the button. The radio button options are the same as in the first screenshot.</p>

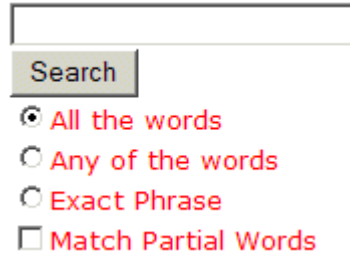
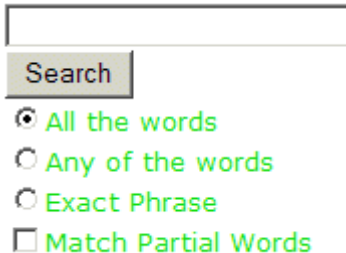
Font Face

This is the font that is used for the search options that are displayed on the Web page.

Font Face	Web Page View
Cooper Black	 <p>A screenshot of a web page showing search options rendered in the Cooper Black font. It includes a search input field, a 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox).</p>
Kristen ITC	 <p>A screenshot of a web page showing search options rendered in the Kristen ITC font. It includes a search input field, a 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox).</p>

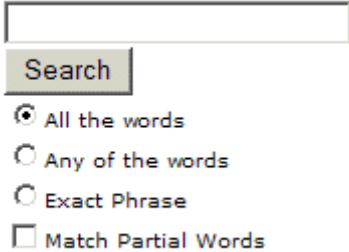
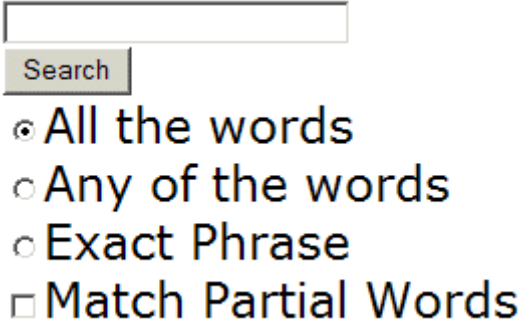
Font Color

This is the color of the text in the search tag. You may enter the color name or the hexadecimal value of the color.

Font Color	Web Page View
Red	 <p>The screenshot shows a search interface 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'. All text in the interface is colored red.</p>
#14DA14	 <p>The screenshot shows the same search interface as above, but with a green font color. The text in the input field, the 'Search' button, and the radio button options are all colored green.</p>

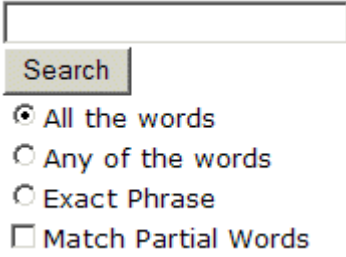

Font Size

This is the size of the text that is in the search tag.

Font Size	Web Page View
1	 <p>A search form 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' (checkbox).</p>
5	 <p>The same search form as above, but with a larger font size. The text is significantly larger and more prominent.</p>

Horizontal

This variable determines whether the search options are displayed vertically or horizontally.

Attribute	Web Page View
0	 A vertical layout of search options. It features a text input field at the top, followed by a 'Search' button. Below the button are four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.
1	 A horizontal layout of search options. It features a text input field and a 'Search' button on the left, followed by four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.

Spare 1

At the end of the Search tag, there is a variable called Spare 1. This variable has not yet been assigned an attribute. As of right now, when you enter a search tag, you must pass this variable ""

Search Display Function

The search display custom tag created the area where all the search results show after the search has been executed.

Shown here is the format of the ecmSearchDisplay tag:

```
<% ecmSearchDisplay( Max Number returned,
                    "StyleInfo",
                    Showdate
                ) %>
```

Below is a description of each of the values of the tag:

Attribute	Description	Required
Max Number of content blocks Returned	Maximum number of content blocks returned. (0=Unlimited)	Yes
Style Info	An HTML style string used within the HTML "hyperlink" generated by the search output. Can control font, color, size, etc...	No
ShowDate	Displays the last modified date of the content block. 0=No 1=Yes	No

Search Display Example

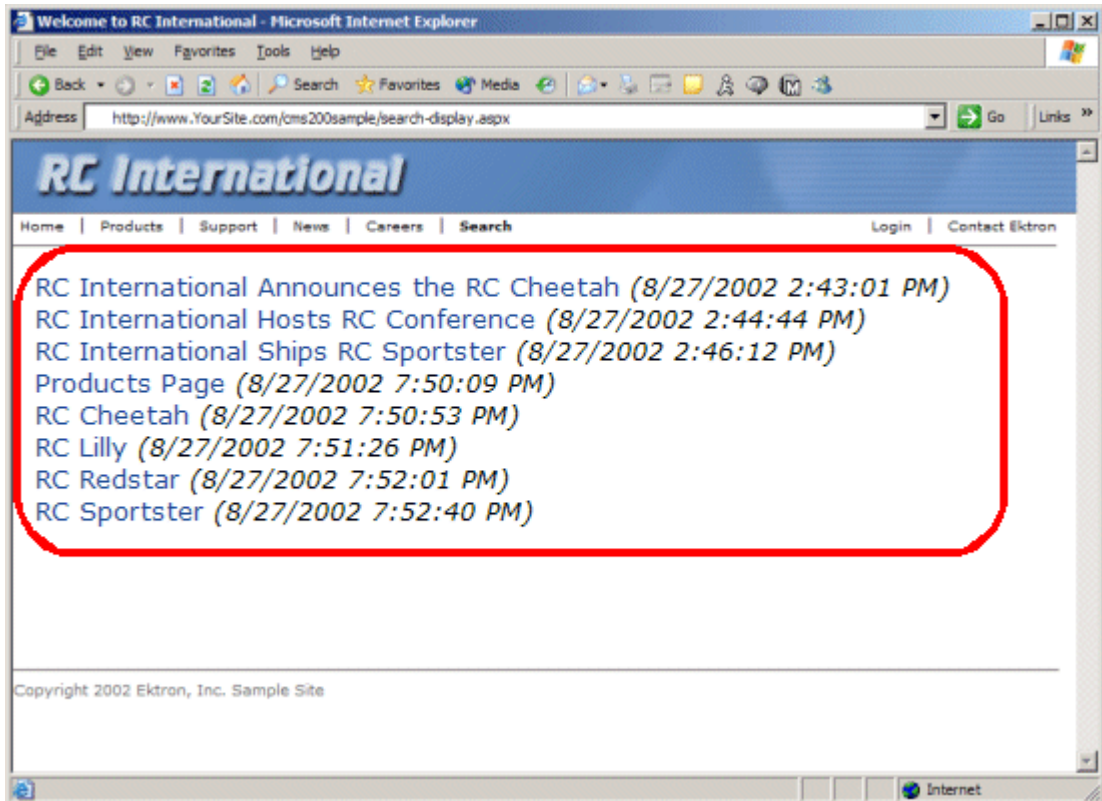
The following must be inserted on the page where you would like to display the search results. You may place it on the same page as the search tag, or insert it on a separate page.

```
<tr>
  <td>
    <% ecmSearchDisplay (0,"","1") %>
  </td>
</tr>
```

The previous search display example has the following properties:

- Returns unlimited results when the search is performed
- Contains no style information
- Shows the last modified date of the content

When inserted into a template, and a search has been performed, the results will appear similar to the following.



Search Display Variables

Listed below are all the variables for the SearchDisplay tag.

Max Number of Content Blocks Returned

This variable determines the amount of search results that will be displayed on the page. You may enter any integer into this

variable. If you would like to display an unlimited amount, enter 0 (zero).

Max Returns	Web Page View
0	<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>
3	<p>Search Results</p> <p>RC Sportster RC Redstar RC Lilly</p>

Style Info

An HTML style string used within the HTML “hyperlink” generated by the search output. Can control font, color, size, etc.

Style Info	Web Page View
""	<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>
"font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:50%;"	<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>

Show Date

Here, you can determine if you would like to include the Last Modified Date of the content block next to the result. This is useful

to determine which is the most recent content block in the results.
You have two choices for this:

Attribute	Web Page View
0	<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>
1	<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>

List Summary Function

The list summary function is used to display the list of content summaries on a Web page.

Shown here is the format of the list summary function:

```
<% ecmListSummary ( "StartingFolder",
    Recursive,
    ShowSummary,
    "StyleInfo",
    "OrderBy, Asc/Desc",
    "ShowInfo",
    MaxNumber,
    "Spare1",
    "Spare2"
) %>
```

A list and description of the variables are shown in this table:

Attribute	Description	Options
Folder	This is the folder the summary information will come from.	Any folder
Recursive	Allow the tag to get summary info from child folders.	0 = Not Recursive 1 = Recursive
ShowSummary	Do you want the content summary to be displayed?	0 = No 1 = Yes
StyleInfo	Inline style information for the generated text hyperlink.	Any string

Attribute	Description	Options
OrderBy, ASC/DESC	Order the hyperlinks by:	"Title" "DateModified" "DateCreated" "Startdate" "AuthorFname" "AuthorLname"
	Order the hyperlinks in ascending or descending order.	"Title,asc" "Title,desc"
ShowInfo	Information that will appear to the right of the hyperlink.	"DateModified" "DateCreated" "AuthorFname" "AuthorLname"
MaxNumber	Maximum number of summaries returned.	0 = Unlimited
Spare1	Not yet implemented	MUST BE ""
Spare2	Not yet implemented	MUST BE ""

List Summary Example

Shown below is an example of the ListSummary function in an ASP Web page.

```

<TR>
  <TD>
    <% ecmListSummary ("Marketing\News",0,1,"","Title,asc",
      "DateModified",0,"","") %>
  </TD>
</TR>

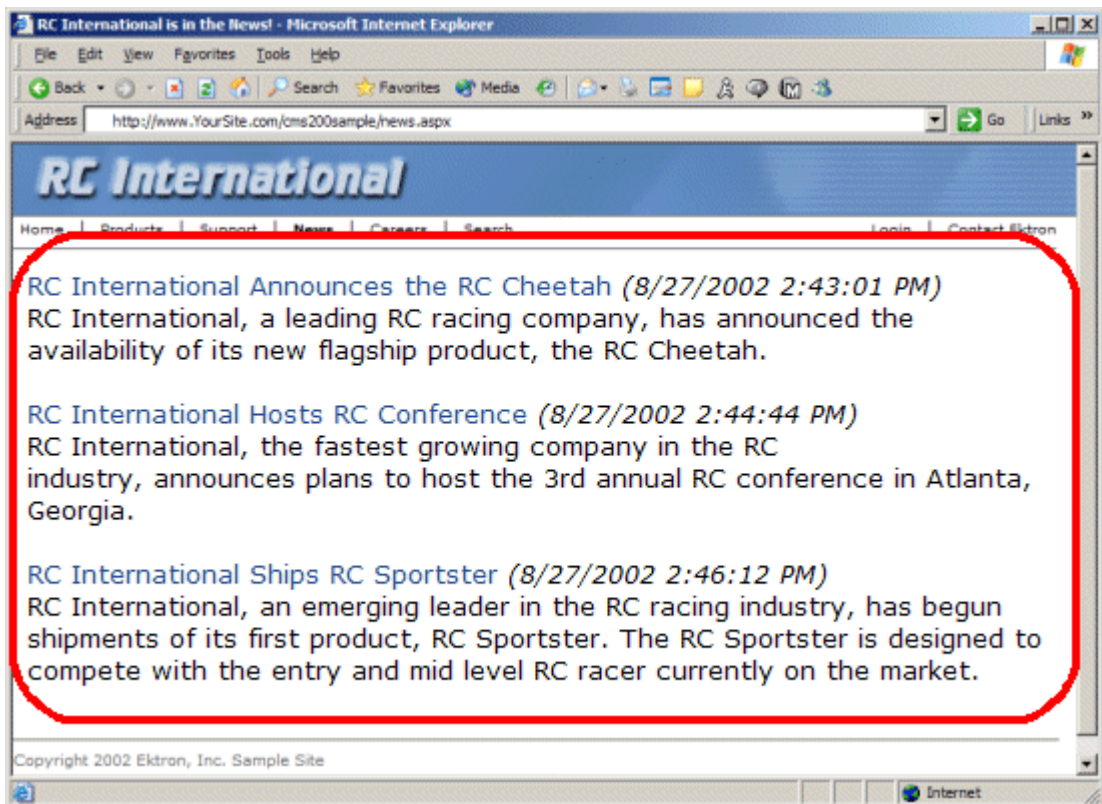
```

The previous example has the following properties:

- Displays summaries from the \Marketing\News folder
- Not recursive.
- The summaries will be shown
- No style was applied to the hyperlinks

- The results will be displayed in ascending order by title
- The date modified of each content block will be displayed
- Unlimited results will be shown.

Here is how the summaries will appear on the web page.



List Summary Variables

Listed below are all the different variables and descriptions about them.

Starting Folder

In this variable, you will need to specify which content folder you will want to get all the summaries from. There are two options for this variable:

Value	Description
""	By leaving the value blank, all the summaries from the root content folder will be displayed.
"\Folder Path"	You can specify a specific folder to display the summaries from by inserting the folder path

Recursive

When you setup the search to be recursive, you are allowing the search to search all the sub-folders associated with the starting folder you had specified. You have two options for this variable:

Value	Description
1	Recursive - summaries in the starting folder's sub-folders will also be displayed
0	Non-recursive - only summaries in the starting folder will be displayed.

Show Summary

The show summary attribute will allow or disallow the summary to be displayed in the summary list. You have two options for this variable:

Attribute	Web Page View
0	<p data-bbox="525 482 995 511">RC International Ships RC Sportster</p> <p data-bbox="525 546 1023 575">RC International Hosts RC Conference</p> <p data-bbox="525 609 1108 638">RC International Announces the RC Cheetah</p>
1	<p data-bbox="386 778 796 808">RC International Ships RC Sportster</p> <p data-bbox="386 808 1268 888">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.</p> <p data-bbox="386 919 821 948">RC International Hosts RC Conference</p> <p data-bbox="386 948 1258 1028">RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p data-bbox="386 1059 895 1088">RC International Announces the RC Cheetah</p> <p data-bbox="386 1088 1155 1139">RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Style Info

In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

Style Info	Web Page View
""	<p>RC International Ships RC Sportster 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.</p> <p>RC International Hosts RC Conference RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>
"font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:100%;"	<p>RC International Ships RC Sportster 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.</p> <p>RC International Hosts RC Conference RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Order By

This variable will sort the summary results by different options. Here are the different ways to sort the summaries:

Value	Summaries sorted by
Title	title
DateModified	date modified
Date Created	date created
StartDate	Start Date assigned to the content block when created.
AuthorFname	authors' first name
AuthorLname	authors' last name

Ascending/Descending

This variable is an addition to the previous OrderBy variable. By default, the list of the summaries are descending, meaning they will follow the path of A-B-C, or 1-2-3. When set to ascending, they will order by Z-Y-X or 3-2-1. You will have the option to re arrange these to display either in ascending or descending order.

To do this, add the following to the OrderBy attribute:

Value	Description
asc, 1	This will order the Summaries in an ascending order
desc, 0	This will order the Summaries in a descending order. This is the default setting.

By using the ascending/descending option, you will have the option to set the order of content blocks from most recent to oldest, or vice versa. As well as ordering the list by the user's first names or last names in ascending or descending order.

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are four different options that you may choose from:

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

Max Number

In this variable, you will determine how many summaries will be displayed on the page. You may enter any number into this field,

and that number will be the amount of summaries that will be displayed. If you want all the summaries to be displayed, enter 0 (zero) and you will display unlimited results.

Max Number	Web Page View
0	<p>RC International Ships RC Sportster 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.</p> <p>RC International Hosts RC Conference RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>
1	<p>RC International Ships RC Sportster 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.</p>

Spare 1 & 2

There are two variable fields at the end that are currently not used. They are there for future scalability. When you insert a ListSummary tag into your template, you must leave those variables as ""

Single Summary Function

The single summary tag is used to display the summary of one content block in the Ektron CMS200/300 Web site, instead of a list of all the summaries in a content folder. Seen below is the single summary tag:

```
<% ecmSingleSummary ( "ContentID",
    ShowSummary,
    "StyleInfo",
    "ShowInfo",
    "Spare1"
) %>
```

You can look at the single summary tag as just a simpler form of the List summary tag. The format for each of them is almost identical, but the single summary tag does not require as many variables to be passed.

A list and description of the variables are shown in this table:

Attribute	Description	Required?
Content ID	This is the ID number of the content block the summary information will come from.	Yes
ShowSummary	Do you want the content summary to be displayed?	Yes
StyleInfo	Inline style information for the generated text hyperlink.	No
ShowInfo	Information that will appear to the right of the hyperlink.	No
Spare1	Not yet implemented	MUST BE ""

Single Summary Example

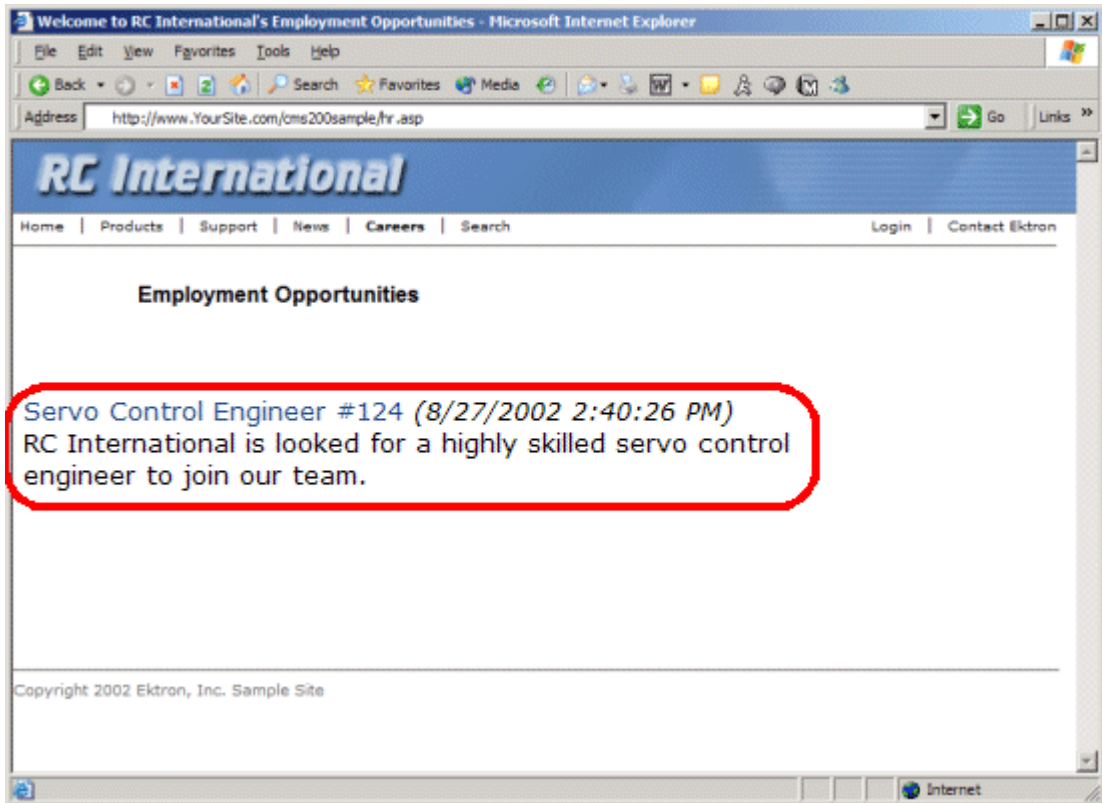
Shown below is an example of the single summary function in an ASP Web page.

```
<TR>
  <TD>
    <% ecmSingleSummary ("14",1,"","DateModified","") %>
  </TD>
</TR>
```

The previous example had the following properties:

- Displays the summary for the content block with an ID=14
- The summary will be shown
- No style was applied to the hyperlinks
- The last date the content was modified will be displayed

Here is how the summary will appear on the web page.



Single Summary Variables

Listed below are all the different variables and descriptions about them.

Content ID

In this variable, you will need to specify which content block you will want the summary to be displayed from.

Show Summary

The show summary attribute will allow or disallow the summary to be displayed in the summary list. You have two options for this variable:

Value	Web Page View
0	Servo Control Engineer #124
1	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.

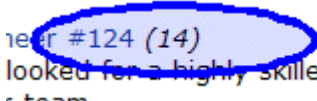
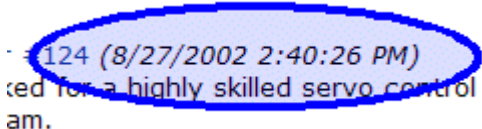
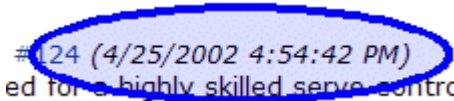
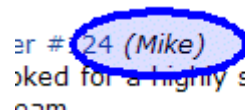
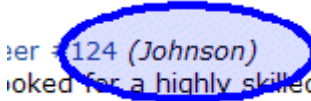
Style Info

In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

Style Info	Web Page View
""	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.
font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:100%;	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are five different options that you may choose from:

Value	Displays the...	Web Page View
ID	ID number for the content block.	
DateModified	date the content was last modified.	
DateCreated	date when the content was first created.	
AuthorFname	first name of the user who last edited the content block.	
AuthorLname	last name of the user who last edited the content block	

Spare 1

There is one variable fields at the end that are currently not used. This is here for future scalability. When you insert a single summary tag into your template, you must leave this variables as ""

Collection Function

The collection function is used to display the list of links that were created as a collection in the Ektron CMS200/300 Workarea. The ecmCollection function is highly customizable, allowing you to easily define how the collection data will be displayed on the Web page.

Shown here is the format of the ecmCollection function.

```
<% ecmCollection( "id",  
                "displayFunction" )  
%>
```

Each of the attributes in the ecmCollection function are explained in the following table.

Attribute	Description
id	The id of the collection that you would like to be displayed is defined here.
displayFunction	Specify a function that is defined in the API that will define how the collection data is displayed on the Web page. The display function must also be defined in the ecmCollection function.

Collection Examples

The following are examples of the collection function in the Ektron CMS200/300 sample Web site. One example uses the "ecmNavigation" display function and the other uses the "ecmTeaser" display function.

The ecmNavigation and ecmTeaser display functions are sample display functions included with Ektron CMS200/300. For information about creating your own custom display functions, see ["Creating Custom Display Functions" on page 207](#).

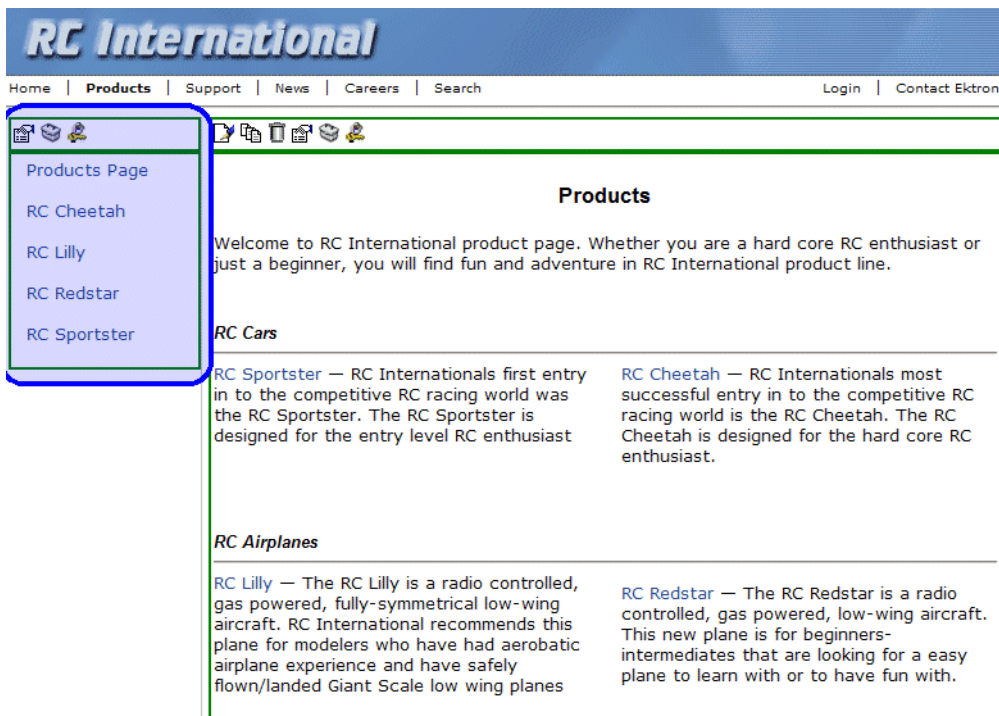
Example Using the ecmNavigation Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS200/300 sample Web site.

```
<% ecmCollection (1, "ecmNavigation") %>
```

In this example, the collection with an ID=1 is displayed, and the function “ecmNavigation” is used to define how the collection data is displayed on the Web page.

Shown here is how the collection would appear on the Web page.



As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used.

Shown below is the source code for the display function "ecmNavigation"

```
Function ecmNavigation(cInfo)
    dim html, info
    html = "<table border=""0"" cellspacing=""0"" cellpadding=""0""
width=""100%"">"
    for each info in cInfo
        html = html & "<tr><td>&nbsp;&nbsp;&nbsp;<a href=""
        html = html & info("ContentLinks")
        html = html & "">"
        html = html & Cstr(info("ContentTitle"))
        html = html & "</a></td></tr><tr><td>&nbsp;&nbsp;&nbsp;</td></tr>"
    next
    html = html & "</table>"
    ecmNavigation = html
end Function
```

As you can see, the ecmNavigation function is a simple ASP function that creates a table of the links that belong to the collection specified.

The ecmNavigation function also displays the title of the content blocks by using the "info("ContentTitle")) building block. See ["Creating Custom Display Functions" on page 207](#) for more information.

Example Using the ecmTeaser Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS200/300 sample Web site.

```
<% ecmCollection (2, "ecmTeaser") %>
```

In this example, the collection with an ID=2 is displayed, and the function "ecmTeaser" is used to define how the collection data is displayed on the Web page.

Shown here is how the collection would appear on the Web page.



As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used.

Shown below is the source code for the display function “ecmTeaser”

```
Function ecmTeaser(cInfo)
    dim html, info
    html = "<table border=""0"" cellspacing=""0"" cellpadding=""0""
width=""100%"">"
    for each info in cInfo
        html = html & "<tr><td><a href=""
html = html & CStr(info("ContentLinks"))
html = html & "" >"
html = html & CStr(info("ContentTitle"))
html = html & "</a>&nbsp;("
html = html & info("DateModified")
html = html & ")</td></tr><tr><td>"
html = html & CStr(info("ContentTeaser"))
html = html & "</td></tr><tr><td>&nbsp;</td></tr>"
    next
```

```
html = html & "</table>"  
ecmTeaser = html  
end Function
```

As you can see, the ecmTeaser function is a simple ASP function that creates a table of the links that belong to the collection specified.

The ecmTeaser function also displays the:

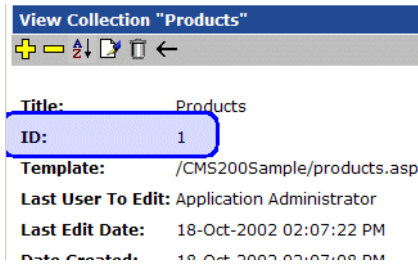
- Content Title
- Date Modified
- Content Teaser

See ["Creating Custom Display Functions" on page 207](#) for more information.

Collection Variables

ID

The collection ID is the ID number that is assigned to the collection when it is created in the workarea.



DisplayFunction

As explained earlier, the display function is a function that you create that will define how the collection data will be displayed on the Web page.

By using simple ASP scripting, you can create your own display functions, or use the three included functions.

The display function must be defined in two places.

- In the ecmCollection function as a case
- In the API file as a function.

Creating Custom Display Functions

The following section explains the basics behind creating custom display functions for your ecmCollection function.

Function Structure

As shown in the examples above, the display functions use a similar structure:

```
Function yourCollectionDisplayFunction(cInfo)
    dim html, info
    html = html & "<b>List of Content Titles:</b><br>"
    for each info in cInfo
        html = html & info("ContentTitle") & "<br>"
    next
    yourCollectionDisplayFunction = html
end Function
```

Creating a Basic Display Function

Explained below is a how to create a basic display function.

1. In the API file, begin by defining the display function you are creating	Function MyDisplayFunction(cInfo) end Function
2. Declare any necessary variables	Function MyDisplayFunction(cInfo) dim html, info end Function
3. Add any html or text that you would like to apply to the collection data as a whole, not individually (for example, add a opening and closing html table tag)	Function MyDisplayFunction(cInfo) dim html, info html = "" html = html & "" end Function

<p>4. Add a For loop that will be used to cycle through each of the collection items, and apply the information defined.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo next html = html & "" end Function</pre>
<p>5. Inside the for loop, create the string that will be applied to each of the items generated by the collection data.</p> <p>For information about the building blocks, see "Building Blocks" on page 212.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo next html = html & "" end Function</pre>
<p>This will place a at the beginning and the end of each of the items displayed.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" next html = html & "" end Function</pre>
<p>This will create the hyperlink for the collection item. When the collection item is generated on the Web page, the hyperlink will send the visitor to the corresponding content block.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" & Cstr(info("ContentTitle")) & "" html = html & "" next html = html & "" end Function</pre>

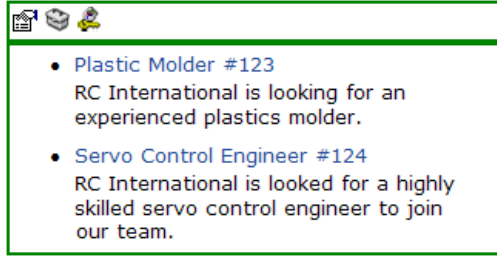

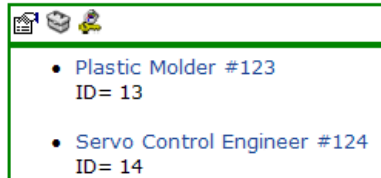
<p>4. Add a For loop that will be used to cycle through each of the collection items, and apply the information defined.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo next html = html & "" end Function</pre>
<p>5. Inside the for loop, create the string that will be applied to each of the items generated by the collection data.</p> <p>For information about the building blocks, see "Building Blocks" on page 212.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo next html = html & "" end Function</pre>
<p>This will place a at the beginning and the end of each of the items displayed.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" next html = html & "" end Function</pre>
<p>This will create the hyperlink for the collection item. When the collection item is generated on the Web page, the hyperlink will send the visitor to the corresponding content block.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" & Cstr(info("ContentTitle")) & "" html = html & "" next html = html & "" end Function</pre>

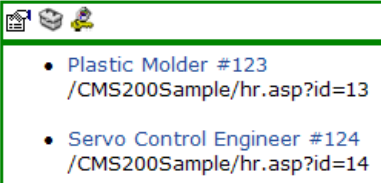



<p>4. Add a For loop that will be used to cycle through each of the collection items, and apply the information defined.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo next html = html & "" end Function</pre>
<p>5. Inside the for loop, create the string that will be applied to each of the items generated by the collection data.</p> <p>For information about the building blocks, see "Building Blocks" on page 212.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo next html = html & "" end Function</pre>
<p>This will place a at the beginning and the end of each of the items displayed.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" next html = html & "" end Function</pre>
<p>This will create the hyperlink for the collection item. When the collection item is generated on the Web page, the hyperlink will send the visitor to the corresponding content block.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" & Cstr(info("ContentTitle")) & "" html = html & "" next html = html & "" end Function</pre>





<p>This will display the summary for the content block that is displayed with the collection item.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" & Cstr(info("ContentTitle")) & "" html = html & Cstr(info("ContentTeaser")) html = html & "" next html = html & "" end Function</pre>
<p>6. Finally, add this line at the end of the function to return the HTML to be outputted to the ecmCollection function.</p>	<pre>Function MyDisplayFunction(cInfo) dim html, info html = "" for each info in cInfo html = html & "" html = html & "" & Cstr(info("ContentTitle")) & "" html = html & Cstr(info("ContentTeaser")) html = html & "" next html = html & "" MyDisplayFunction = html end Function</pre>

Building Blocks

The sample display function that we just created only uses two building blocks. This section explains each of the possible building blocks that you may use when creating a custom display function.

Name	Description	Web Page View
info("ContentTeaser")	Displays the summary that has been created for the content block that is displayed in the collection data.	 <ul style="list-style-type: none"> • 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.
info("Comment")	Displays the comment added to the content block.	 <ul style="list-style-type: none"> • Plastic Molder #123 This is the historical comment for Plastics Molder #123 • Servo Control Engineer #124 This is the historical comment for Servo Control Engineer #124
info("ContentID")	Displays the ID number of the content block that is assigned to it by Ektron CMS200/300.	 <ul style="list-style-type: none"> • Plastic Molder #123 ID= 13 • Servo Control Engineer #124 ID= 14

Name	Description	Web Page View
info("ContentLinks")	Inserts the path of the content block with respect to your site root.	 <ul style="list-style-type: none"> • Plastic Molder #123 /CMS200Sample/hr.asp?id=13 • Servo Control Engineer #124 /CMS200Sample/hr.asp?id=14
info("ContentTitle")	Displays the title of the content block that is displayed with the collection data.	 <ul style="list-style-type: none"> • Plastic Molder #123 /CMS200Sample/hr.asp?id=13 • Servo Control Engineer #124 /CMS200Sample/hr.asp?id=14
info("DateModified")	Displays the date and time the content block was last modified.	 <ul style="list-style-type: none"> • Plastic Molder #123 8/27/2002 2:38:34 PM • Servo Control Engineer #124 8/27/2002 2:40:26 PM
info("DisplayDateModified")	Displays the date and time the content block was last modified using the date/time format specified in Setup.asp.	 <ul style="list-style-type: none"> • Plastic Molder #123 27-Aug-2002 02:38:34 PM • Servo Control Engineer #124 27-Aug-2002 02:40:26 PM

Name	Description	Web Page View
info("DisplayEndDate")	Displays the end date and time for the content block using the date/time format specified in Setup.asp.	
info("DisplayGoLive")	Displays the start date and time for the content block using the date/time format specified in Setup.asp.	 <ul style="list-style-type: none"> • Plastic Molder #123 This content went live on: 30-Oct-2002 10:58 AM • Servo Control Engineer #124 This content went live on: 29-Oct-2002 06:51 PM
info("EditorFname")	Displays the first name of the user who last edited the content block.	 <ul style="list-style-type: none"> • Plastic Molder #123 Application • Servo Control Engineer #124 Application
info("EditorLname")	Displays the last name of the editor who last edited the content block.	 <ul style="list-style-type: none"> • Plastic Molder #123 Administrator • Servo Control Engineer #124 Administrator
info("EndDate")	Displays the end date and time for the content block	
info("GoLive")	Displays the start date and time for the content block.	 <ul style="list-style-type: none"> • Plastic Molder #123 This content went live on: 10/30/2002 10:58:28 AM • Servo Control Engineer #124 This content went live on: 10/29/2002 6:51:02 PM

Defining Display Functions

Once a display function has been created in the API, you need to define it in the ecmCollection function as a new case.

To define a display function as a new case:

1. Locate the ecmCollection function in the API file.
2. Find the line that contains "Select Case displayFunction"
3. Under that line, type in the following substituting "MyDisplayFunction" with the actual name of the function that you created:

```
Case "MyDisplayFunction"  
    html = MyDisplayFunction(cInfo)
```

4. Save and close the file

With the display function defined, and added to the ecmCollection function as a new case, you are ready to use the display function to define how collections look.

Using a Custom Display Function

To use the custom display function that you have created, simply add the ecmCollection function to a template, specify the ID of the collection that will be displayed, and add the display function that will define how the collection is displayed.

```
<% ecmCollection ("2", "MyDisplayFunction") %>
```

When displayed on the Web site, the collection will look like the following.



Random Content

The random content function is used to randomly display the content of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

```
<% ShowRandomContent ("ID", "XSLTId") %>
```

Where the “**id**” represents the ID number of the collection the content blocks, that will be displayed, belong to



And “**XSLTId**” is the ID number of the display XSLT that will be applied to the content blocks in the collections.

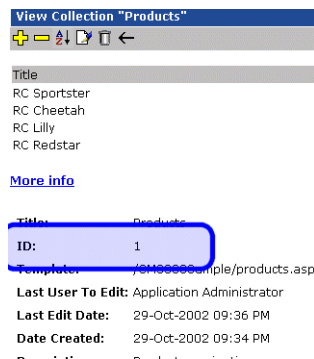
Random Summary

Similar to the random content function, the random summary function is used to randomly display the summary of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

<% ShowRandomSummary (id) %>

Where the “id” represents the ID number of the collection the content blocks, that will be displayed, belong to



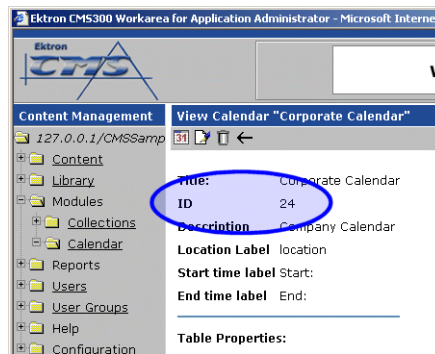
Event Calendar Function

The ecmEvtCalendar function is used to display an event calendar that has been created in the Ektron CMS200/300 Workarea.

Displayed here is the format for the event calendar custom function:

```
<% ecmEvtCalendar (id) %>
```

Where the “id” represents the ID number of the event calendar in the Workarea.



Event Calendar Style Sheet

Before you insert an event Calendar into your Ektron CMS200/300 template, it is recommended that you make a reference to the style sheet, “\CMS300Sample\ekCalendar.css” in your template.

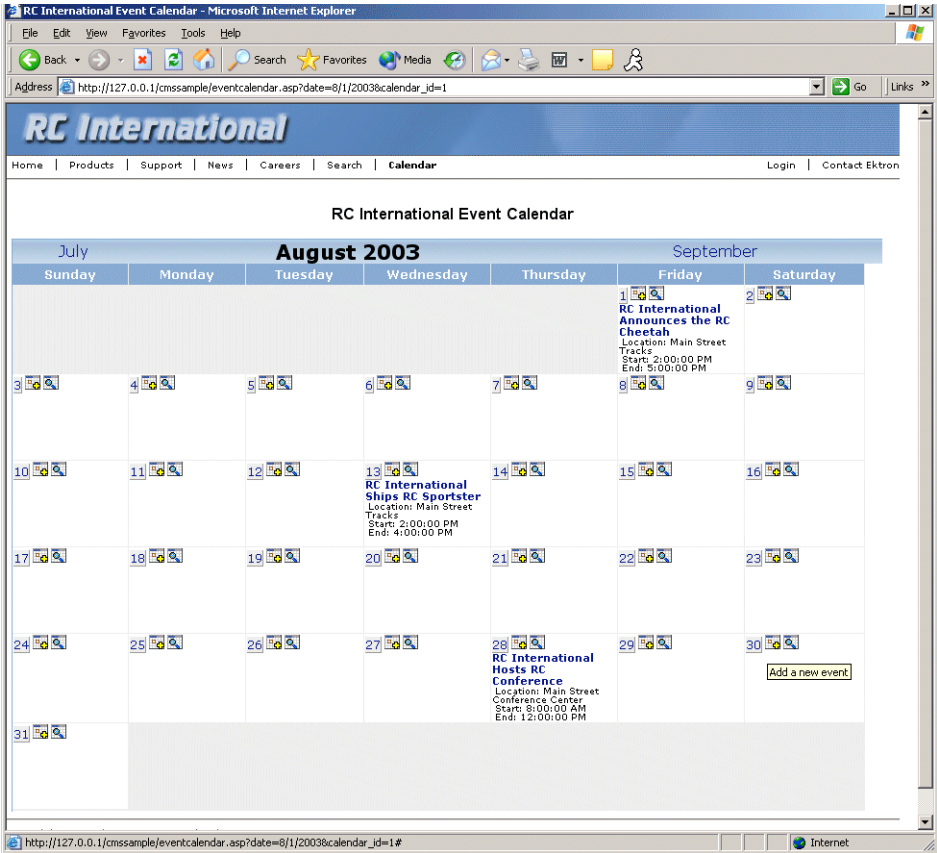
This style sheet contains styles that define the Calendar.

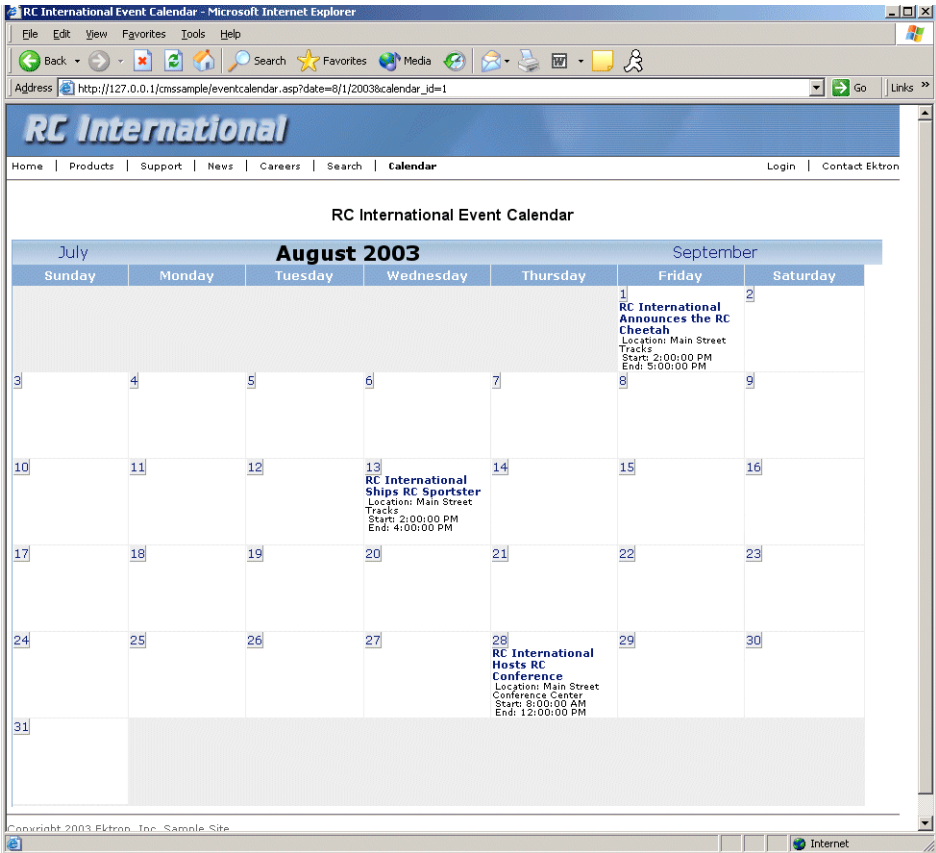
Event Calendar Sample

Displayed here is an example of the event Calendar in an Ektron CMS200/300 template:

```
<tr>
  <td>
    <% ecmEvtCalendar (1) %>
  </td>
</tr>
```

When this code is displayed in a browser, the Calendar with an ID-1 is displayed. Shown here is the Calendar displayed on the Web Page

If...	Calendar Display...
Logged in	

If...	Calendar Display...
Logged Out	

As you can see, when you are logged into the Web site, you are given options to add and view Calendar events. When logged out, only the calendar events are displayed.

Abbreviating Day Names

To further customize the event calendar function, you may specify in ApplicationAPI.aspx to display abbreviated day names (for example, Mon instead of Monday, Wed instead of Wednesday).

Displayed here is an example of the ecmEvtCalendar using abbreviated day names.

November		December 2003			January
Mon	Tue	Wed	Thu	Fri	
1	2	3	4	5	
8	9	10	11	12	

Configuring ApplicationAPI.aspx

To configure the ecmEvtCalendar function to display abbreviated day names, perform the following steps.

NOTE

Before editing ApplicationAPI.aspx, Ektron recommends to create a copy of it, referencing it in site_scripts_path.aspx, and making changes to the new file. See "Custom API" on page 235 for additional information.

1. Open the file that contains the ecmEvtCalendar function you want to edit.
2. Locate the ecmEvtCalendar function in the file.

```

536 Function ecmEvtCalendar(byVal cID)
537     Dim CalendarObj, Arg, CalView, tmpHolder
538     'check to see if we are logged in if we need some script
539     if (Not(InitExecuted)) then
540         InitEcm()
541     end if
542     if (PopUpLoaded = 0) then
543         Response.Write(ecmGetPopupScript())
544         PopUpLoaded = 1
545     end if
546     CalendarObj = CreateObject(MODULE_OBJ)
547     Arg = CreateObject("Scripting.Dictionary")
548     if((request.QueryString("ekM") <> "") And (request.QueryString("ekY") <> "")) then
549         Arg.Add ("Month", (request.QueryString("ekM")))
550         Arg.Add ("Year", (request.QueryString("ekY")))
551     else
552         Arg.Add ("Month", "")
553         Arg.Add ("Year", "")
554     end if
555     Arg.Add ("ImagePath", AppImgPath)
556     Arg.Add ("AppPath", AppPath)
557     Arg.Add ("CallerID", ecmCookie("user_id"))
558     Arg.Add ("Site", ecmCookie("site_id"))
559     Arg.Add ("CalendarID", cID)
560     Arg.Add ("Workspace", "")
561     tmpHolder = Request.ServerVariables("query_string")
562     Arg.Add ("QueryString", tmpHolder)
563     tmpHolder = Request.ServerVariables("URL")
564     Arg.Add ("URLfile", tmpHolder)
565     tmpHolder = Request.ServerVariables("path_info")
566     Arg.Add ("PathInfo", tmpHolder)
567     tmpHolder = Request.ServerVariables("Server_name")
568     Arg.Add ("ServerName", tmpHolder)
569     Arg.Add ("PreviewMode", cmsPreview)
570     Arg.Add ("AbbreviateDayNames", false)
571     CalView = CalendarObj.ecmCalendar(AppConfStr, Arg)
572     Response.write(CalView)
573     CalendarObj = nothing
574     Arg = nothing
575 End function

```

3. Locate the line that reads:

```

Arg.Add ("AbbreviateDayNames", false)

tmpHolder = Request.ServerVariables("Server_name")
Arg.Add ("ServerName", tmpHolder)
Arg.Add ("PreviewMode", cmsPreview)
Arg.Add ("AbbreviateDayNames", false)
CalView = CalendarObj.ecmCalendar(AppConfStr, Arg)
Response.write(CalView)
CalendarObj = nothing

```

4. On that line, change "false" to "true"
This will force the ecmEvtCalendar to use abbreviated day names.
5. Save and close the file.
6. View the Web page that contains the updated function.

Form Function

The ecmFormBlock function displays a content block associated with a form.

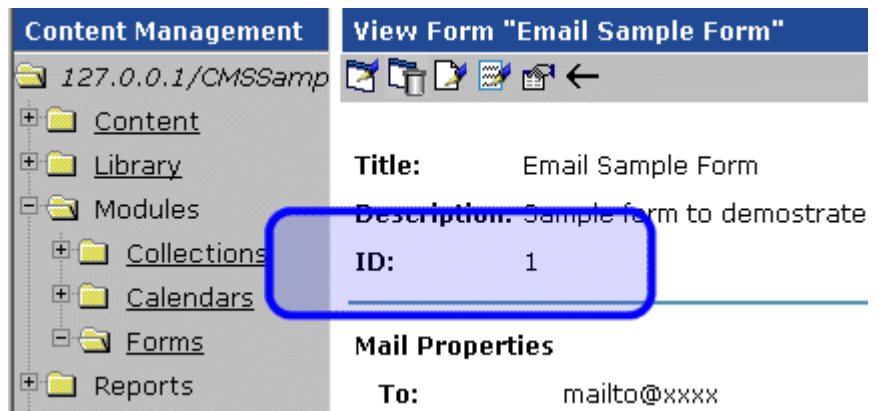
IMPORTANT!

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 CMS200/300 Administrator manual chapter "Managing Library Assets", section "Updating Default Template for Multiple Quicklinks."

Here is the format for the form content block custom function:

`<% ecmFormBlock (id) %>`

The "id" represents the ID number of the form in the Workarea.



NOTE

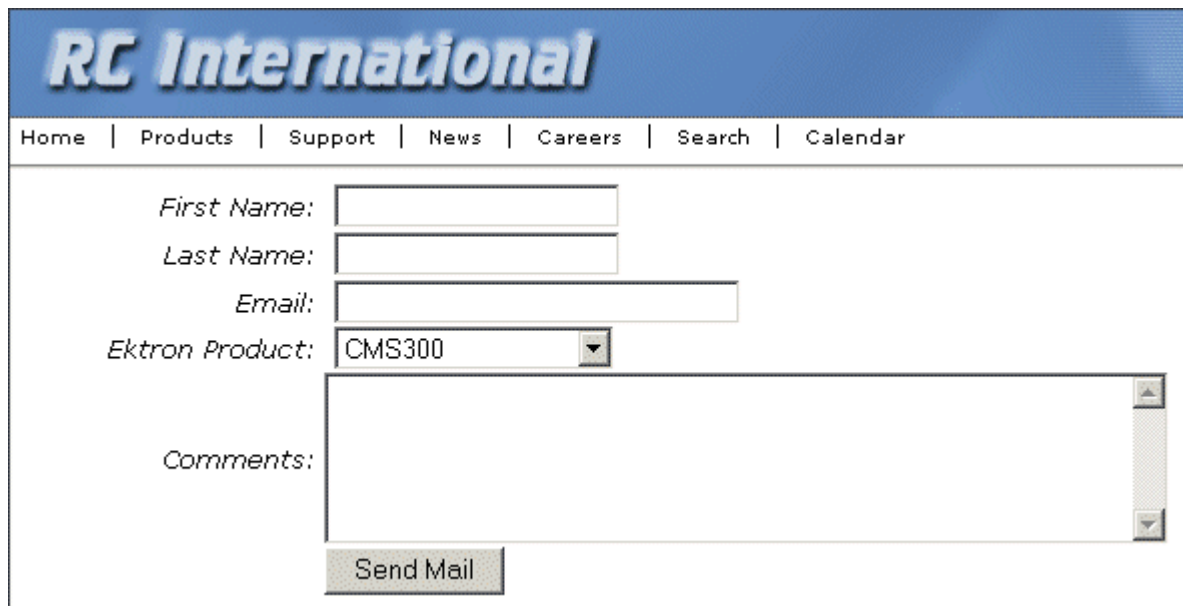
As an alternative, you can use the multipurpose function, which can display either a content block or a content block associated with a form. For more information, see "MultiPurpose Function" on page 227.

Form Sample

Displayed here is an example of the event Calendar in an Ektron CMS200/300 template:

```
<tr>
  <td>
    <% ecmFormBlock (1) %>
  </td>
</tr>
```

When this code is displayed in a browser, the following is displayed:

The screenshot shows a web page for "RC International". At the top is a blue header with the company name in a stylized font. Below the header is a navigation bar with links: Home, Products, Support, News, Careers, Search, and Calendar. The main content area contains a contact form. The form has labels for "First Name:", "Last Name:", "Email:", and "Ektron Product:". The "First Name" and "Last Name" fields are short text boxes. The "Email" field is a longer text box. The "Ektron Product" field is a dropdown menu with "CMS300" selected. Below these fields is a large text area for "Comments:". At the bottom of the form is a "Send Mail" button.

When a visitor submits the form, the form uses the information defined for the form module to process the information.

MultiPurpose Function

The ecmMultiPurpose function displays either a content block or an content block associated with a form. As a result, you can use one template to display both types of content block.

Here is the format for the Multipurpose function.

<% ecmMultiPurpose (ID) %>

To pass in a standard content block, pass the id value in the url parameter, as illustrated below.

`http://localhost/cms300sample/index.asp?id=2`

To pass in a Form content block, pass the ekfrm value in the url parameter, as illustrated below.

`http://localhost/cms300sample/index.asp?ekfrm=2`

NOTE For more information about passing the id value as a url parameter, see ["Dynamic Content Block" on page 162.](#)

If the id value or ekfrm value does not exist, a blank template appears.

If the url parameter is not passed in or recognized, the function calls ecmContentBlock and passes the DefaultID value as the id argument. If the DefaultID value is zero (0), ecmContentBlock is not called.

Show Site Language Function

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



(For more information, see the “Multi-Language Support “ chapter of the **Ektron CMS200/300** Administrator manual.)

Below is an example of the Show Site Language custom tag.

<%ecmShowSiteLanguage(FieldName)%>

You can place this function in any location of any page on your site. By default, it appears in the lower right corner of the home page.

The variable is described in this table.

Variable	Description
FieldName	<p>An optional argument. You can leave it blank and allow the drop-down list to be populated by the available languages for your Web site. For example:</p> <pre><%=ecmShowSiteLanguage('')%></pre> <p>Or you can pass in an argument such as:</p> <pre><%=ecmShowSiteLanguage('mysitelanguage')%></pre>

Metadata Function

The metadata tag is used to display the metadata for one or more content blocks in the Ektron CMS200/300 Web site. Inserting this custom tag will allow you to generate the list of metadata that is edited by the authors, and insert it in the source of the Web page. Shown below is an example of what the metadata custom tag looks like.:

```
<% ecmMetaData ( "ItemList", "Spare1" ) %>
```

A list and description of the variables are shown in this table:

Attribute	Description	Required?
ItemList	List of the content block ID numbers, and exclusions	No
Spare1	Not yet implemented	MUST BE ""

Creating an Item List

The ItemList for the ecmMetadata tag has the following format:

```
"[content block id; Exclusion list]...[content block id; Exclusion list]"
```

Attribute	Description	Required?
[(left bracket)	Delineates the beginning of a content block id/Exclusion List pair.	Yes
Content Block ID	A number that represents the content block you wish to retrieve the metadata from	Yes
; (semicolon)	Separator: Separates the content block id from the exclusion list	No (Yes if including an exclusion list)

Attribute	Description	Required?
Exclusion List	A comma delimited list of metadata you do not wish to be retrieved for this content block.	No
] (right bracket)	Delineates the end of a content block id/Exclusion List pair.	Yes

ecmMetadata Example

For this example, our administrator has defined the following metadata types for Ektron CMS200/300.

Name	Style	Type	Editable	Required	Remove Dups	Case Sensitive
Title	n/a	HTML	Yes	Yes	No	n/a
Keywords	Name	Meta	Yes	Yes	Yes	No
Content-Type	http-equiv	Meta	No	Yes	Yes	n/a

Our example Web page has two content blocks displayed. Content block 1, the company's home page introduction text, and content block 2, a sidebar containing the lead story for the week. The metadata entered by your content editors for these two content blocks are:

Content Block 1:

Title – "Welcome to Our Home Page"

Keywords – "Home; page; company; Welcome"

Content-Type -- "text/html; charset=iso-8859-1"

Content Block 2:

Title – "Announcing our New Product!"

Keywords – "Announce; company; product; widget"

Content-Type -- "text/html; charset=iso-8859-1"

To include both content blocks metadata in the web page our ecmMetadata call would look like:

```
<html>
  <head>
    <% ecmMetadata ("[1][2;Title,Content-Type]", "") %>
    (rest of head)
  </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

In the above example the ecmMetadata call is contained in the <head> of the HTML document (since all HTML metadata is required to exist there). Once the Web page is viewed the Web page source will contain the following information:

```
<html>
  <head>
    <Title>Welcome to Our Home Page</Title>
    <meta name="Keywords" content= Home; page; company; Welcome;
Announce;
    product; widget">
    <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
    (rest of head)
  </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

The ecmMetadata constructed these three HTML tags based on the administrator defined parameters and the ecmMetadata call.

The metadata tag was constructed in the page because:

- **[1]** - Will display all the metadata content for the content block ID=1. In this example, the title, keywords, and content-type were defined, so they were all displayed as:

```
<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome;
Announce;
product; widget">
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
```

- **[2;title,content-type]** - Will display all the metadata content for content block ID=2, EXCEPT for its title and content-type definitions.

```
<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome;
Announce;
product; widget">
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1">
```

The next section will provide you with a few more examples

ecmMetadata Variables

Listed below are all the different variables and descriptions about each of the them.

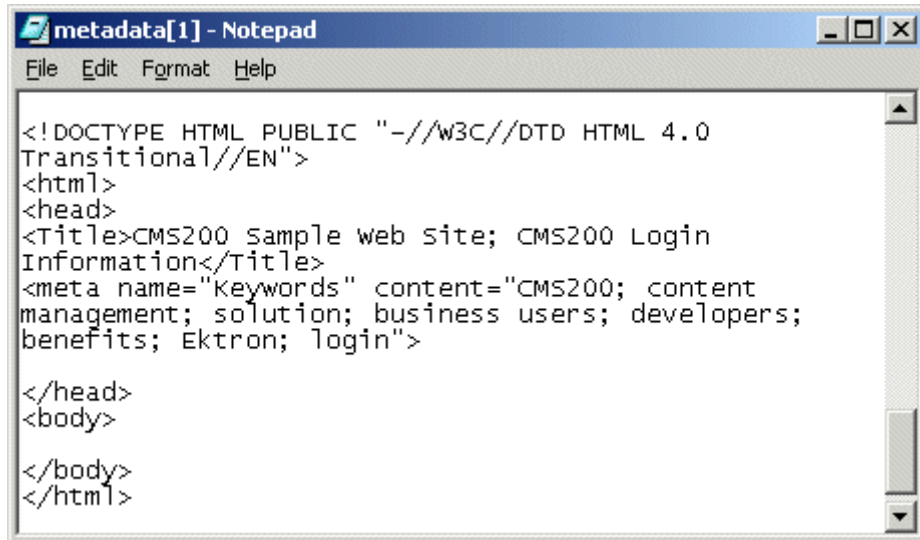
Item List

In this variable of the ecmMetadata tag, you will need to specify two values:

Value	Description
ID	Specify the content block ID to specify which content block will have the metadata displayed.
Exclusion List	After you specify the ID for the content block you want to display the metadata, you can also specify the metadata that you wish to be excluded.

Some examples might be:


```
<% ecmMetaData ("[1][2]", "") %>
```



```

metadata[1] - Notepad
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Sample web site; CMS200 Login
Information</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits; Ektron; login">

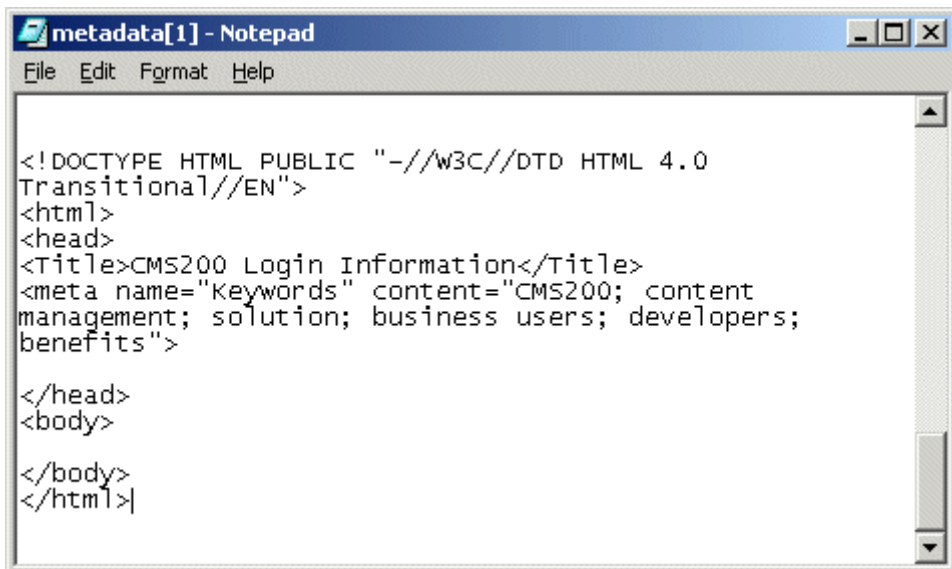
</head>
<body>

</body>
</html>

```

In this example, all the metadata from the content block ID=1 and 2 will be displayed.

```
<% ecmMetaData ("[1;title][2;keywords]", "") %>
```



```

metadata[1] - Notepad
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Login Information</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

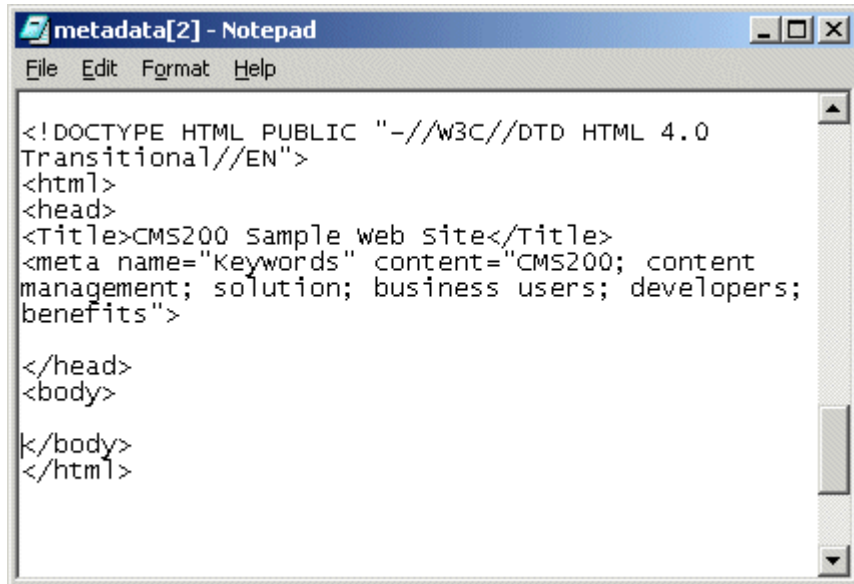
</head>
<body>

</body>
</html>

```

In this example, all the metadata from content block ID=1 will be displayed EXCEPT for the title. Also, all the metadata for content block ID=2 will be displayed Except for its keywords.

```
<% ecmMetaData ("[1][2;title,keywords]","") %>
```



In this example, all the metadata from content block ID=1 will be displayed, and all the metadata from content block ID=2 will be displayed except for its title and keywords.

Spare 1

There is one variable fields at the end that are currently not used. This is here for future scalability. When you insert a single metadata tag into your template, you must leave this variables as ""

Custom API

As you have seen, the formatting for the custom functions used in Ektron CMS200/300 for ASP.Net are defined in the file `ApplicationAPI.aspx`. As a developer, you may make changes to this file to further enhance how the custom functions appear on your Web site.

However, Ektron recommends creating a copy of the file, and renaming it to an arbitrary name, for example, `MyCustomAPI.aspx`, and including that file when creating Ektron CMS200/300 templates. This ensures that your custom API will not be overwritten if you upgrade to another version of Ektron CMS200/300.

Custom ColdFusion Functions

Ektron CMS200/300 maintains and display content through the use of custom Ektron ColdFusion functions and ComObjects.



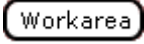
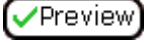

To customize you Ektron CMS200/300 application, you will need to be familiar with these tags

Tag	Description	More Information
Collection	Displays the links generated by collections.	"Collection Function" on page 280
Content Block	Displays and manages a content block.	"Content Block Function" on page 240
Event Calendar	Displays the calendar defined in the workarea.	"Event Calendar Function" on page 288
Form	Displays an HTML form content block.	"Form Function" on page 292
List Summary	Displays the list of the content block summaries.	"List Summary Function" on page 266
Login	Paints the login or logout button which allows the user to login and use the Ektron CMS200/300 application.	"Login Function" on page 238
Metadata	Displays the metadata in the source of the Web page the content is displayed on.	"Metadata Function" on page 296
MultiPurpose	Can display either a content block or a form	"MultiPurpose Function" on page 294
Random Content	Randomly displays the body content of a collection item.	"Random Content" on page 248
Random Summary	Randomly displays the summary content of a collection item.	"Random Summary" on page 249
Search	Inserts a text box on the template with a search button.	"Search Function" on page 250

Tag	Description	More Information
Search Display	Creates area where search results will be displayed.	"Search Display Function" on page 261
Single Summary	Displays the summary for a single content block.	"Single Summary Function" on page 275
Show Site Language Function	Lets user select language in which to view Web site.	"Show Site Language Function" on page 295

Login Function

This tag is responsible for displaying the following buttons on the Web page.

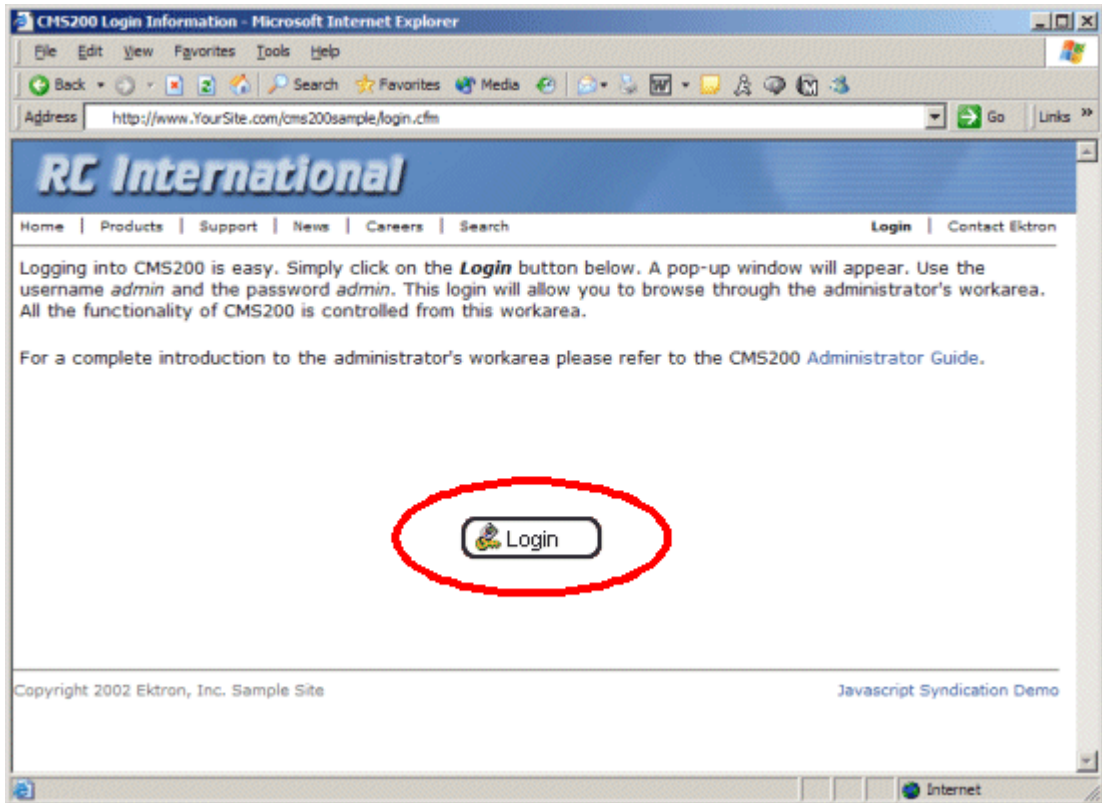
Button	Name	Description
	Login	When user is not logged into Ektron CMS200/300, this button will be displayed. Clicking this button will allow the user to log into the application.
	Logout	Once logged in, this button will replace the login button to allow the user to log out from the Ektron CMS200/300 Web site.
	Workarea	When logged in, this button will be displayed under the logout button allowing the user to access their workarea.
	Preview On	This button will allow the user to preview the entire Web site as if all checked-in content was published.
	Preview Off	Turns off the site preview mode.

Login Tag Example

The following example places a login button in a table cell inside a ColdFusion template.

```
<tr>
  <td>
    <cfmodule template="#request.ecm.AppPath#ecmLogin.cfm">
  </td>
</tr>
```

When inserted properly, a login button will be placed on the Web page.



Clicking on the login tag will open the login window where a user will be able to enter their username and password. Upon successful completion, the user will then be logged into the Ektron CMS200/300 Web site.

Placing Your Login Button

You are able to add as many login buttons to a template as you would like. You might want to include a login button on each template, or have a special Web page called login.cfm where you can log in there without the public being able to access the login page.

Content Block Function

The content block tag is used to display a content block on an Ektron CMS200/300 Web page.

Shown here is the format for the content block function:

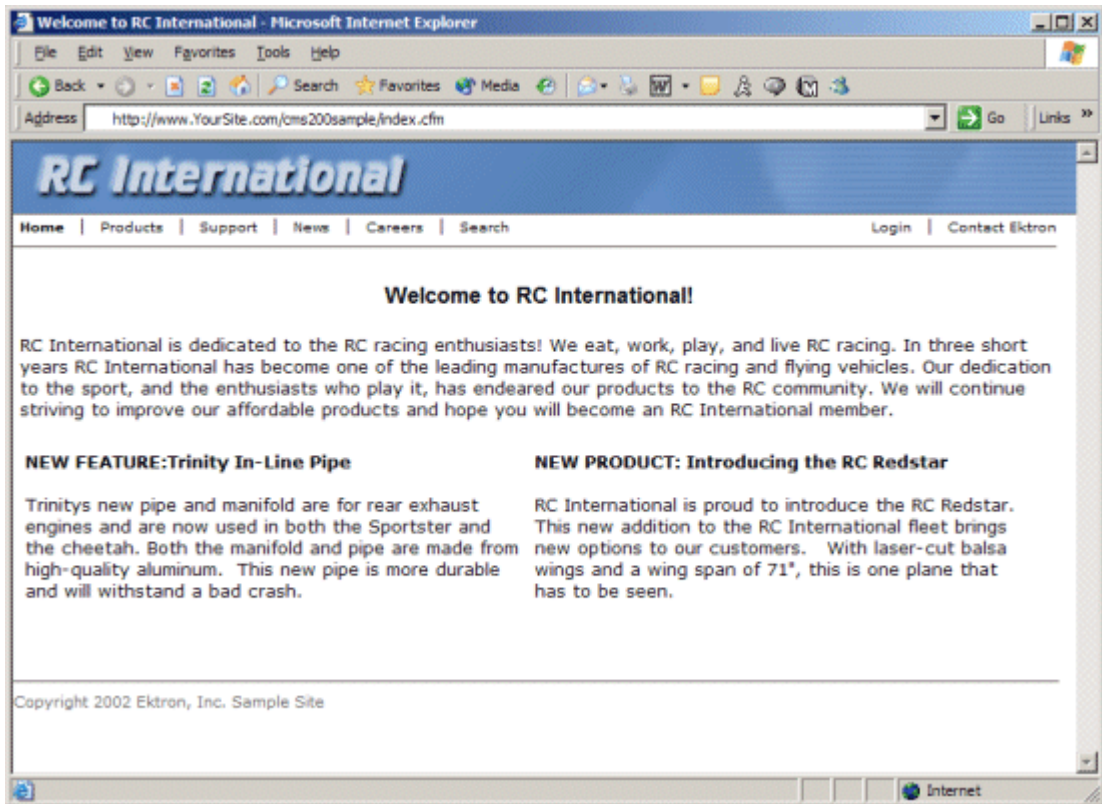
```
<CFMODULE template="#request.ecm.AppPath#ecmContentBlockEx.cfm"
           ID=" "
           XSLT=" "
           Override=" "
>
```

NOTE As an alternative, you can use the [multipurpose function](#), which can display either a content block or a content block associated with a form. For more information, see ["MultiPurpose Function" on page 294](#).

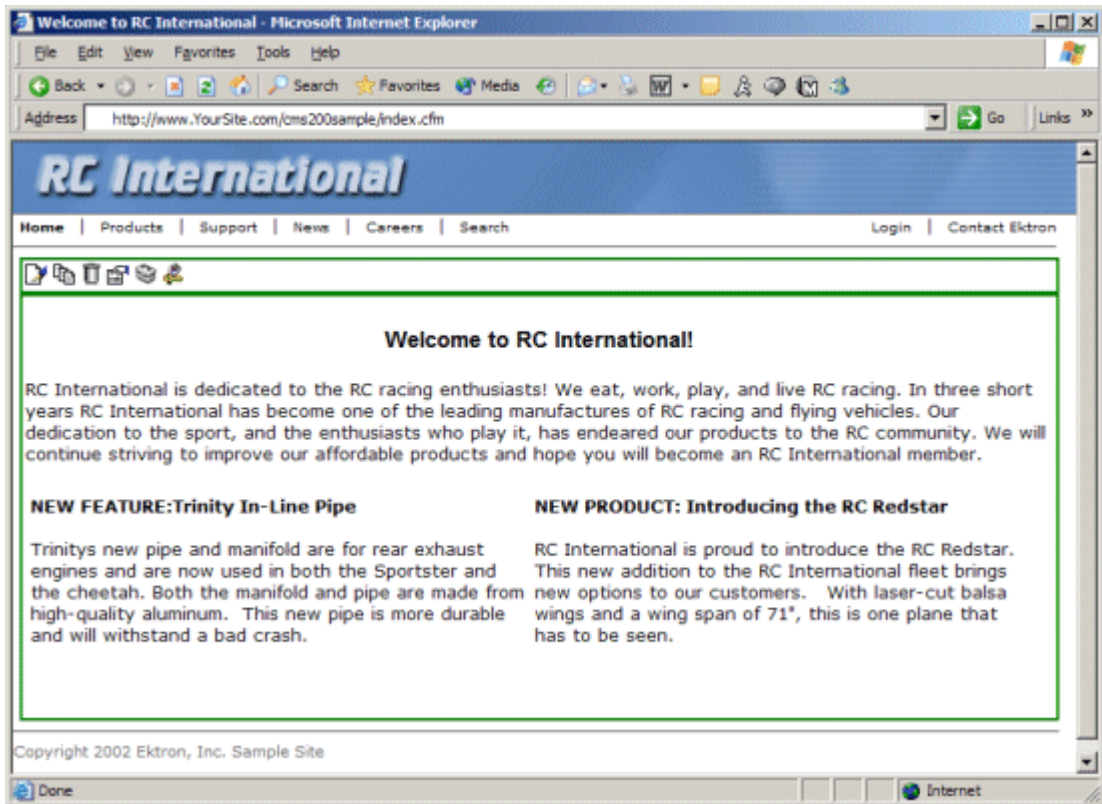
The table below explains the attributes of the content block function.

Attribute	Description	Required
ID	ID number assigned to a content block when created in Ektron CMS200/300.	Yes
XSLT	File name and path to a valid XSLT to transform the XML content.	No
Override	If no XSLT is specified, then enter 1 to use XSLT1, 2 for XSLT2, or 3 for XSLT3.	No

In normal view, this tag retrieves the content block from the database and returns it to the calling template.



If a user is logged into Ektron CMS200/300, the content block tag paints a border around the content block and provides the menu for managing that piece of content.



Content Block Examples

Ektron CMS200/300 has three types of content blocks.

Type	Used to display	For more information, see
Static	One content block on a Web page	"Static Content Block" on page 243
Dynamic	Content block whose ID is passed through the URL	"Dynamic Content Block" on page 244

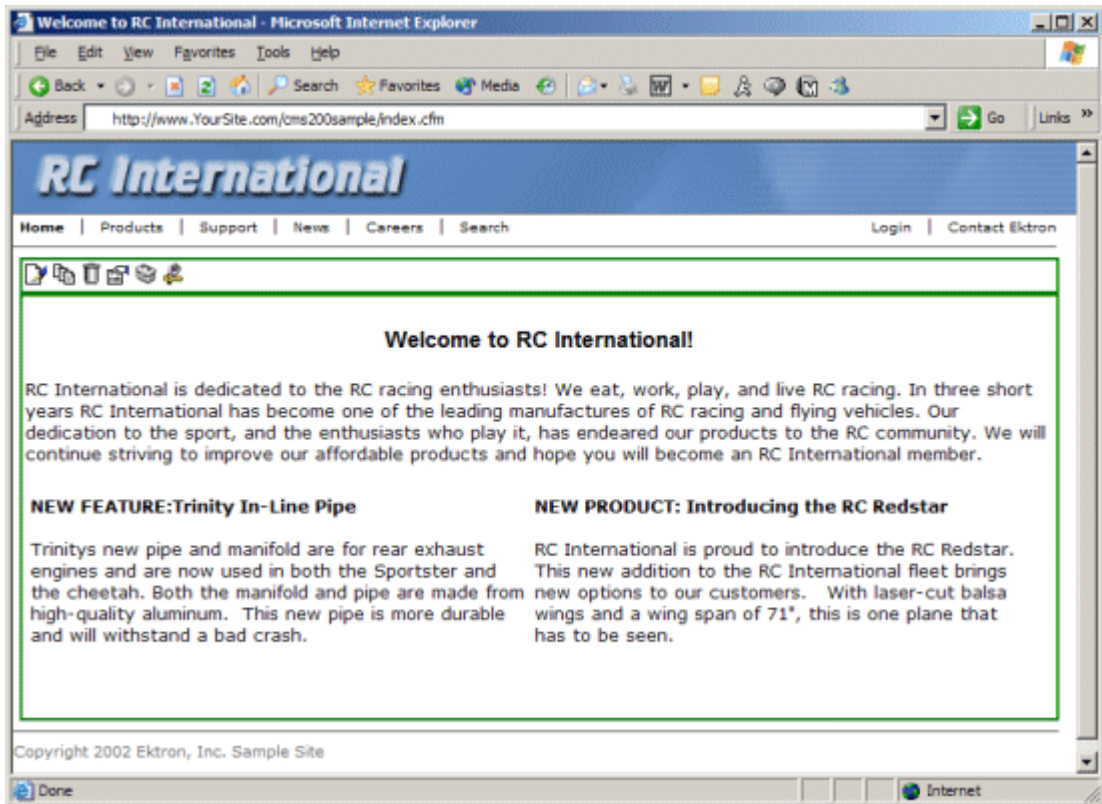
Type	Used to display	For more information, see
XML	XML content transformed by an XSLT specified in the function <hr/> Note: Not available in CMS200. <hr/>	"XML Content Block" on page 245

Static Content Block

The following example explains how to insert a static content block on a template.

```
<tr>
  <td>
    <CFMODULE id="1" template="#request.ecm.AppPath#ecmContentBlock.cfm">
  </td>
</tr>
```

In the example above, the function calls the content block with an id of 1 from the database to the browser.

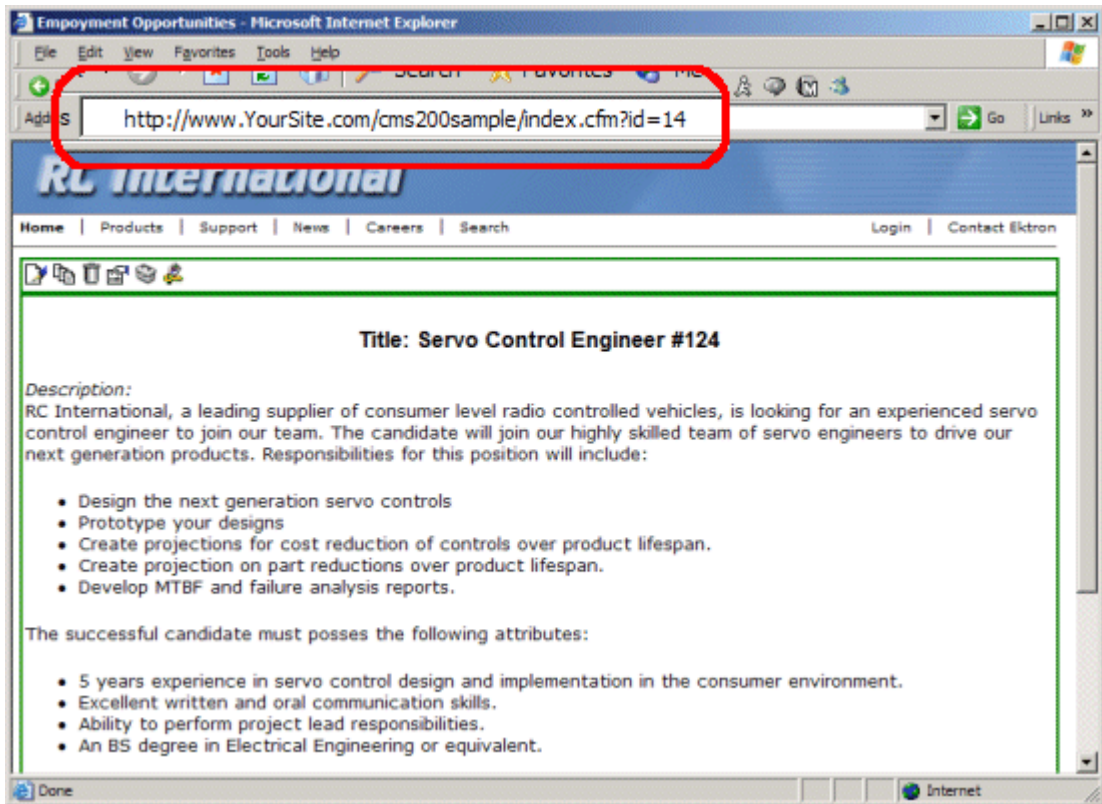


Dynamic Content Block

The following example shows how the content block tag can be used where the id is being passed in as a URL parameter. This tag would be used on a dynamic template.

```
<tr>
<td>
    <CFMODULE id="#id#" template="#request.ecm.AppPath#ecmContentBlock.cfm">
</td>
</tr>
```

When a user accesses this page and adds the ID number to the template this tag lives on, the content block with the corresponding ID is displayed on the Web page using the specified template.



XML Content Block

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

Here is an example of displaying an XML content block. Note that you specify both a content block and an XSLT that determines how the content block is displayed.

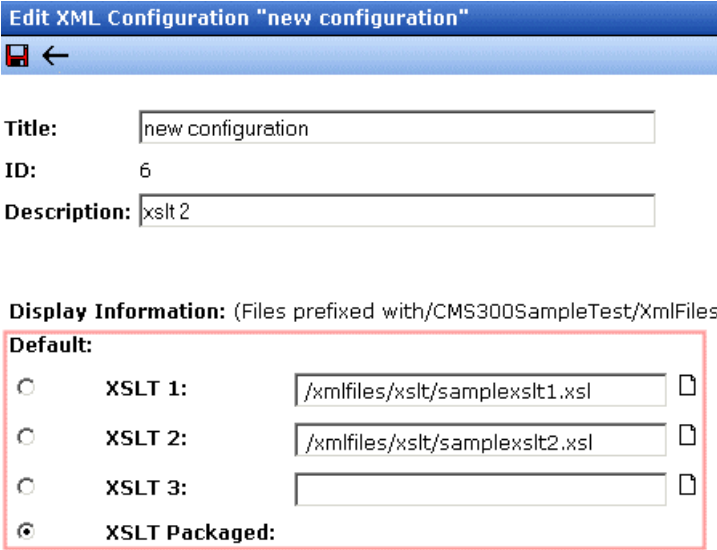
```
<tr>
  <td>
    <CFMODULE id="13" XSLT= "" override= "1"
      template="#request.ecm.AppPath#ecmContentBlockEx.cfm">
    </td>
</tr>
```

This example would display the content block ID=13, and use the XSLT1 as the display XSLT.

The Function's Arguments

The `ecmContentBlockEx` function takes three arguments. The first argument identifies the content block to display. The second one (`XSLT`) is optional and, if used, specifies an external XSLT file.

If the second argument does not exist, the third argument (`override`) specifies an XSLT identified in the Edit XML Configuration screen. The following table provides additional detail about the second and third arguments.

To use this display XSLT	Enter this for argument XSLT	Enter this for argument override
<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> 	""	none
XSLT Packaged: from the Edit XML configuration screen (that is, the XSLT created in the Data Designer)	""	0
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 for argument XSLT	Enter this for argument override
An absolute or relative path to an XSLT file.	An external XSLT file not specified in the Edit XML configuration screen. For example <% ecmContentBlockEx 13, "samplexslt.xsl"%>	does not matter - if a value exists in argument 2, argument 3 is ignored

See Also: The “Managing XML” chapter of the **Ektron CMS200/300** Administrator manual

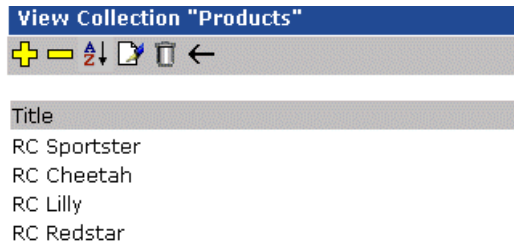
Override = 0 forces the XML content to use the default XSLT.

Random Content

The random content function is used to randomly display the content of a content block that belongs in a specified collection. Below is the format for the random content function:

```
<CFMODULE template="#request.ecm.AppPath#ecmShowRandomContent.cfm"
           ID=" "
           XSLT=" "
           Override=" "
>
```

ID represents the ID number of the collection to which the content blocks belong



[More info](#)

Title	Products
ID:	1
Template:	/CMS0000/sample/products.asp
Last User To Edit:	Application Administrator
Last Edit Date:	29-Oct-2002 09:36 PM
Date Created:	29-Oct-2002 09:34 PM
...	...

XSLT is the ID number of the display XSLT applied to the content blocks in the collection.

Random Summary

Similar to the random content function, the random summary function can randomly display a content block summary that belongs in a specified collection. Here is the format for the random content function

```
<CFMODULE template="#request.ecm.AppPath#ecmShowRandomSummary.cfm"
           ID=" "
>
```

Where the "id" represents the ID number of the collection the content blocks, that will be displayed, belong to

View Collection "Products"

Title
RC Sportster
RC Cheetah
RC Lilly
RC Redstar

[More info](#)

Title:	Products
ID:	1
Template:	/CMS200Sample/products.asp
Last User To Edit:	Application Administrator
Last Edit Date:	29-Oct-2002 09:36 PM
Date Created:	29-Oct-2002 09:34 PM
Product ID:	1

Search Function

The search tag is used to display a text box, with a search button on the template. When seen on the template on the Web page, the users will be allowed to enter text and click on the search button to search through all the content on the Web site.

Shown here is the format for the ecmSearch tag.

```
<CFMODULE template="#request.ecm.AppPath#ecmSearch.cfm"
    StartingFolder=" "
    Recursive=" "
    TargetPage=" "
    TextBoxSize=" "
    MaxCharacters=" "
    ButtonImageSrc=" "
    ButtonText=" "
    FontFace=" "
    FontColor=" "
    FontSize=" "
    Horizontal=" "
>
```

The following table provides information about all the variables for the search tag.

Attribute	Description	Required?
Starting Folder	The folder you wish to search	Yes
Recursive	1 or 0 (zero). Whether to search sub-folders. (0=No)	Yes
Target Page	The Web page to output the search results.	Yes
Text Box	The size of the display text box for user input.	Yes
Maximum Characters	The maximum characters the input text box should accept.	No
Button Image Source	The image location for the search button. Blank ("") = standard submit button.	No

Attribute	Description	Required?
Button Text	The text used for the button if the standard submit button is chosen, or alternative text for the button if an image is chosen	No
Font Face	The font used for the text in the search tag.	No
Font Color	The font color used for the text in the search tag. Blank ("") = Black.	No
Font Size	The font size for the text in the search tag.	No
Horizontal	Determines if the search options are in a row or a column	Yes
Spare1	Currently not used	Must Be ""

Search Tag Example

The following example places a text field and a search button in a table cell inside an ColdFusion template.

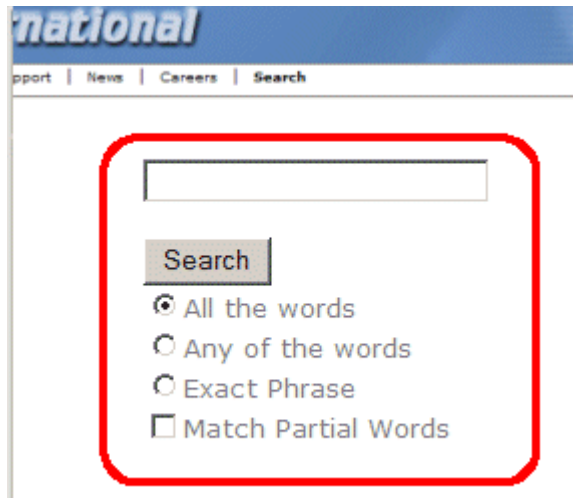
```
<tr>
  <td>
    <CFMODULE template="#request.ecm.AppPath#ecmSearch.cfm"
      StartingFolder="\marketing" Recursive="1" TargetPage="search-
display.cfm" TextBoxSize="25" MaxCharacters="200"
      ButtonImageSrc="" ButtonText="Search" FontFace="Verdana"
      FontColor="##808080" FontSize="2" Horizontal="0">
    </td>
  </tr>
```

The previous example has the following properties:

- Searches content in the `\marketing` folder
- Search is recursive
- The target page is `search-display.cfm`
- The size of the text box is 25
- The search allows a maximum of 200 characters
- No image is used for a button. Instead, a standard submit button is used.
- The text on the submit button is **Search**

- The search options appear in Verdana with a color of #808080 and a size of 2pt.
- The search options are arranged vertically

When added to a template, and visited, the search tag appears similar to the following:



Search Tag Variables

Listed below are the variables and their descriptions.

Starting Folder

In this variable, specify the content folder in which you want to begin the search. There are two options for this variable:

Value	Search begins
"\"	At the root folder
"folder path"	At the folder specified

Recursive

When you setup the search to be recursive, you are allowing the search to search all the sub-folders associated with the starting folder you had specified. You have two options for this variable:

Value	Description
1	Recursive search (will search sub-folders of the starting folder)
0	Non-Recursive search (will only search starting folder)

Target Page

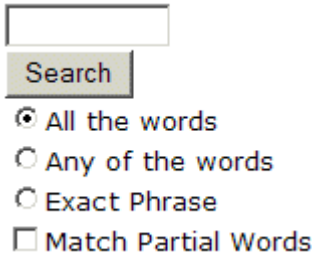
The target page is the page where the search results will be displayed. You may enter any form of a URL for the Web page you will have the search result displayed to. You can enter any Web page that has the searchdisplay tag inserted in it. If you want to use the same page, you can enter “#” into the field.

NOTE

Target page *must* have the searchdisplay function inserted in it for the search results to be displayed.

Text Box

This variable sets the size of the text box that is displayed. It can be any number depending on how big you would like the search text box to be.

Text Box Size	Web Page View
10	

Text Box Size	Web Page View
50	

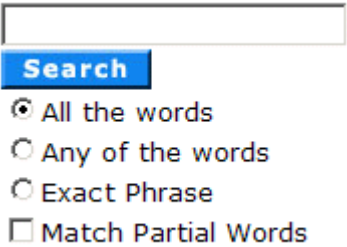
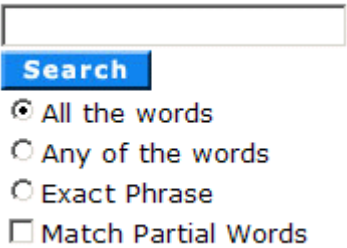
Maximum Characters

The value in this attribute determines the amount of characters

Button Image Source

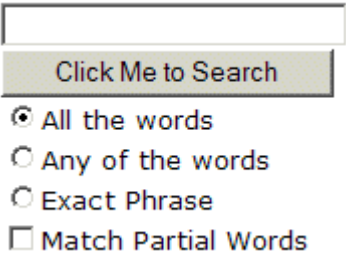
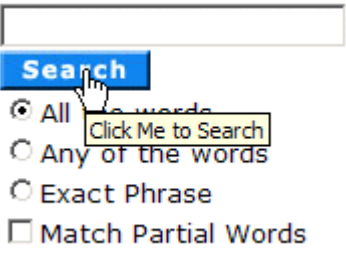
If you would like to use an image for the search button instead of the default, you may enter the filename and path of the image.

Button Image Source	Web Page View
image	

Button Image Source	Web Page View
"search.jpg"	 <p>The image shows a web page snippet with a search input field, a blue 'Search' button, 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>The image shows a web page snippet with a search input field, a blue 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox).</p>

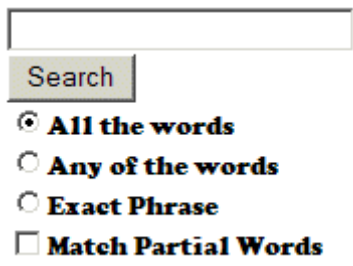
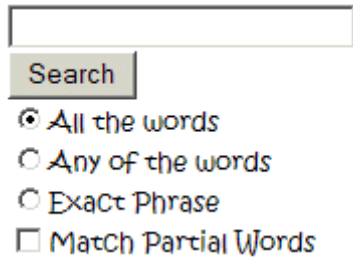
Button Text

If no search button image was specified, then the text that is entered in this field will be the text that appears on the button. If an image is being used for the search, then this text will be the alt text for the button.

Button Text	Web Page View
"Click Me to Search"	 <p>The screenshot shows a web page with a search input field and a button labeled "Click Me to Search". Below the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words".</p>
"Click Me to Search"	 <p>The screenshot shows a web page with a search input field and a button labeled "Search". A mouse cursor is hovering over the button, and a tooltip displays the text "Click Me to Search". Below the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words".</p>

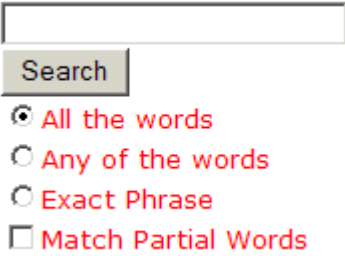
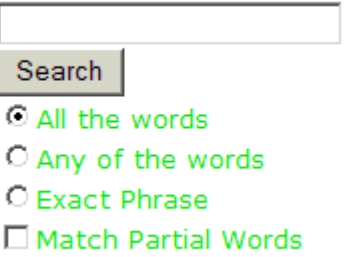
Font Face

This is the font that is used for the search options that are displayed on the Web page.

Font Face	Web Page View
Cooper Black	 <p>A screenshot of a web page showing search options. At the top is a text input field. Below it is a 'Search' button. Under the button are four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (disabled with a square checkbox).</p>
Kristen ITC	 <p>A screenshot of a web page showing search options. At the top is a text input field. Below it is a 'Search' button. Under the button are four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (disabled with a square checkbox).</p>

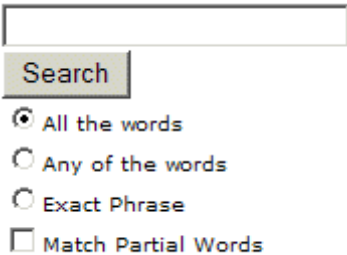
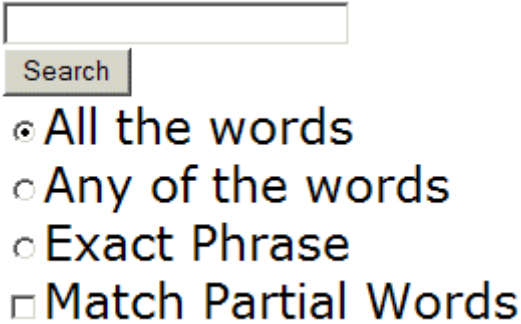
Font Color

This is the color of the text in the search tag. You may enter the color name or the hexadecimal value of the color.

Font Color	Web Page View
Red	 A screenshot of a web page search interface. It features a text input field at the top, followed by a 'Search' button. Below the button are four radio button options: 'All the words', 'Any of the words', 'Exact Phrase', and 'Match Partial Words'. All the text in this interface is colored red.
##14DA14	 A screenshot of a web page search interface, identical in layout to the one above. It has a text input field, a 'Search' button, and four radio button options. In this version, all the text is colored green, matching the hexadecimal code ##14DA14.

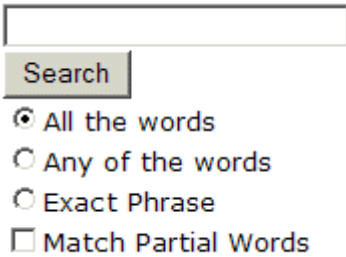
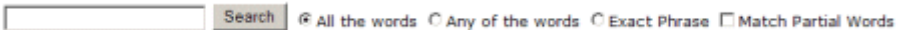
Font Size

This is the size of the text that is in the search tag.

Font Size	Web Page View
1	 <p>A screenshot of a web page search interface with a font size of 1. It features a search input field, a 'Search' button, and four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words' (checkbox). The text is extremely small and difficult to read.</p>
5	 <p>A screenshot of the same web page search interface with a font size of 5. The text is significantly larger and more legible than in the font size 1 example. The layout and elements are identical, including the search input, 'Search' button, and the four search options.</p>

Horizontal

This variable determines whether the search options are displayed vertically or horizontally.

Attribute	Web Page View
0	 A vertical layout of search options. It features a text input field at the top, followed by a 'Search' button. Below the button are four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.
1	 A horizontal layout of search options. It features a text input field and a 'Search' button side-by-side. To the right of the button are four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.

Spare 1

At the end of the Search tag, there is a variable called Spare 1. This variable has not yet been assigned an attribute. As of right now, when you enter a search tag, you must pass this variable ""

Search Display Function

The search display custom tag created the area where all the search results show after the search has been executed.

Shown here is the format of the ecmSearchDisplay tag:

```
<CFMODULE template="#request.ecm.AppPath#ecmSearchDisplay.cfm"
    MaxNumber=" "
    StyleInfo=" "
    ShowDate=" "
>
```

Below is a description of each of the values of the tag:

Attribute	Description	Required
Max Number of content blocks Returned	Maximum number of content blocks returned. (0=Unlimited)	Yes
Style Info	An HTML style string used within the HTML "hyperlink" generated by the search output. Can control font, color, size, etc...	No
ShowDate	Displays the last modified date of the content block. 0=No 1=Yes	No

Search Display Example

The following must be inserted on the page where you would like to display the search results. You may place it on the same page as the search tag, or insert it on a separate page.

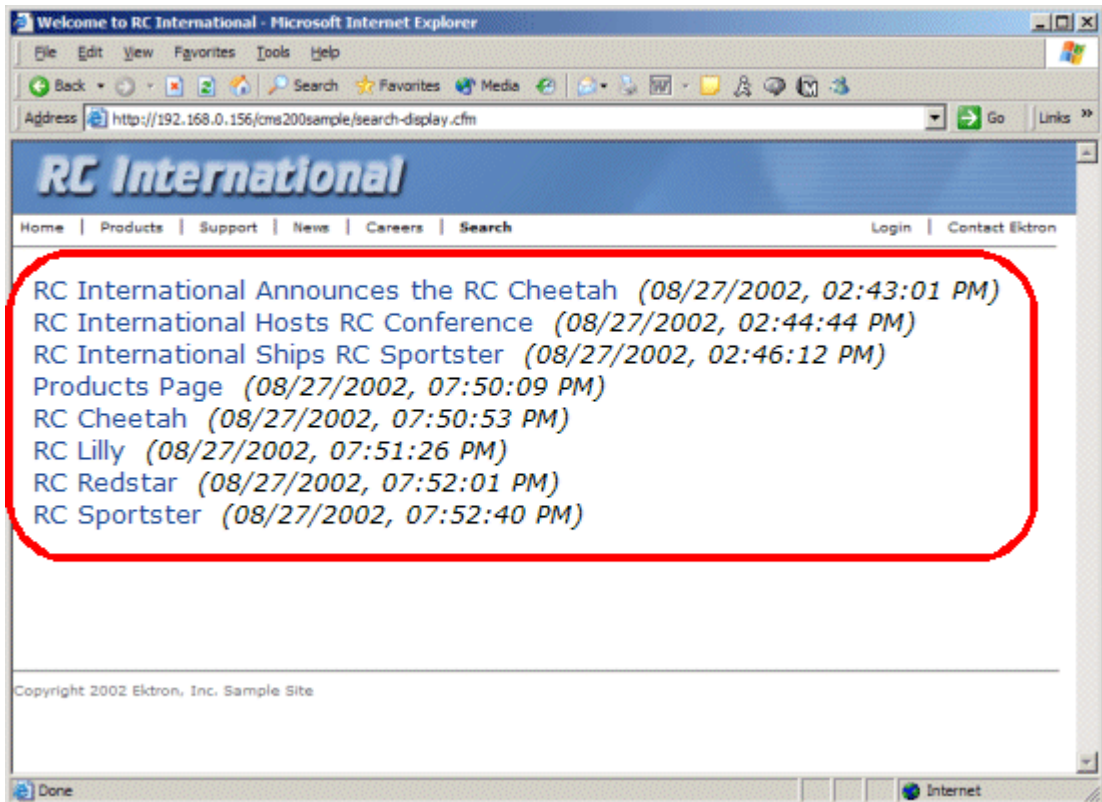
```
<tr>
  <td>
    <CFMODULE template="#request.ecm.AppPath#ecmSearchDisplay.cfm"
      MaxNumber="0" StyleInfo=" " ShowDate="1">
    </td>
</tr>
```

The previous search display example has the following properties:

- Returns unlimited results when the search is performed

- Contains no style information
- Shows the last modified date of the content

When inserted into a template, and a search has been performed, the results will appear similar to the following.



Search Display Variables

Listed below are all the variables for the SearchDisplay tag.

Max Number of Content Blocks Returned

This variable determines the amount of search results that will be displayed on the page. You may enter any integer into this variable. If you would like to display an unlimited amount, enter 0 (zero).

Max Returns	Web Page View
0	RC International Announces the RC Cheetah RC International Hosts RC Conference RC International Ships RC Sportster Products Page RC Cheetah RC Lilly RC Redstar RC Sportster
3	RC International Announces the RC Cheetah RC International Hosts RC Conference RC International Ships RC Sportster

Style Info

An HTML style string used within the HTML “hyperlink” generated by the search output. Can control font, color, size, etc.

Style Info	Web Page View
<pre> <<<< </pre>	RC International Announces the RC Cheetah RC International Hosts RC Conference RC International Ships RC Sportster Products Page RC Cheetah RC Lilly RC Redstar RC Sportster

Style Info	Web Page View
"font-family:arial; font-weight:bold; background-color:##cccccc; border:solid blue 1pt; margin:2px; width:50%;"	RC International Announces the RC Cheetah RC International Hosts RC Conference RC International Ships RC Sportster Products Page RC Cheetah RC Lilly RC Redstar RC Sportster

Show Date

Here, you can determine if you would like to include the Last Modified Date of the content block next to the result. This is useful to determine which is the most recent content block in the results. You have two choices for this:

Attribute	Web Page View
0	RC International Announces the RC Cheetah RC International Hosts RC Conference RC International Ships RC Sportster Products Page RC Cheetah RC Lilly RC Redstar RC Sportster

Attribute	Web Page View
1	<p>RC International Announces the RC Cheetah (08/27/2002, 02:43:01 PM) RC International Hosts RC Conference (08/27/2002, 02:44:44 PM) RC International Ships RC Sportster (08/27/2002, 02:46:12 PM) Products Page (08/27/2002, 07:50:09 PM) RC Cheetah (08/27/2002, 07:50:53 PM) RC Lilly (08/27/2002, 07:51:26 PM) RC Redstar (08/27/2002, 07:52:01 PM) RC Sportster (08/27/2002, 07:52:40 PM)</p>

List Summary Function

The list summary function is used to display the list of content summaries on a Web page.

Shown here is the format of the list summary function:

```
<CFMODULE template="#request.ecm.AppPath#ecmListSummary.cfm"
    Folder=" "
    MaxNumber=" "
    Recursive=" "
    ShowSummary=" "
    StyleInfo=" "
    OrderBy=" "
    ShowInfo=" "
>
```

A list and description of the variables are shown in this table:

Attribute	Description	Options
Folder	This is the folder the summary information will come from.	Any folder
MaxNumber	Maximum number of summaries returned.	0 = Unlimited
Recursive	Allow the tag to get summary info from child folders.	0 = Not Recursive 1 = Recursive
ShowSummary	Do you want the content summary to be displayed?	0 = No 1 = Yes
StyleInfo	Inline style information for the generated text hyperlink.	Any string
OrderBy, ASC/DESC	Order the hyperlinks by:	"Title" "DateModified" "DateCreated" "AuthorFname" "AuthorLname"
	Order the hyperlinks in ascending or descending order.	"Title,asc" "Title,desc"

Attribute	Description	Options
ShowInfo	Information that will appear to the right of the hyperlink.	"DateModified" "DateCreated" "AuthorFname" "AuthorLname"
Spare1	Not yet implemented	MUST BE ""
Spare2	Not yet implemented	MUST BE ""

List Summary Example

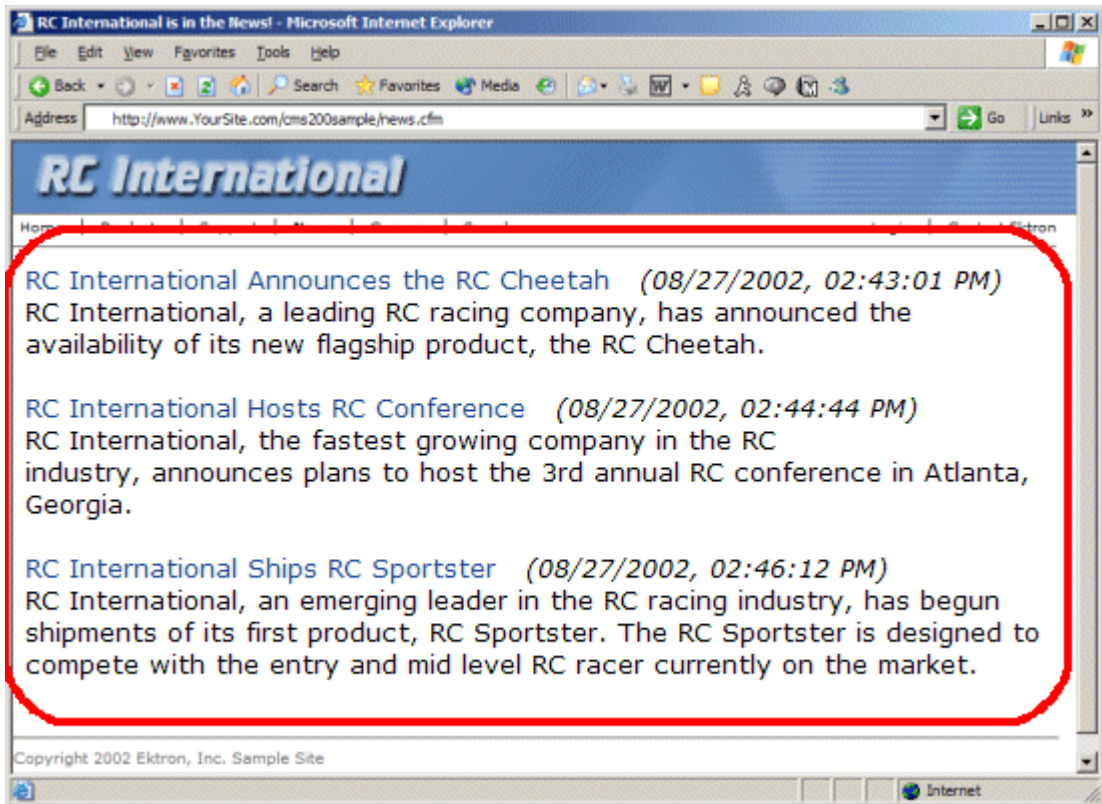
Shown below is an example of the ListSummary function in an ColdFusion Web page.

```
<TR>
  <TD>
    <CFMODULE template="#request.ecm.AppPath#ecmListSummary.cfm"
      Folder="\Marketing\News" MaxNumber="0" Recursive="0"
      ShowSummary="1" StyleInfo="" OrderBy="Title,asc" ShowInfo="" >
  </TD>
</TR>
```

The previous example has the following properties:

- Displays summaries from the \Marketing\News folder
- Unlimited results will be shown.
- Not recursive.
- The summaries will be shown
- No style was applied to the hyperlinks
- The results will be displayed in ascending order by title
- The date modified of each content block will be displayed

Here is how the summaries will appear on the web page.



List Summary Variables

Listed below are all the different variables and descriptions about them.

Starting Folder

In this variable, you will need to specify which content folder you will want to get all the summaries from. There are two options for this variable:

Value	Description
""	By leaving the value blank, all the summaries from the root content folder will be displayed.

Value	Description
"\Folder Path	You can specify a specific folder to display the summaries from by inserting the folder path

Recursive

When you setup the search to be recursive, you are allowing the search to search all the sub-folders associated with the starting folder you had specified. You have two options for this variable:

Value	Description
1	Recursive - summaries in the starting folder's sub-folders will also be displayed
0	Non-recursive - only summaries in the starting folder will be displayed.

Show Summary

The show summary attribute will allow or disallow the summary to be displayed in the summary list. You have two options for this variable:

Attribute	Web Page View
0	<p>RC International Ships RC Sportster</p> <p>RC International Hosts RC Conference</p> <p>RC International Announces the RC Cheetah</p>

Attribute	Web Page View
1	<p>RC International Ships RC Sportster 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.</p> <p>RC International Hosts RC Conference RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Style Info

In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

Style Info	Web Page View
""	<p>RC International Ships RC Sportster 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.</p> <p>RC International Hosts RC Conference RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Style Info	Web Page View
"font-family:arial; font-weight:bold; background-color:##cccccc; border:solid blue 1pt; margin:2px; width:100%;"	<p>RC International Ships RC Sportster</p> <p>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.</p> <p>RC International Hosts RC Conference</p> <p>RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah</p> <p>RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Order By

This variable will sort the summary results by different options. Here are the different ways to sort the summaries:

Value	Summaries sorted by
Title	title
DateModified	date modified
Date Created	date created
StartDate	Start Date assigned to the content block when created.
AuthorFname	authors' first name
AuthorLname	authors' last name

Ascending/Descending

This variable is an addition to the previous OrderBy variable. By default, the list of the summaries are descending, meaning they will follow the path of A-B-C, or 1-2-3. When set to ascending, they will order by Z-Y-X or 3-2-1. You will have the option to re arrange these to display either in ascending or descending order.

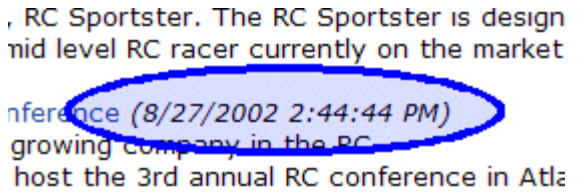
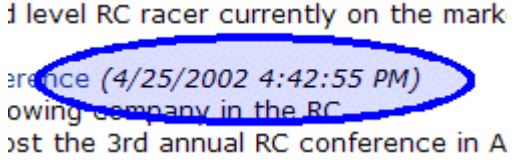
To do this, add the following to the OrderBy attribute:

Value	Description
asc, 1	This will order the Summaries in an ascending order
desc, 0	This will order the Summaries in a descending order. This is the default setting.

By using the ascending/descending option, you will have the option to set the order of content blocks from most recent to oldest, or vice versa. As well as ordering the list by the user's first names or last names in ascending or descending order.

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are four different options that you may choose from:

Value	Description	Web Page View
DateModified	Displays the date the content was last modified.	
DateCreated	Displays the date when the content was first created.	

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

Max Number

In this variable, you will determine how many summaries will be displayed on the page. You may enter any number into this field, and that number will be the amount of summaries that will be displayed. If you want all the summaries to be displayed, enter 0 (zero) and you will display unlimited results.

Max Number	Web Page View
0	<p>RC International Ships RC Sportster</p> <p>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.</p> <p>RC International Hosts RC Conference</p> <p>RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah</p> <p>RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Max Number	Web Page View
1	<p>RC International Ships RC Sportster</p> <p>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.</p>

Spare 1 & 2

There are two variable fields at the end that are currently not used. They are there for future scalability. When you insert a ListSummary tag into your template, you must leave those variables as ""

Single Summary Function

The single summary tag is used to display the summary of one content block in the Ektron CMS200/300 Web site, instead of a list of all the summaries in a content folder. Seen below is the single summary tag:

```
<cfmodule template="#request.ecm.AppPath#ecmSingleSummary.cfm"
    ContentID=" "
    ShowSummary="1"
    StyleInfo=" "
    ShowInfo=" "
>
```

You can look at the single summary tag as just a simpler form of the List summary tag. The format for each of them is almost identical, but the single summary tag does not require as many variables to be passed.

A list and description of the variables are shown in this table:

Attribute	Description	Required?
Content ID	This is the ID number of the content block the summary information will come from.	Yes
ShowSummary	Do you want the content summary to be displayed?	Yes
StyleInfo	Inline style information for the generated text hyperlink.	No
ShowInfo	Information that will appear to the right of the hyperlink.	No

Single Summary Example

Shown below is an example of the single summary function in an ColdFusion Web page.

```
<TR>
<TD>
    <cfmodule template="#request.ecm.AppPath#ecmSingleSummary.cfm"
        ContentID="14" ShowSummary="1" StyleInfo=" "
```

```

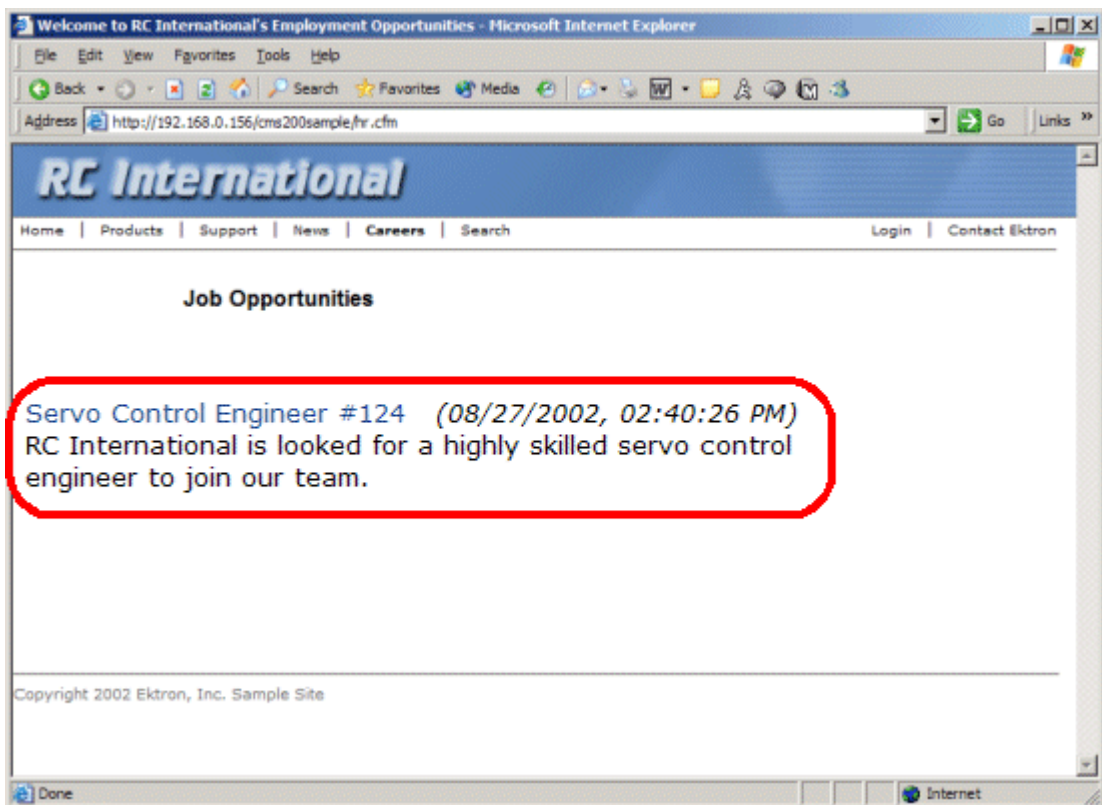
        ShowInfo="DateModified">
    </TD>
</TR>

```

The previous example had the following properties:

- Displays the summary for the content block with an ID=14
- The summary will be shown
- No style was applied to the hyperlinks
- The last date the content was modified will be displayed

Here is how the summary will appear on the web page.



Single Summary Variables

Listed below are all the different variables and descriptions about them.

Content ID

In this variable, you will need to specify which content block you will want the summary to be displayed from.

Show Summary

The show summary attribute will allow or disallow the summary to be displayed in the summary list. You have two options for this variable:

Value	Web Page View
0	Servo Control Engineer #124
1	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.

Style Info

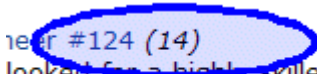
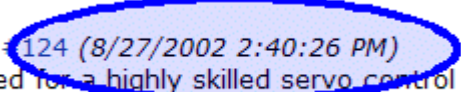
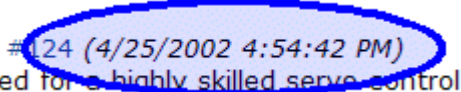
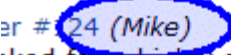
In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

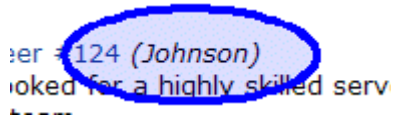
Style Info	Web Page View
""	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.

Style Info	Web Page View
font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:100%;	<div>Servo Control Engineer #124</div> <p>RC International is looked for a highly skilled servo control engineer to join our team.</p>

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are five different options that you may choose from:

Value	Displays the...	Web Page View
ID	ID number for the content block.	
DateModified	date the content was last modified.	
DateCreated	date when the content was first created.	
AuthorFname	first name of the user who last edited the content block.	

Value	Displays the...	Web Page View
AuthorLname	last name of the user who last edited the content block	

Spare 1

There is one variable fields at the end that are currently not used. This is here for future scalability. When you insert a single summary tag into your template, you must leave this variables as ""

Collection Function

The collection function is used to display the list of links that were created as a collection in the Ektron CMS200/300 Workarea. The ecmCollection function is highly customizable, allowing you to easily define how the collection data will be displayed on the Web page.

Shown here is the format of the ecmCollection function.

```
<CFMODULE template="#request.ecm.AppPath#ecmCollection.cfm"
           id="2"
           displayFunction="ecmTeaser"
>
```

Each of the attributes in the ecmCollection function are explained in the following table.

Attribute	Description
id	The id of the collection that you would like to be displayed is defined here.
displayFunction	Specify a function that is defined in the API that will define how the collection data is displayed on the Web page. The display function must also be defined in the ecmCollection function.

Collection Examples

The following are examples of the collection function in the Ektron CMS200/300 sample Web site. One example uses the “ecmNavigation” display function and the other uses the “ecmTeaser” display function.

The ecmNavigation and ecmTeaser display functions are sample display functions included with Ektron CMS200/300.

Example Using the ecmNavigation Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS200/300 sample Web site.


```
<CFMODULE id="1" displayFunction="ecmNavigation"
template="#request.ecm.AppPath#ecmCollection.cfm">
```

In this example, the collection with an ID=1 is displayed, and the function “ecmNavigation” is used to define how the collection data is displayed on the Web page.

Here is how a collection appears on a Web page.



As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used. Below is the source code for the display function “ecmNavigation”.

```
<cfparam name="Attributes.nId" default="">
<cfparam name="Attributes.href" default="">
<cfparam name="Attributes.target" default="">
<cfparam name="Attributes.startStyle" default="">
<cfparam name="Attributes.endStyle" default="">
<cfset LocalError = "">
<cfobject class="CMS200/300ContentBrv2.CMS200/300Content" Action="Create" name="cmsObj">
<cfset cNavs =
cmsObj.GetEcmCollectionItems(request.ecm.AppConfStr,Attributes.nId,0,request.ecm.cmsPreview,r
equest.ecm.user_id,request.ecm.site_id,LocalError)>
```

```

<cfif cNavs.Count(>
    <cfset nString = ">
    <cfloop collection=#cNavs# Item=nav>
        <cfset nString = nString & "<a href="">
        <cfif len(Attributes.href)>
            <cfif FindNoCase("?", Attributes.hrefs)>
                <cfset nString = nString & Attributes.href & "&id=" &
                    nav.Item("ContentID")>
            <cfelse>
                <cfset nString = nString & Attributes.href & "?id=" &
                    nav.Item("ContentID")>
            </cfif>
        </cfif>
    <cfelse>
        <cfset nString = nString & nav.Item("ContentLinks")>
    </cfif>
    <cfset nString = nString & "" ">
    <cfif len(Attributes.target)>
        <cfset nString = nString & "target="" & Attributes.target & "" ">
    </cfif>
    <cfset nString = nString & ">>
    <cfif len(Attributes.startStyle)>
        <cfset nString = nString & Attributes.startStyle>
    </cfif>
    <cfset nString = nString & nav.Item("ContentTitle")>
    <cfif len(Attributes.endStyle)>
        <cfset nString = nString & Attributes.endStyle>
    </cfif>
    <cfset nString = nString & "</a">
</CFLOOP>
    <cfmodule str="#nString#" nId="#Attributes.nId#"
template="#request.ecm.AppPath#ecmshowCollectionsContent.cfm">
</cfif>
<cfset nNavs = "Nothing">
<cfset cmsObj = "Nothing">

```

As you can see, the ecmNavigation function is a simple ASP function that creates a table of the links that belong to the collection specified. The function also displays the title of the content blocks by using the “info(“ContentTitle”)) building block.

Example Using the ecmTeaser Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS200/300 sample Web site.

```

<CFMODULE id="2" displayFunction="ecmTeaser"
template="#request.ecm.AppPath#ecmCollection.cfm">

```

In this example, the collection with an ID=2 is displayed, and the function “ecmTeaser” defines how to display the collection data on a Web page. Here is how the collection would appear on a Web page.



Employment Opportunities



Plastic Molder #123 (8/27/2002 2:38:34 PM)

RC International is looking for an experienced plastics molder.

Servo Control Engineer #124 (8/27/2002 2:40:26 PM)

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

As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used. Below is the source code for the display function “ecmTeaser”.

```
<cfparam name="Attributes.nID" default="">
<cfparam name="Attributes.href" default="">
<cfparam name="Attributes.ShowSummary" default="">
<cfparam name="Attributes.StyleInfo" default="">
<cfparam name="Attributes.target" default="">
<cfset ErrorString = "">
<cfobject class="CMS200/300ContentBrv2.CMS200/300Content" Action="Create" name="tObj">
<cfset cTeasers = tObj.GetEcmCollectionItems(request.ecm.AppConfStr, Attributes.nID, 0,
request.ecm.cmsPreview, request.ecm.user_id, request.ecm.site_id, ErrorString)>
<cfset tString = "<table border=""0"" cellspacing=""0"" cellpadding=""0"" width=""100%"">">
<cfif cTeasers.Count()>
  <cfloop collection=#cTeasers# Item=cT>
    <cfset tString = tString & "<tr><td>">
    <cfset tString = tString & "<a href="">">
    <cfif len(Attributes.href)>
      <cfif FindNoCase("?", Attributes.hrefs)>
        <cfset tString = tString & Attributes.href & "&id=" &
          cT.Item("ContentID")>
      <cfelse>
        <cfset tStringi = tString & Attributes.href & "?id=" &
          cT.Item("ContentID")>
      </cfif>
    </cfif>
    <cfset tString = tString & cT.Item("ContentLinks")>
    </cfif>
    <cfset tString = tString & "" " ">
    <cfif len(Attributes.target)>
      <cfset tString = tString & "target=""& Attributes.target & "" " ">
    </cfif>
    <cfset tString = tString & ">">
    <cfset tString = tString & cT.Item("ContentTitle") & "</a>" & "&nbsp;((" &
      DateFormat(cT.Item("DateModified"), "mm/dd/yyyy") & " " &
      TimeFormat(cT.Item("DateModified"), "h:mm:ss tt") & ")</td></tr><tr><td>" &
```

```

cT.Item("ContentTeaser")>
<cfset tString = tString & "</td></tr><tr><td>&nbsp;</td></tr>">
</cfloop>
<cfset tString = tString & "</table>">
<cfmodule str="#tString#" nId="#Attributes.nId#"
template="#request.ecm.AppPath#ecmshowCollectionsContent.cfm">
</cfif>

```

As you can see, the ecmTeaser function is a simple ASP function that creates a table of the links that belong to the collection specified.

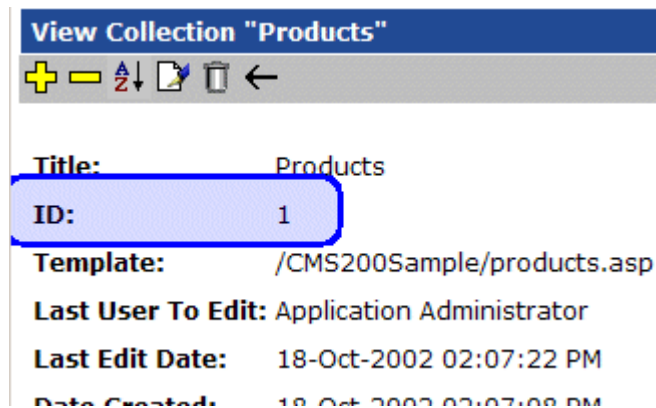
The ecmTeaser function also displays the:

- Content Title
- Date Modified
- Content Teaser

Collection Variables

ID

The collection ID is the ID number that is assigned to the collection when it is created in the workarea.

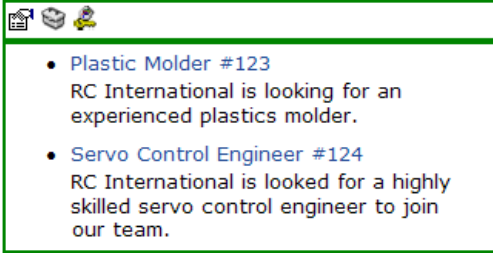


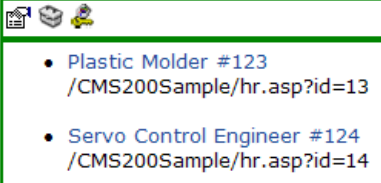


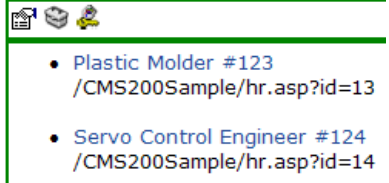



DisplayFunction


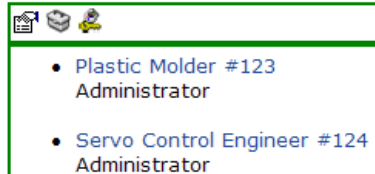

The display function defines how collection data appears on a Web page. Using ColdFusion scripting, you can create your own display functions, or use the three included functions.

Building Blocks

This section explains each possible building block that you may use when creating a custom display function.

Name	Description	Web Page View
info("ContentTeaser")	Displays the summary that has been created for the content block that is displayed in the collection data.	 <ul style="list-style-type: none"> • 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.
info("Comment")	Displays the comment added to the content block.	 <ul style="list-style-type: none"> • Plastic Molder #123 This is the historical comment for Plastics Molder #123 • Servo Control Engineer #124 This is the historical comment for Servo Control Engineer #124
info("ContentID")	Displays the ID number of the content block that is assigned to it by Ektron CMS200/300.	 <ul style="list-style-type: none"> • Plastic Molder #123 ID= 13 • Servo Control Engineer #124 ID= 14
info("ContentLinks")	Inserts the path of the content block with respect to your site root.	 <ul style="list-style-type: none"> • Plastic Molder #123 /CMS200Sample/hr.asp?id=13 • Servo Control Engineer #124 /CMS200Sample/hr.asp?id=14

Name	Description	Web Page View
info("ContentTitle")	Displays the title of the content block that is displayed with the collection data.	 <ul style="list-style-type: none"> • Plastic Molder #123 /CMS200Sample/hr.asp?id=13 • Servo Control Engineer #124 /CMS200Sample/hr.asp?id=14
info("DateModified")	Displays the date and time the content block was last modified.	 <ul style="list-style-type: none"> • Plastic Molder #123 8/27/2002 2:38:34 PM • Servo Control Engineer #124 8/27/2002 2:40:26 PM
info("DisplayDateModified")	Displays the date and time the content block was last modified using the date/time format specified in Setup.asp.	 <ul style="list-style-type: none"> • Plastic Molder #123 27-Aug-2002 02:38:34 PM • Servo Control Engineer #124 27-Aug-2002 02:40:26 PM
info("DisplayEndDate")	Displays the end date and time for the content block using the date/time format specified in Setup.asp.	
info("DisplayGoLive")	Displays the start date and time for the content block using the date/time format specified in Setup.asp.	 <ul style="list-style-type: none"> • Plastic Molder #123 This content went live on: 30-Oct-2002 10:58 AM • Servo Control Engineer #124 This content went live on: 29-Oct-2002 06:51 PM

Name	Description	Web Page View
info("EditorFname")	Displays the first name of the user who last edited the content block.	
info("EditorLname")	Displays the last name of the editor who last edited the content block.	
info("EndDate")	Displays the end date and time for the content block.	
info("GoLive")	Displays the start date and time for the content block.	

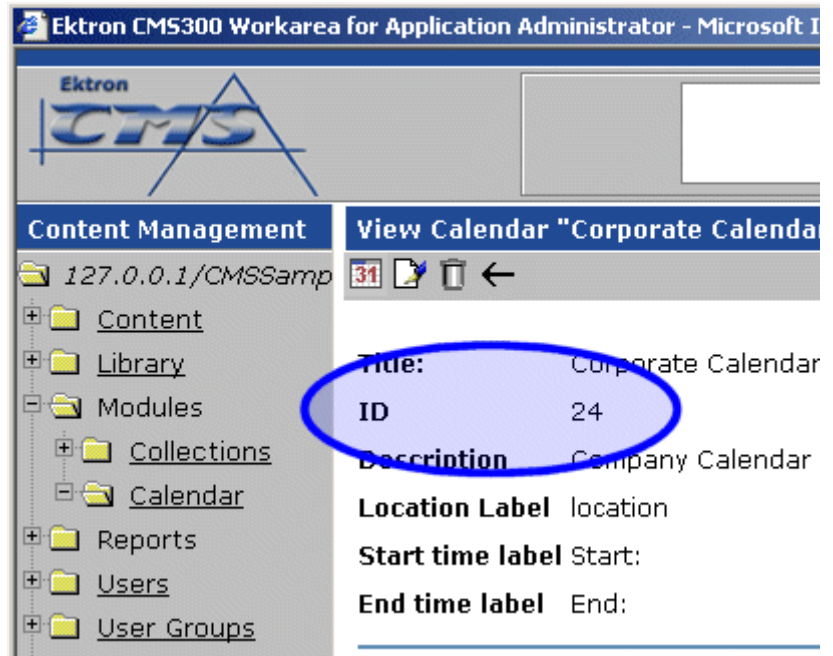
Event Calendar Function

The ecmEvtCalendar function is used to display an event calendar that has been created in the Ektron CMS200/300 Workarea.

Displayed here is the format for the event calendar custom function:

```
<CFMODULE template="#request.ecm.AppPath#ecmEvtCalendar.cfm" Calendar_ID=" " >
```

Where the "Calendar_id" represents the ID number of the event calendar in the Workarea.



Event Calendar Style Sheet

Before you insert an event Calendar into your Ektron CMS200/300 template, it is recommended that you make a reference to the style sheet, "\CMS200/300Sample\ekCalendar.css" in your template.

This style sheet contains styles that define the Calendar.

Event Calendar Sample

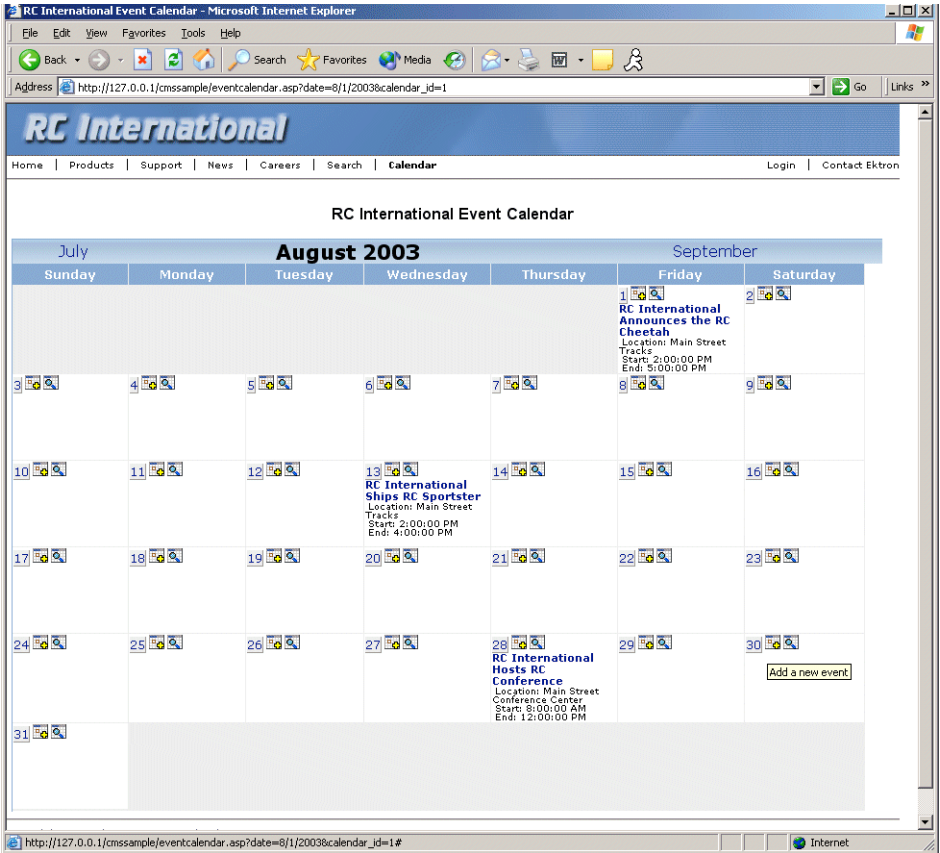
Displayed here is an example of the event Calendar in an Ektron CMS200/300 template.

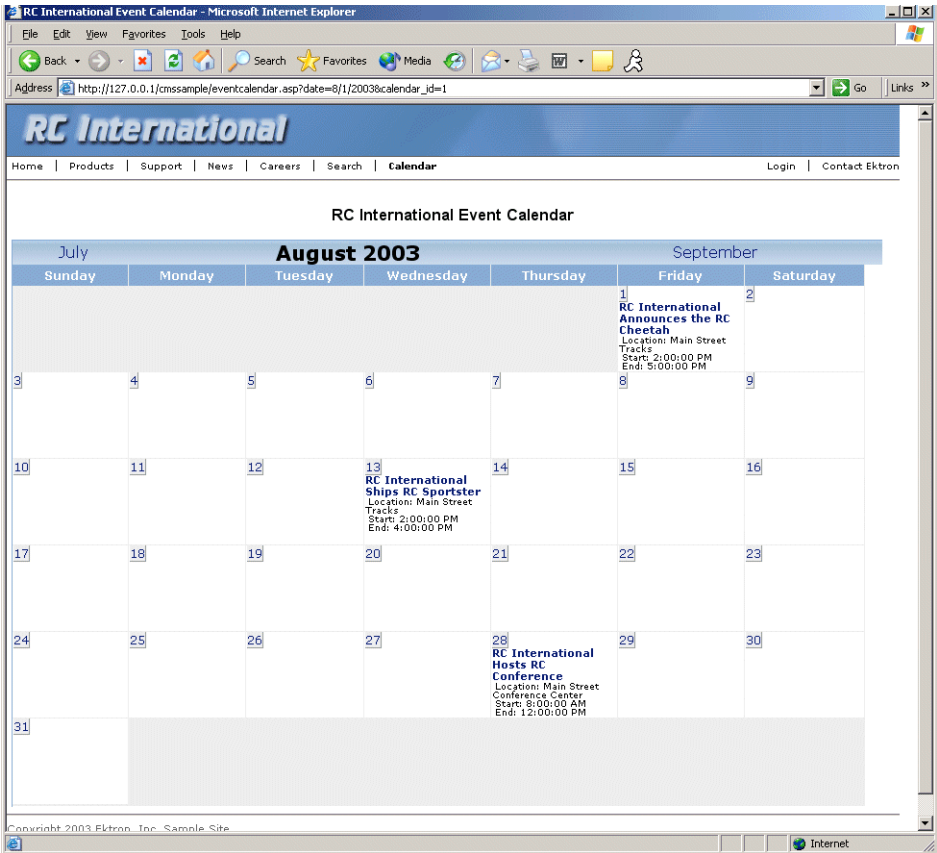
```
<tr>
  <td>

    <CFModule template="#request.ecm.AppPath#ecmEvtCalendar.cfm" calendar_id="24">

    </td>
  </tr>
```

:When this code is displayed in a browser, the Calendar with an ID-1 is displayed. Shown here is the Calendar displayed on the Web Page

If...	Calendar display
Logged in	

If...	Calendar display
Logged Out	

As you can see, when you are logged into the Web site, you are given options to add and view Calendar events. When logged out, only the calendar events are displayed.

Abbreviating Day Names

To further customize the event calendar function, you may specify in `ecmEvtCalendar.cfm` to display abbreviated day names (for example, Mon instead of Monday, Wed instead of Wednesday).

Displayed here is an example of the ecmEvtCalendar using abbreviated day names.

November		December 2003			January
Mon	Tue	Wed	Thu	Fri	
1	2	3	4	5	
8	9	10	11	12	

Configuring ecmEvtCalendar.cfm

To configure the ecmEvtCalendar function to display abbreviated day names, perform the following steps.

NOTE Before editing ecmEvtCalendar.cfm, Ektron recommends to create a copy of it and make changes to the new file. See "Custom ColdFusion Tags" on page 301 for additional information.

1. Open the ecmEvtCalendar.cfm file that you want to edit.
2. Locate the line that reads

```

    <cfset x=infoObj.Add("AbbreviateDayNames", FALSE)>
34 <cfset x=infoObj.Add("PathInfo",cgi.PATH_INFO)>
35 <cfset x=infoObj.Add("ServerName",cgi.SERVER_NAME)>
36 <cfset x=infoObj.Add("PreviewMode",request.ecmPreviewMode)>
37 <cfset x=infoObj.Add("AbbreviateDayNames", FALSE)>
38
39 <cfobject class="#request.ecm.MODULE_OBJ#" Action="Create"
40 <cfset variables.CalView = calendarObj.ecmCalendar(request)

```

3. On that line, change **false** to **true**. This instructs ecmEvtCalendar to use abbreviated day names.
4. Save and close the file.
5. View the Web page that contains the updated function.

Form Function

The `ecmFormBlock` function displays an HTML form content block. For more information about HTML forms, see the Ektron CMS200/300 Administrator manual chapter “Working with HTML Forms.”

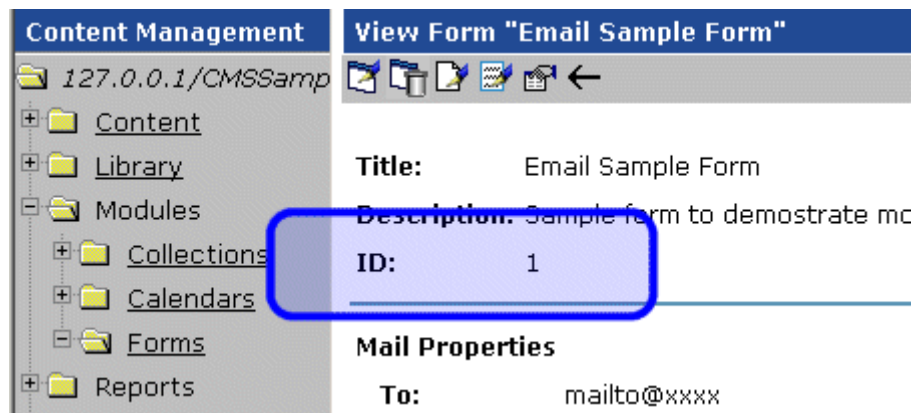
IMPORTANT!

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 CMS200/300 Administrator manual chapter “Managing Library Assets”, section “Updating Default Template for Multiple Quicklinks.”

Here is the format for the form custom function

```
<CFMODULE template="#request.ecm.AppPath#ecmFormBlock.cfm" ID="" >
```

The “id” argument represents the form’s ID number, visible when viewing the form from the View Form screen.



NOTE

As an alternative, you can use the multipurpose function, which can display either a content block or a content block associated with a form. For more information, see “MultiPurpose Function” on page 294.

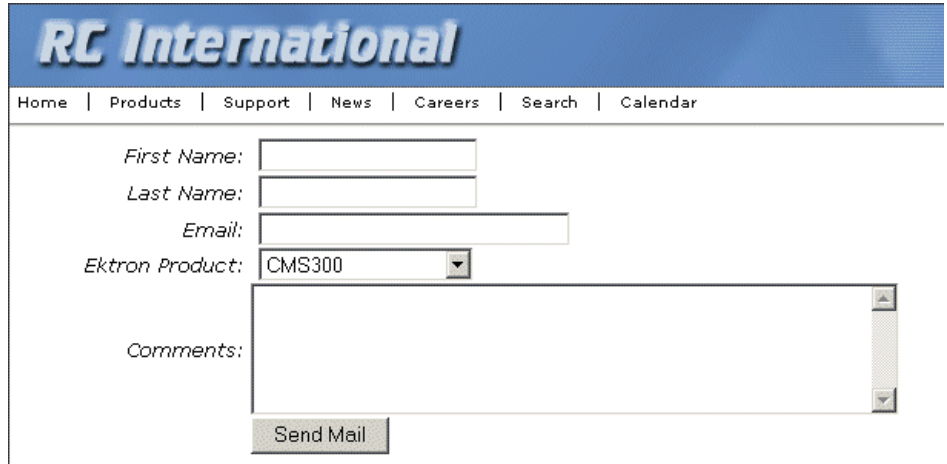
Here is an example of a form content block in an Ektron CMS200/300 template.

```
<tr>
  <td>
```

```
<CFModule template="#request.ecm.AppPath#ecmFormBlock" calendar_id="1">
```

```
</td>  
</tr>
```

When this code is read by a browser, the following is displayed.



The screenshot shows a web form for "RC International". The header is a blue bar with the company name in a stylized font. Below the header is a navigation menu with links: Home, Products, Support, News, Careers, Search, and Calendar. The form itself is white and contains the following fields: "First Name:" with a text input, "Last Name:" with a text input, "Email:" with a text input, "Ektron Product:" with a dropdown menu showing "CMS300", and "Comments:" with a large text area. A "Send Mail" button is located at the bottom of the form.

When a visitor to your site submits a form, the form uses the form information to process the data.

MultiPurpose Function

The ecmMultiPurpose function displays either a content block or an content block associated with a form. As a result, you can use one template to display both types of content block.

Here is the format for the Multipurpose function

```
<CFMODULE template="#request.ecm.AppPath#ecmMultiPurpose.cfm" DefaultID="" >
```

To pass in a standard content block, pass the id value in the url parameter, as illustrated below.

```
http://localhost/cms300sample/index.asp?id=2
```

To pass in a Form content block, pass the ekfrm value in the url parameter, as illustrated below.

```
http://localhost/cms300sample/index.asp?ekfrm=2
```

NOTE For more information about passing the id value as a url parameter, see ["Dynamic Content Block"](#) on page 244.

If the id value or ekfrm value does not exist, a blank template appears.

If the url parameter is not passed in or recognized, the function calls ecmContentBlock and passes the DefaultID value as the id argument. If the DefaultID value is zero (0), ecmContentBlock is not called.

Show Site Language Function

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

Site Language: English (US)

- Danish
- English (US)
- French (Standard)
- German (Standard)

(For more information, see the “Multi-Language Support “ chapter of the **Ektron CMS200/300** Administrator manual.)

Below is an example of the Show Site Language custom tag.

```
<CFMODULE template="#request.ecm.AppPath#ecmShowSiteLanguage.cfm" FieldName="" >
```

You can place this function in any location of any page on your site. By default, it appears in the lower right corner of the home page.

The variable is described in this table.

Variable	Description
FieldName	<p>An optional argument. You can leave it blank and allow the drop-down list to be populated by the available languages for your Web site. For example:</p> <pre><%=ecmShowSiteLanguage('')%></pre> <p>Or you can pass in an argument such as:</p> <pre><%=ecmShowSiteLanguage('mysitelanguage')%></pre>

Metadata Function

The metadata tag is used to display the metadata for one or more content blocks in the Ektron CMS200/300 Web site. Inserting this custom tag will allow you to generate the list of metadata that is edited by the authors, and insert it in the source of the Web page. Shown below is an example of what the metadata custom tag looks like.:

```
<cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm" ItemList="" >
```

A list and description of the variables are shown in this table:

Attribute	Description	Required?
ItemList	List of the content block ID numbers, and exclusions	No

Creating an Item List

The ItemList for the ecmMetadata tag has the following format:

```
"[content block id; Exclusion list]...[content block id; Exclusion list]"
```

Attribute	Description	Required?
[(left bracket)	Delineates the beginning of a content block id/Exclusion List pair.	Yes
Content Block ID	A number that represents the content block you wish to retrieve the metadata from	Yes
; (semicolon)	Separator: Separates the content block id from the exclusion list	No (Yes if including an exclusion list)
Exclusion List	A comma delimited list of metadata you do not wish to be retrieved for this content block.	No
] (right bracket)	Delineates the end of a content block id/Exclusion List pair.	Yes

ecmMetadata Example

For this example, our administrator has defined the following metadata types for Ektron CMS200/300.

Name	Style	Type	Editable	Required	Remove Dups	Case Sensitive
Title	n/a	HTML	Yes	Yes	No	n/a
Keywords	Name	Meta	Yes	Yes	Yes	No
Content-Type	http-equiv	Meta	No	Yes	Yes	n/a

Our example Web page has two content blocks displayed. Content block 1, the company's home page introduction text, and content block 2, a sidebar containing the lead story for the week. The metadata entered by your content editors for these two content blocks are:

Content Block 1:

Title – "Welcome to Our Home Page"

Keywords – "Home; page; company; Welcome"

Content-Type -- "text/html; charset=iso-8859-1"

Content Block 2:

Title – "Announcing our New Product!"

Keywords – "Announce; company; product; widget"

Content-Type -- "text/html; charset=iso-8859-1"

To include both content blocks metadata in the web page our ecmMetadata call would look like:

```
<html>
  <head>
    <cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm"
      ItemList="[1][2;Title,Content-Type]" >(rest of head)
    </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

In the above example the ecmMetadata call is contained in the <head> of the HTML document (since all HTML metadata is required to exist there). Once the Web page is viewed the Web page source will contain the following information:

```
<html>
  <head>
    <Title>Welcome to Our Home Page</Title>
    <meta name="Keywords" content= Home; page; company; Welcome; Announce;
product; widget">
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    (rest of head)
  </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

The ecmMetadata constructed these three HTML tags based on the administrator defined parameters and the ecmMetadata call.

The metadata tag was constructed in the page because:

- **[1]** - Will display all the metadata content for the content block ID=1. In this example, the title, keywords, and content-type were defined, so they were all displayed as:

```
<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome; Announce;
product; widget">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

- **[2;title,content-type]** - Will display all the metadata content for content block ID=2, EXCEPT for its title and content-type definitions.

```
<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome; Announce;
product; widget">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

The next section will provide you with a few more examples

ecmMetadata Variables

Listed below are all the different variables and descriptions about each of the them.

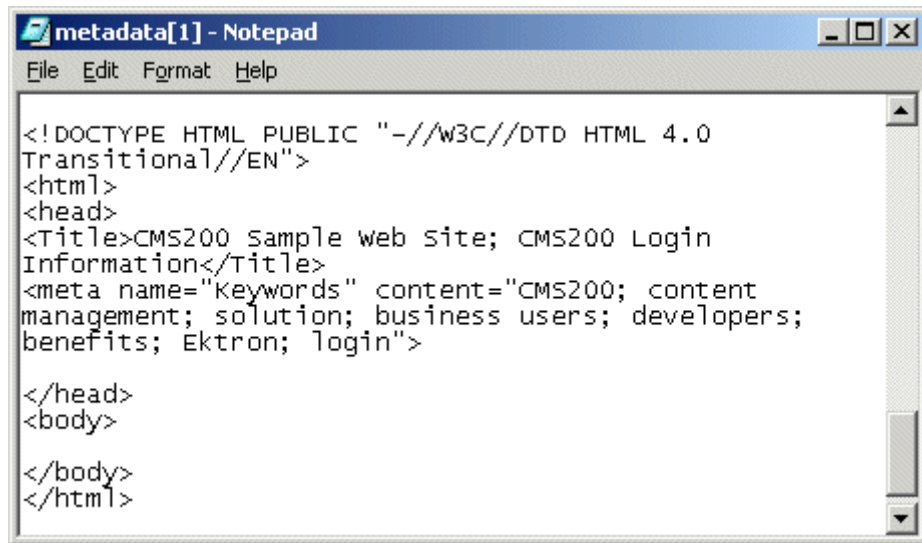
Item List

In this variable of the ecmMetadata tag, you will need to specify two values:

Value	Description
ID	Specify the content block ID to specify which content block will have the metadata displayed.
Exclusion List	After you specify the ID for the content block you want to display the metadata, you can also specify the metadata that you wish to be excluded.

Some examples might be:

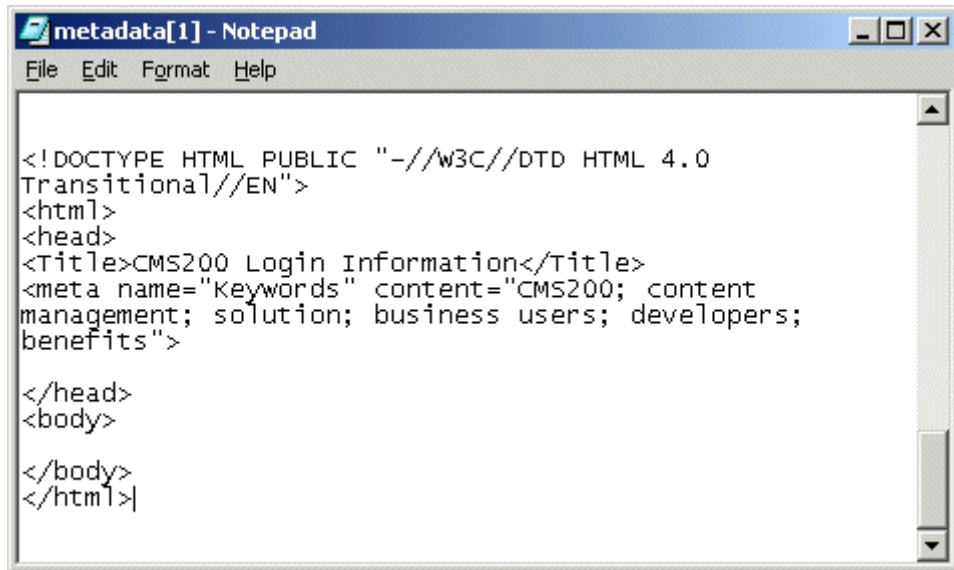
```
<cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm" ItemList="[1][2]">
```



In this example, all the metadata from the content block ID=1 and 2 will be displayed.

```
<cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm"
```

```
ItemList="[1;title][2;keywords]>
```



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Login Information</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

</head>
<body>

</body>
</html>

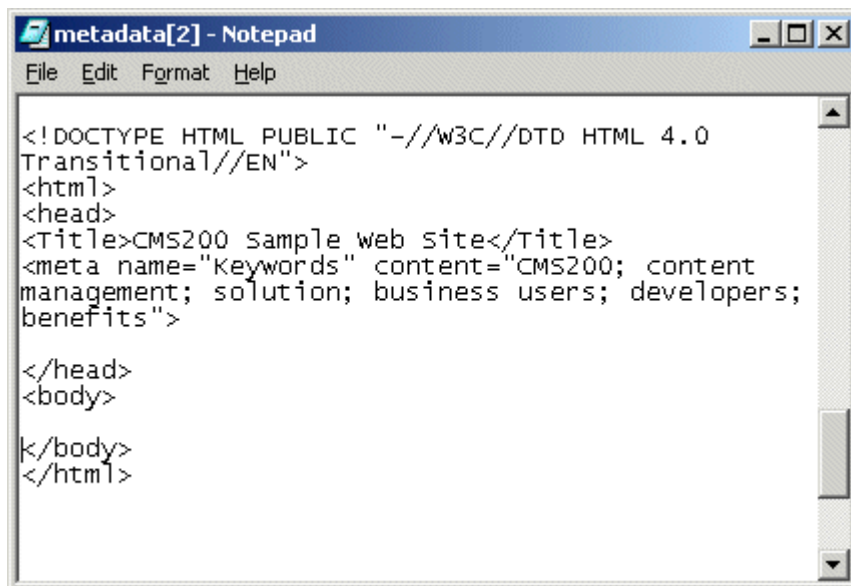
```

In this example, all the metadata from content block ID=1 will be displayed EXCEPT for the title. Also, all the metadata for content block ID=2 will be displayed Except for its keywords.

```

<cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm"
ItemList="[1][2;title,keywords]>

```



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Sample web site</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

</head>
<body>

</body>
</html>

```

In this example, all the metadata from content block ID=1 will be displayed, and all the metadata from content block ID=2 will be displayed except for its title and keywords.

Custom ColdFusion Tags

As you have seen, the formatting for the custom ColdFusion functions used in Ektron CMS200/300 for ColdFusion are defined in the following files:

- /CMS200/300Sample/CMS200/300scripts/ecmLogin.cfm
- /CMS200/300Sample/CMS200/300scripts/ecmContentBlock.cfm
- /CMS200/300Sample/CMS200/300scripts/ecmSearch.cfm
- /CMS200/300Sample/CMS200/300scripts/ecmSearchDisplay.cfm
- /CMS200/300Sample/CMS200/300scripts/ecmListSummary.cfm
- /CMS200/300Sample/CMS200/300scripts/ecmSingleSummary.cfm
- /CMS200/300Sample/CMS200/300scripts/ecmNavigation.cfm
- /CMS200/300Sample/CMS200/300scripts/ecmTeaser.cfm
- /CMS200/300Sample/CMS200/300scripts/ecmMetadata.cfm

As a developer, you may make changes to this file to further enhance how the custom functions appear on your Web site.

However, if you choose to extend the custom functions, Ektron recommends

- creating a copy of the file
- renaming it to an arbitrary name, for example, MySearch.cfm
- including that file when adding Ektron CMS200/300 tags to your templates

This procedure prevents your custom from being overwritten if you upgrade your version of Ektron CMS200/300.

Custom PHP Functions

Ektron CMS200/300 maintains and displays content via custom Ektron PHP functions and ComObjects. On your template, include the following file:

```
<?php include 'site_scripts_path.php'; ?>
```

This file calls the Ektron custom functions file to allow the page to function properly. It *must* appear at the top of the template file.



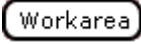
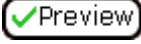

To customize you Ektron CMS200/300 application, you need to be familiar with these tags.

Tag	Description	More Information
Collection	Displays the links generated by collections.	"Collection Function" on page 342
Content Block	Displays and manages a content block.	"Content Block Tag" on page 306
Event Calendar	Displays an event calendar defined in the Workarea.	"Event Calendar Function" on page 355
Form	Displays an HTML form	"Form Function" on page 361
List Summary	Displays the list of the content block summaries.	"List Summary Tag" on page 328
Login	Paints the login or logout button which allows the user to login and use the Ektron CMS200/300 application.	"Login Tag" on page 304
Metadata	Displays the metadata in the source of the Web page the content is displayed on.	"Metadata Tag" on page 364
MultiPurpose	Displays either a content block or a content block associated with a form	"MultiPurpose Function" on page 363
Random Content	Randomly displays the body content of a collection item.	"Random Content" on page 351

Tag	Description	More Information
Random Summary	Randomly displays the summary content of a collection item.	"Random Summary" on page 353
Search	Inserts a text box on the template with a search button.	"Search Tag" on page 312
Search Display	Creates area where search results will be displayed.	"Search Display Tag" on page 323
Single Summary	Displays the summary for a single content block.	"Single Summary Tag" on page 337
Show Site Language Function	Lets user select language in which to view Web site.	"Show Site Language Function" on page 370

Login Tag

This tag is responsible for displaying the following buttons on the Web page:

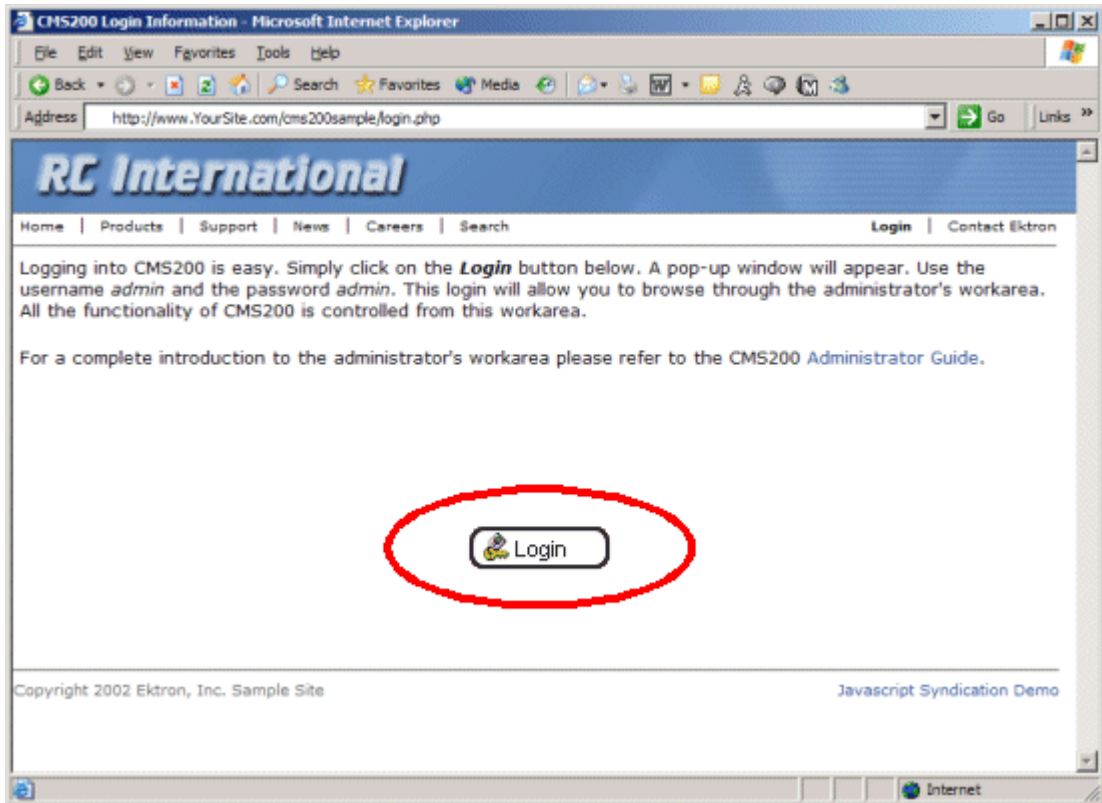
Button	Name	Description
	Login	When user is not logged into Ektron CMS200/300, this button will be displayed. Clicking this button will allow the user to log into the application.
	Logout	Once logged in, this button will replace the login button to allow the user to log out from the Ektron CMS200/300 Web site.
	Workarea	When logged in, this button will be displayed under the logout button allowing the user to access their workarea.
	Preview On	This button will allow the user to preview the entire Web site as if all checked-in content was published.
	Preview Off	Turns off the site preview mode.

Login Tag Example

The following example places a login button in a table cell inside a asp template.

```
<tr>
  <td>
    <?php ecmlogin(); ?>
  </td>
</tr>
```


When inserted properly, a login button will be placed on the Web page.



Clicking on the login tag will open the login window where a user will be able to enter their username and password. Upon successful completion, the user will then be logged into the Ektron CMS200/300 Web site.

Placing Your Login Button

You are able to add as many login buttons to a template as you would like. You might want to include a login button on each template, or have a special Web page called `login.aspx` where you can log in there without the public being able to access the login page.

Content Block Tag

The content block tag is used to display a content block on an Ektron CMS200/300 Web page.

NOTE As an alternative, you can use the [multipurpose function](#), which can display either a content block or a content block associated with a form. For more information, see ["MultiPurpose Function" on page 363](#).

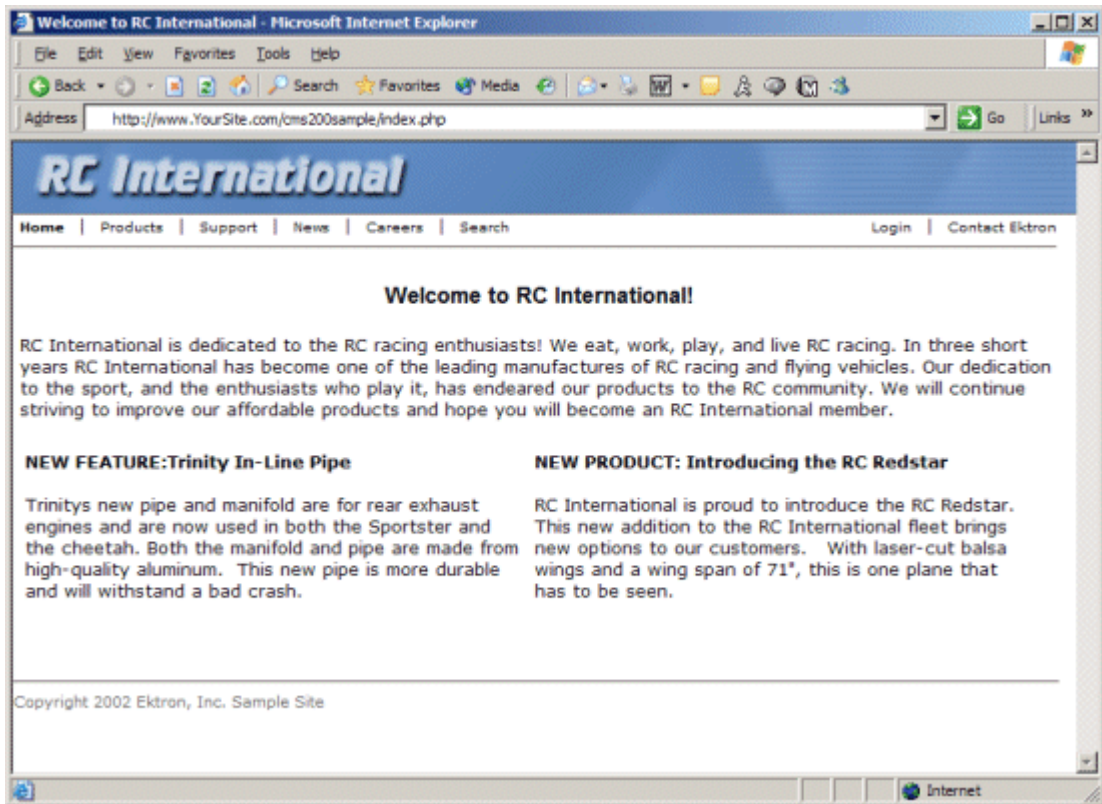
Shown here is the format for the content block function:

```
<?php ecmContentBlockEx (ID, "XSLT", Override) ?>
```

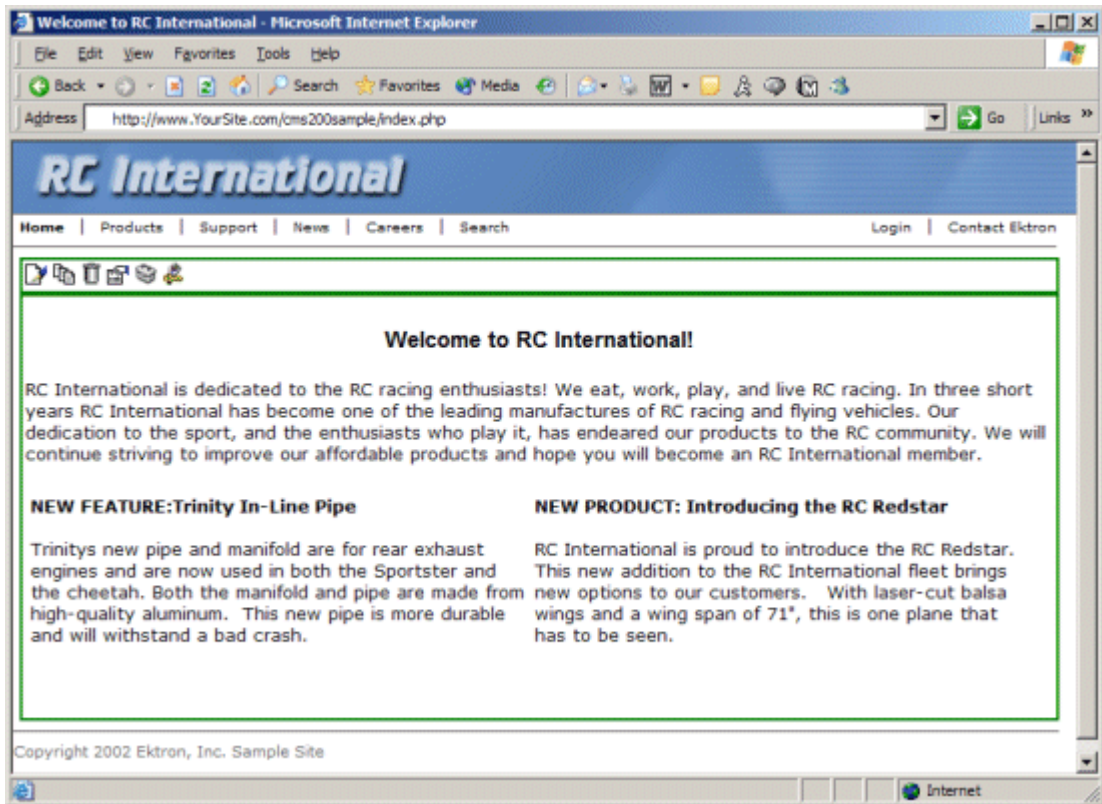
Explained in the table below are the attributes for the content block function.

Attribute	Description	Required
ID	ID number assigned to a content block when created in Ektron CMS200/300.	Yes
XSLT	File name and path to a valid XSLT to transform the XML content.	No
Override	If no XSLT is specified, then enter 1 to use XSLT1, 2 for XSLT2, or 3 for XSLT3.	No

In normal view, this tag will retrieve the content block from the database and return it to the calling template.



If a user is logged into Ektron CMS200/300, the content block tag will paint a border around the content block and provide the menu for managing that piece of content.



Content Block Examples

In Ektron CMS200/300, there are three different types of content blocks:

- Static content blocks
- Dynamic content blocks
- XML Content Blocks

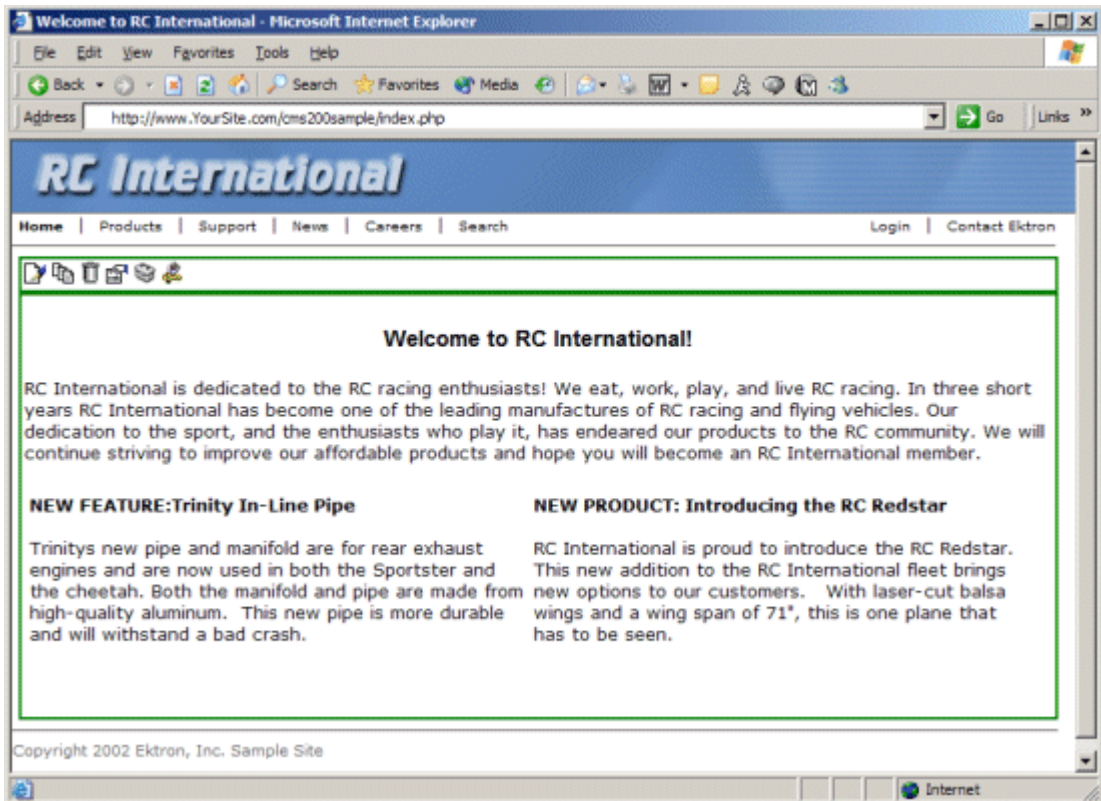
Static content blocks are used to display one specific content block on the Web page, where a dynamic content block is used to display the content block of the ID that is passed through the URL. XML content blocks are used to display XML content transformed by an XSLT specified in the function.

Static Content Block

The following example explains what to insert for a static content block on a template.

```
<tr>
  <td>
    <?php ecmContentBlock(1) ?>
  </td>
</tr>
```

In the example displayed above, the function will call the content block with an id of 1 from the database to the browser.



Dynamic Content Block

The following example shows how the content block tag can be used where the id is being passed in as a URL parameter. This is the tag which would be used on a dynamic template.

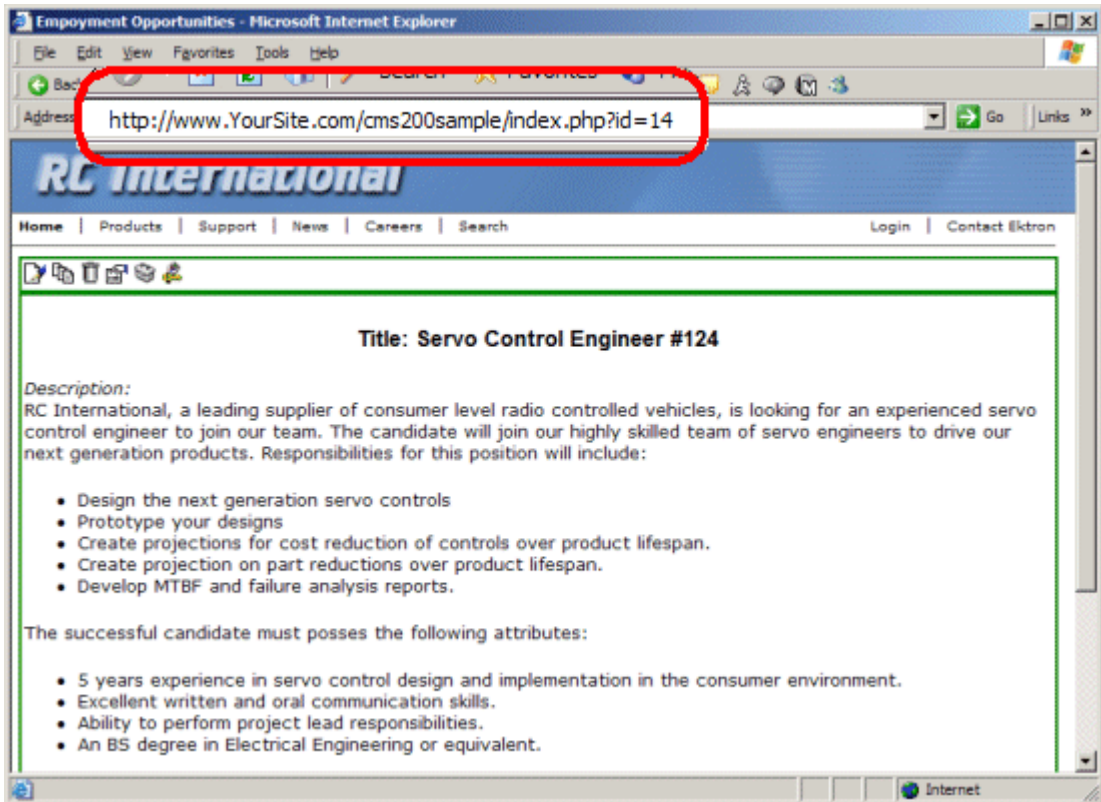
```
<tr>
  <td>
```

```

        <?php    ecmContentBlock($id) ?>
    </td>
</tr>

```

When a user accesses this page and adds the ID number to the template this tag lives on, the content block with the corresponding ID will be displayed on the Web page in the template that is specified.



XML Content Block

NOTE This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

Here is an example of displaying an XML content block. Note that you specify both a content block and an XSLT that determines how the content block is displayed.

```
<tr>
  <td>
    <?php ecmContentBlockEx (13, "", 1) ?>
  </td>
</tr>
```

This example would display the content block ID=13, and use the XSLT1 as the display XSLT.

To learn about the function's arguments, see ["The Function's Arguments" on page 53](#).

Search Tag

The search tag is used to display a text box, with a search button on the template. When seen on the template on the Web page, the users will be allowed to enter text and click on the search button to search through all the content on the Web site.

Shown here is the format for the ecmSearch tag:

```
<?php ecmSearch( "StartingFolder",
Recursive,
"targetpage",
textbox,
MaxCharacters,
"ButtonImgSrc",
"ButtonText",
"FontFace",
"FontColor",
"FontSize",
Horizontal,
"Spare" );
?>
```

The following table provides information about all the variables for the search tag.

Attribute	Description	Required?
Starting Folder	The folder you wish to search	Yes
Recursive	1 or 0 (zero). Whether to search sub-folders. (0=No)	Yes
Target Page	The Web page to output the search results.	Yes
Text Box	The size of the display text box for user input.	Yes
Maximum Characters	The maximum characters the input text box should accept.	No
Button Image Source	The image location for the search button. Blank ("") = standard submit button.	No

Attribute	Description	Required?
Button Text	The text used for the button if the standard submit button is chosen, or alternative text for the button if an image is chosen	No
Font Face	The font used for the text in the search tag.	No
Font Color	The font color used for the text in the search tag. Blank ("") = Black.	No
Font Size	The font size for the text in the search tag.	No
Horizontal	Determines if the search options are in a row or a column	Yes
Spare1	Currently not used	Must Be ""

Search Tag Example

The following example places a text field and a search button in a table cell inside an ASP template.

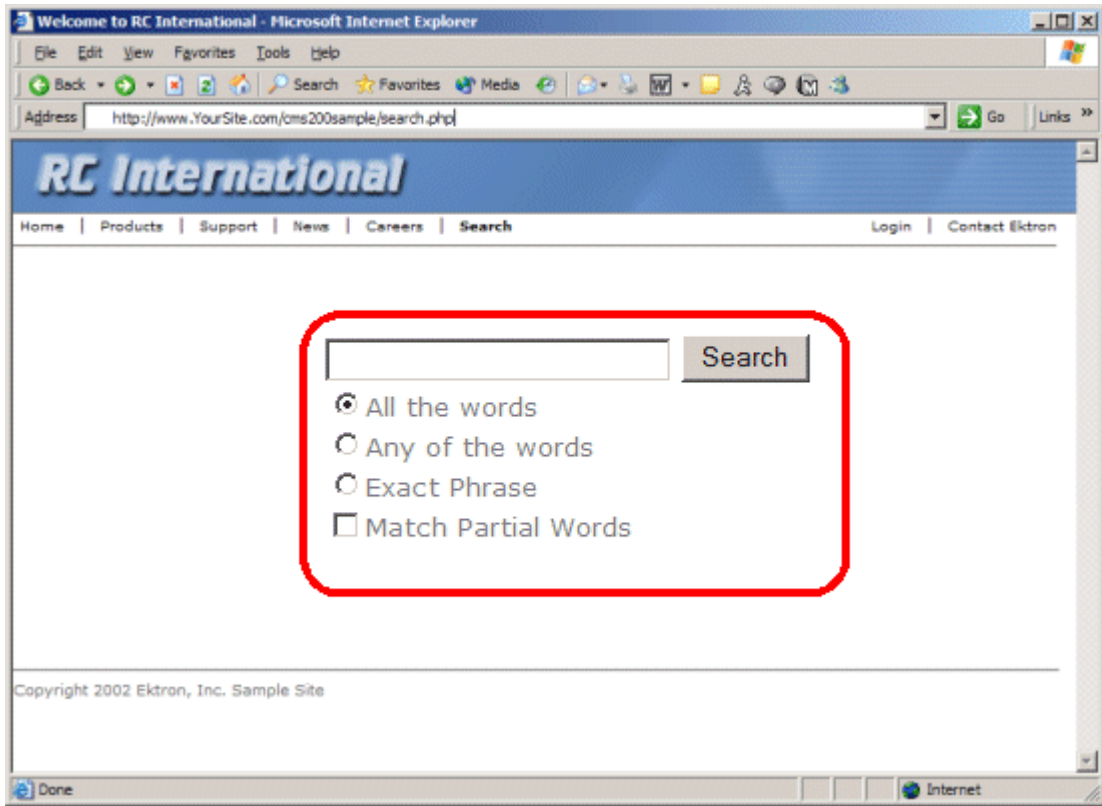
```
<TR>
  <TD>
    <?php ecmSearch("\\Marketing",1,"search-display.php",25,200,"",
      "Search","Verdana","#808080","2",0,""); ?>
  </TD>
</TR>
```

The previous example has the following properties:

- Searches content in the \content\marketing folder
- Search is recursive
- The target page is “search-display.php”
- The size of the text box is 25
- The search will allow a maximum of 200 characters
- No image is used for a button, instead a standard submit button is used.
- The text on the submit button is “Search”
- The search options appear in Verdana with a color of #808080 and a size of 2pt.

- The search options are arranged vertically

When added to a template, and visited, the search tag will appear similar to the following:



Search Tag Variables

Listed below are all the different variables and their descriptions about them.

Starting Folder

In this variable, you will need to specify which content folder you will want to begin the search. There are two options for this variable:

Value	Description
"\"	Search will begin the search at the root folder
"\\folder path"	Search will begin from the folder specified

Recursive

When you setup the search to be recursive, you are allowing the search to search all the sub-folders associated with the starting folder you had specified. You have two options for this variable:

Value	Description
1	Recursive search (will search sub-folders of the starting folder)
0	Non-Recursive search (will only search starting folder)

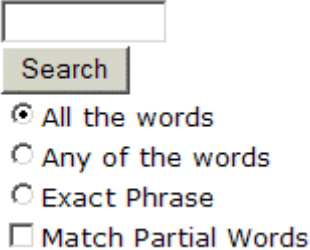
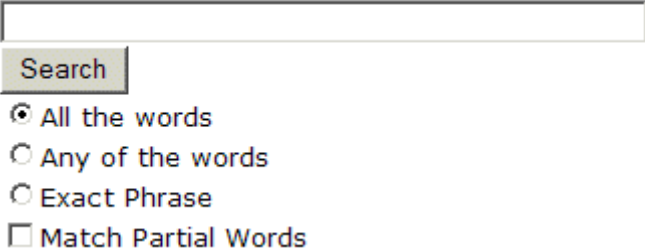
Target Page

The target page is the page where the search results will be displayed. You may enter any form of a URL for the Web page you will have the search result displayed to. You can enter any Web page that has the searchdisplay tag inserted in it. If you want to use the same page, you can enter "#" into the field.

NOTE Target page **MUST** have the searchdisplay function inserted in it for the search results to be displayed.

Text Box

This variable will set the size of the text box that is displayed. This can be any number depending on how big you would like the search text box to be.

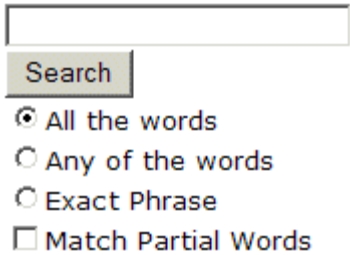
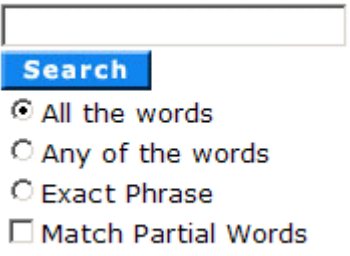
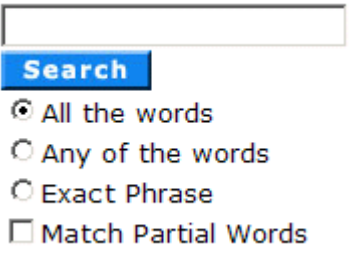
Text Box Size	Web Page View
10	
50	

Maximum Characters

The value in this attribute determines the amount of characters

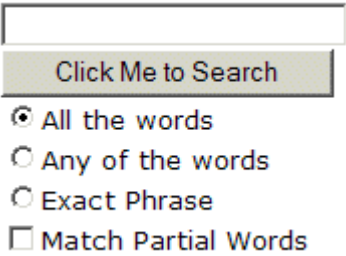
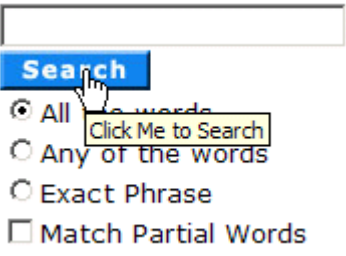
Button Image Source

If you would like to use an image for the search button instead of the default, you may enter the filename and path of the image.

Button Image Source	Web Page View
""	 <p>A screenshot of a web page showing a search form. It includes a text input field, a button labeled "Search", and four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words".</p>
"search.jpg"	 <p>A screenshot of a web page showing a search form. The button is a custom image labeled "Search". The rest of the form, including the input field and radio button options, is identical to the first screenshot.</p>
"http://www.ektron.com/images/search.jpg"	 <p>A screenshot of a web page showing a search form. The button is a custom image labeled "Search". The rest of the form, including the input field and radio button options, is identical to the previous screenshots.</p>

Button Text

If no search button image was specified, then the text that is entered in this field will be the text that appears on the button. If an image is being used for the search, then this text will be the alt text for the button.

Button Text	Web Page View
"Click Me to Search"	 <p>The screenshot shows a search interface. At the top is a text input field. Below it is a button with the text "Click Me to Search". Under the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words".</p>
"Click Me to Search"	 <p>The screenshot shows a search interface. At the top is a text input field. Below it is a blue button with the text "Search". A mouse cursor is clicking the button, and a tooltip with the text "Click Me to Search" is visible. Under the button are four radio button options: "All the words" (selected), "Any of the words", "Exact Phrase", and "Match Partial Words".</p>

Font Face

This is the font that is used for the search options that are displayed on the Web page.

Font Face	Web Page View
Cooper Black	<div><input type="text"/></div> <div>Search</div> <div><input checked="" type="radio"/> All the words</div> <div><input type="radio"/> Any of the words</div> <div><input type="radio"/> Exact Phrase</div> <div><input type="checkbox"/> Match Partial Words</div>
Kristen ITC	<div><input type="text"/></div> <div>Search</div> <div><input checked="" type="radio"/> All the words</div> <div><input type="radio"/> Any of the words</div> <div><input type="radio"/> Exact Phrase</div> <div><input type="checkbox"/> Match Partial Words</div>

Font Color

This is the color of the text in the search tag. You may enter the color name or the hexadecimal value of the color.

Font Color	Web Page View
Red	<div><input type="text"/></div> <div>Search</div> <div><input checked="" type="radio"/> All the words</div> <div><input type="radio"/> Any of the words</div> <div><input type="radio"/> Exact Phrase</div> <div><input type="checkbox"/> Match Partial Words</div>
#14DA14	<div><input type="text"/></div> <div>Search</div> <div><input checked="" type="radio"/> All the words</div> <div><input type="radio"/> Any of the words</div> <div><input type="radio"/> Exact Phrase</div> <div><input type="checkbox"/> Match Partial Words</div>

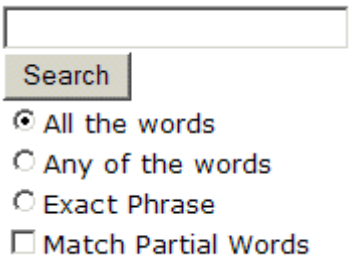
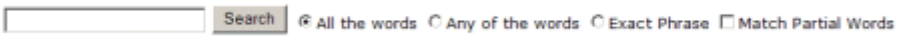
Font Size

This is the size of the text that is in the search tag.

Font Size	Web Page View
1	<div><input type="text"/></div> <div>Search</div> <div><input checked="" type="radio"/> All the words</div> <div><input type="radio"/> Any of the words</div> <div><input type="radio"/> Exact Phrase</div> <div><input type="checkbox"/> Match Partial Words</div>
5	<div><input type="text"/></div> <div>Search</div> <div><input checked="" type="radio"/> All the words</div> <div><input type="radio"/> Any of the words</div> <div><input type="radio"/> Exact Phrase</div> <div><input type="checkbox"/> Match Partial Words</div>

Horizontal

This variable determines whether the search options are displayed vertically or horizontally.

Attribute	Web Page View
0	 A vertical layout of search options. It features a text input field at the top, followed by a 'Search' button. Below the button are four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.
1	 A horizontal layout of search options. It features a text input field and a 'Search' button side-by-side. To the right of the button are four radio button options: 'All the words' (selected), 'Any of the words', 'Exact Phrase', and 'Match Partial Words'.

Spare 1

At the end of the Search tag, there is a variable called Spare 1. This variable has not yet been assigned an attribute. As of right now, when you enter a search tag, you must pass this variable ""

Search Display Tag

The search display custom tag created the area where all the search results show after the search has been executed.

Shown here is the format of the ecmSearchDisplay tag:

```
<?php ecmSearchDisplay( MaxNumbers,
                        "StyleInfo",
                        ShowDate );
?>
```

Below is a description of each of the values of the tag:

Attribute	Description	Required
Max Number of content blocks Returned	Maximum number of content blocks returned. (0=Unlimited)	Yes
Style Info	An HTML style string used within the HTML "hyperlink" generated by the search output. Can control font, color, size, etc...	No
ShowDate	Displays the last modified date of the content block. 0=No 1=Yes	No

Search Display Example

The following must be inserted on the page where you would like to display the search results. You may place it on the same page as the search tag, or insert it on a separate page.

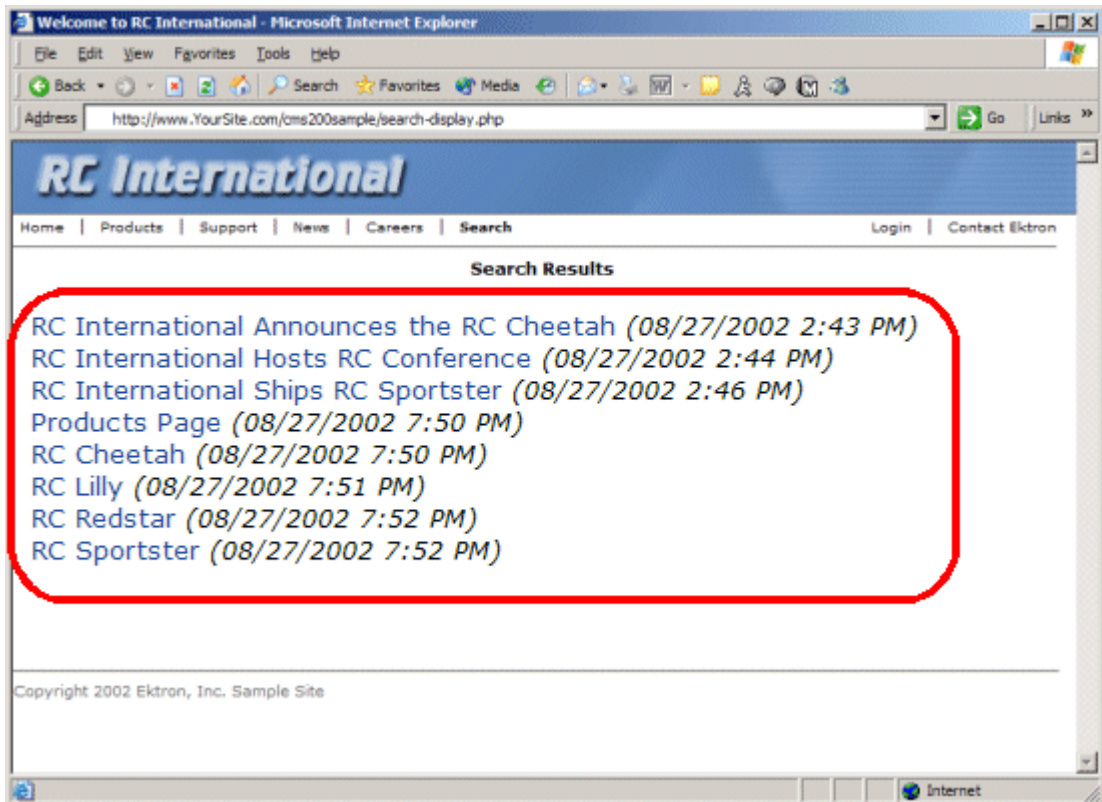
```
<tr>
  <td>
    <?php ecmSearchDisplay( 0,"",1); ?>
  </td>
</tr>
```

The previous search display example has the following properties:

- Returns unlimited results when the search is performed

- Contains no style information
- Shows the last modified date of the content

When inserted into a template, and a search has been performed, the results will appear similar to the following.



Search Display Variables

Listed below are all the variables for the SearchDisplay tag.

Max Number of Content Blocks Returned

This variable determines the amount of search results that will be displayed on the page. You may enter any integer into this variable. If you would like to display an unlimited amount, enter 0 (zero).

Max Returns	Web Page View
0	<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>
3	<p>Search Results</p> <p>RC Sportster RC Redstar RC Lilly</p>

Style Info

An HTML style string used within the HTML “hyperlink” generated by the search output. Can control font, color, size, etc.

Style Info	Web Page View
""	<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>
"font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:50%;"	<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>

Show Date

Here, you can determine if you would like to include the Last Modified Date of the content block next to the result. This is useful

to determine which is the most recent content block in the results.
You have two choices for this:

Attribute	Web Page View
0	<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>
1	<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>

List Summary Tag

The list summary function is used to display the list of content summaries on a Web page.

Shown here is the format of the list summary function:

```
<?php ecmListSummary ( "Folder",
                        Recursive,
                        ShowSummary,
                        "StyleInfo",
                        "OrderBy",
                        "ShowInfo",
                        MaxNumber,
                        "Spare 1",
                        "Spare2" );
?>
```

A list and description of the variables are shown in this table:

Attribute	Description	Options
Folder	This is the folder the summary information will come from.	Any folder
Recursive	Allow the tag to get summary info from child folders.	0 = Not Recursive 1 = Recursive
ShowSummary	Do you want the content summary to be displayed?	0 = No 1 = Yes
StyleInfo	Inline style information for the generated text hyperlink.	Any string

Attribute	Description	Options
OrderBy, ASC/DESC	Order the hyperlinks by:	"Title" "DateModified" "DateCreated" "StartDate" "AuthorFname" "AuthorLname"
	Order the hyperlinks in ascending or descending order.	"Title,asc" "Title,desc"
ShowInfo	Information that will appear to the right of the hyperlink.	"DateModified" "DateCreated" "AuthorFname" "AuthorLname"
MaxNumber	Maximum number of summaries returned.	0 = Unlimited
Spare1	Not yet implemented	MUST BE ""
Spare2	Not yet implemented	MUST BE ""

List Summary Example

Shown below is an example of the ListSummary function in an ASP Web page.

```

<TR>
  <TD>
    <?PHP ecmListSummary ("Marketing\News",0,1,"",
      "Title,asc","DateModified",0,"",""); ?>
  </TD>
</TR>

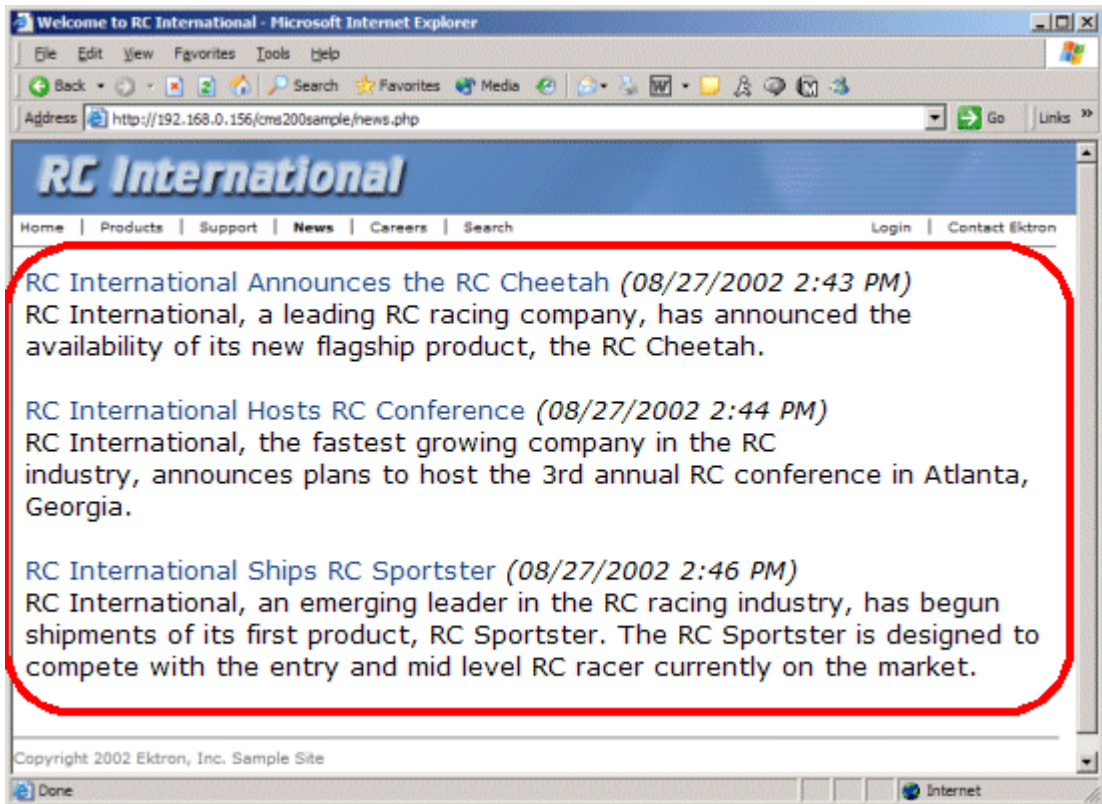
```

The previous example has the following properties:

- Displays summaries from the \Marketing\News folder
- Not recursive.
- The summaries will be shown
- No style was applied to the hyperlinks
- The results will be displayed in ascending order by title

- The date modified of each content block will be displayed
- Unlimited results will be shown.

Here is how the summaries will appear on the web page.



List Summary Variables

Listed below are all the different variables and descriptions about them.

Starting Folder

In this variable, you will need to specify which content folder you will want to get all the summaries from. There are two options for this variable:

Value	Description
"\"	By leaving the value blank, all the summaries from the root content folder will be displayed.
"Folder Path"	You can specify a specific folder to display the summaries from by inserting the folder path

Recursive

When you setup the search to be recursive, you are allowing the search to search all the sub-folders associated with the starting folder you had specified. You have two options for this variable:

Value	Description
1	Recursive - summaries in the starting folder's sub-folders will also be displayed
0	Non-recursive - only summaries in the starting folder will be displayed.

Show Summary

The show summary attribute will allow or disallow the summary to be displayed in the summary list. You have two options for this variable:

Attribute	Web Page View
0	<p>RC International Ships RC Sportster</p> <p>RC International Hosts RC Conference</p> <p>RC International Announces the RC Cheetah</p>

Attribute	Web Page View
1	<p>RC International Ships RC Sportster 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.</p> <p>RC International Hosts RC Conference RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Style Info

In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

Style Info	Web Page View
""	<p>RC International Ships RC Sportster 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.</p> <p>RC International Hosts RC Conference RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Style Info	Web Page View
"font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:100%;"	<p>RC International Ships RC Sportster</p> <p>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.</p> <p>RC International Hosts RC Conference</p> <p>RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah</p> <p>RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Order By

This variable will sort the summary results by different options. There are five different ways to sort the summaries

Value	Summaries sorted by
Title	title
DateModified	date modified
Date Created	date created
StartDate	Start Date assigned to the content block when created.
AuthorFname	authors' first name
AuthorLname	authors' last name

Ascending/Descending

This variable is an addition to the previous OrderBy variable. By default, the list of the summaries are descending, meaning they will follow the path of A-B-C, or 1-2-3. When set to ascending, they will order by Z-Y-X or 3-2-1. You will have the option to re arrange these to display either in ascending or descending order.

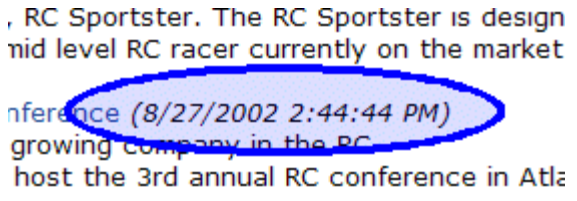
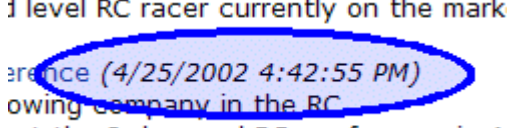
To do this, add the following to the OrderBy attribute:

Value	Description
asc, 1	This will order the Summaries in an ascending order
desc, 0	This will order the Summaries in a descending order. This is the default setting.

By using the ascending/descending option, you will have the option to set the order of content blocks from most recent to oldest, or vice versa. As well as ordering the list by the user's first names or last names in ascending or descending order.

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are four different options that you may choose from:

Value	Description	Web Page View
DateModified	Displays the date the content was last modified.	 <p>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</p>
DateCreated	Displays the date when the content was first created.	 <p>mid level RC racer currently on the mark ference (4/25/2002 4:42:55 PM) owing company in the RC st the 3rd annual RC conference in A</p>

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

Max Number

In this variable, you will determine how many summaries will be displayed on the page. You may enter any number into this field, and that number will be the amount of summaries that will be displayed. If you want all the summaries to be displayed, enter 0 (zero) and you will display unlimited results.

Max Number	Web Page View
0	<p>RC International Ships RC Sportster</p> <p>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.</p> <p>RC International Hosts RC Conference</p> <p>RC International, the fastest growing company in the RC industry, announces plans to host the 3rd annual RC conference in Atlanta, Georgia.</p> <p>RC International Announces the RC Cheetah</p> <p>RC International, a leading RC racing company, has announced the availability of its new flagship product, the RC Cheetah.</p>

Max Number	Web Page View
1	<p>RC International Ships RC Sportster</p> <p>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.</p>

Spare 1 & 2

There are two variable fields at the end that are currently not used. They are there for future scalability. When you insert a ListSummary tag into your template, you must leave those variables as ""

Single Summary Tag

The single summary tag is used to display the summary of one content block in the Ektron CMS200/300 Web site, instead of a list of all the summaries in a content folder. Seen below is the single summary tag:

```
<?php ecmSingleSummary( "ContentID",
                        ShowSummary,
                        "StyleInfo",
                        "ShowInfo",
                        "Spare1" );
?>
```

You can look at the single summary tag as just a simpler form of the List summary tag. The format for each of them is almost identical, but the single summary tag does not require as many variables to be passed.

A list and description of the variables are shown in this table:

Attribute	Description	Required?
Content ID	This is the ID number of the content block the summary information will come from.	Yes
ShowSummary	Do you want the content summary to be displayed?	Yes
StyleInfo	Inline style information for the generated text hyperlink.	No
ShowInfo	Information that will appear to the right of the hyperlink.	No
Spare1	Not yet implemented	MUST BE ""

Single Summary Example

Shown below is an example of the single summary function in an ASP Web page.

```
<TR>
  <TD>
```

```

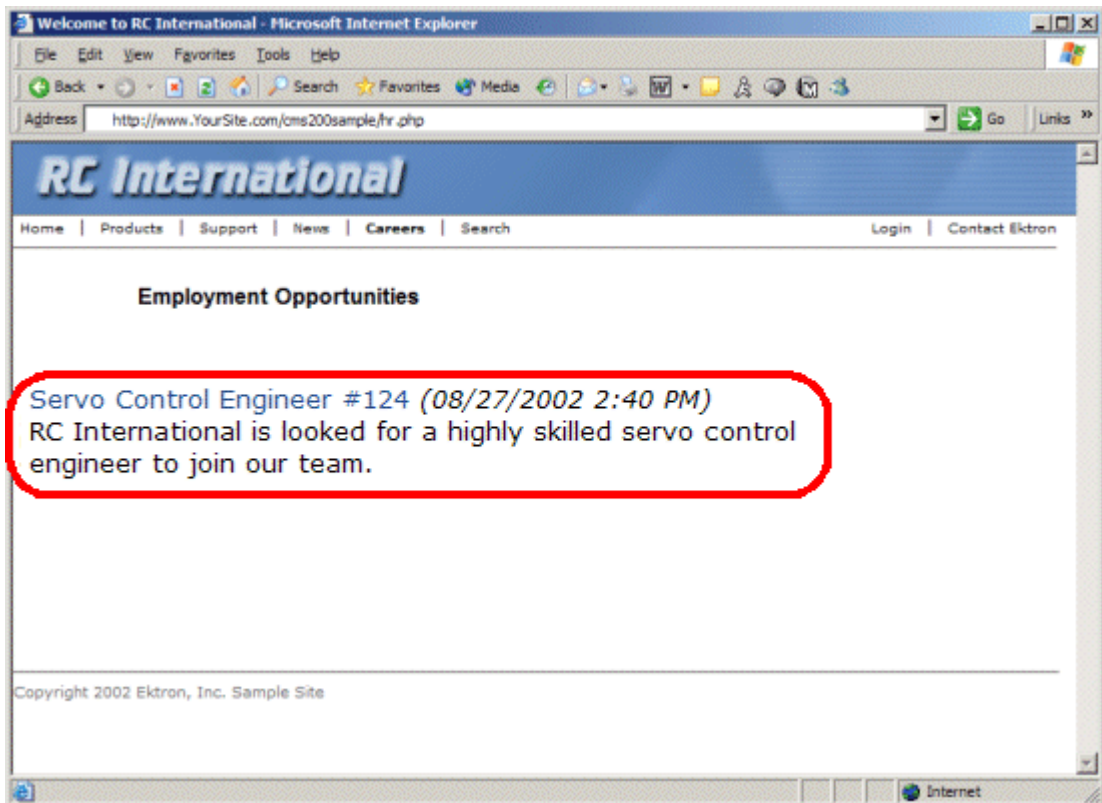
        <?PHP ecmSingleSummary ("14",1,"","DateModified",""); ?>
    </TD>
</TR>

```

The previous example had the following properties:

- Displays the summary for the content block with an ID=14
- The summary will be shown
- No style was applied to the hyperlinks
- The last date the content was modified will be displayed

Here is how the summary will appear on the web page.



Single Summary Variables

Listed below are all the different variables and descriptions about them.

Content ID

In this variable, you will need to specify which content block you will want the summary to be displayed from.

Show Summary

The show summary attribute will allow or disallow the summary to be displayed in the summary list. You have two options for this variable:

Value	Web Page View
0	Servo Control Engineer #124
1	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.

Style Info

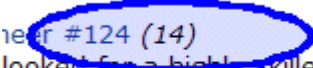
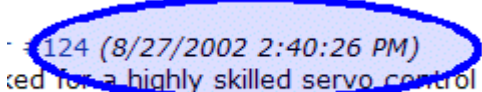
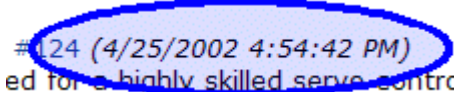
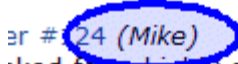
In this field, you may enter inline style information for the generated text hyperlinks displayed on the screen.

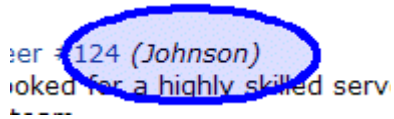
Style Info	Web Page View
""	Servo Control Engineer #124 RC International is looked for a highly skilled servo control engineer to join our team.

Style Info	Web Page View
font-family:arial; font-weight:bold; background-color:#cccccc; border:solid blue 1pt; margin:2px; width:100%;	<div>Servo Control Engineer #124</div> <p>RC International is looked for a highly skilled servo control engineer to join our team.</p>

Show Info

The show info field determines what sort of information is displayed to the right of the hyperlinked title. There are five different options that you may choose from:

Value	Displays the...	Web Page View
ID	ID number for the content block.	
DateModified	date the content was last modified.	
DateCreated	date when the content was first created.	
AuthorFname	first name of the user who last edited the content block.	

Value	Displays the...	Web Page View
AuthorLname	last name of the user who last edited the content block	

Spare 1

There is one variable fields at the end that are currently not used. This is here for future scalability. When you insert a single summary tag into your template, you must leave this variables as ""

Collection Function

The collection function is used to display the list of links that were created as a collection in the Ektron CMS200/300 Workarea. The ecmCollection function is highly customizable, allowing you to easily define how the collection data will be displayed on the Web page.

Shown here is the format of the ecmCollection function.

```
<?PHP ecmCollection( "id",
                    "displayFunction" ;
?>
```

Each of the attributes in the ecmCollection function are explained in the following table.

Attribute	Description
id	The id of the collection that you would like to be displayed is defined here.
displayFunction	Specify a function that is defined in the API that will define how the collection data is displayed on the Web page. The display function must also be defined in the ecmCollection function.

Collection Examples

The following are examples of the collection function in the Ektron CMS200/300 sample Web site. One example uses the "ecmNavigation" display function and the other uses the "ecmTeaser" display function.

The ecmNavigation and ecmTeaser display functions are sample display functions included with Ektron CMS200/300.

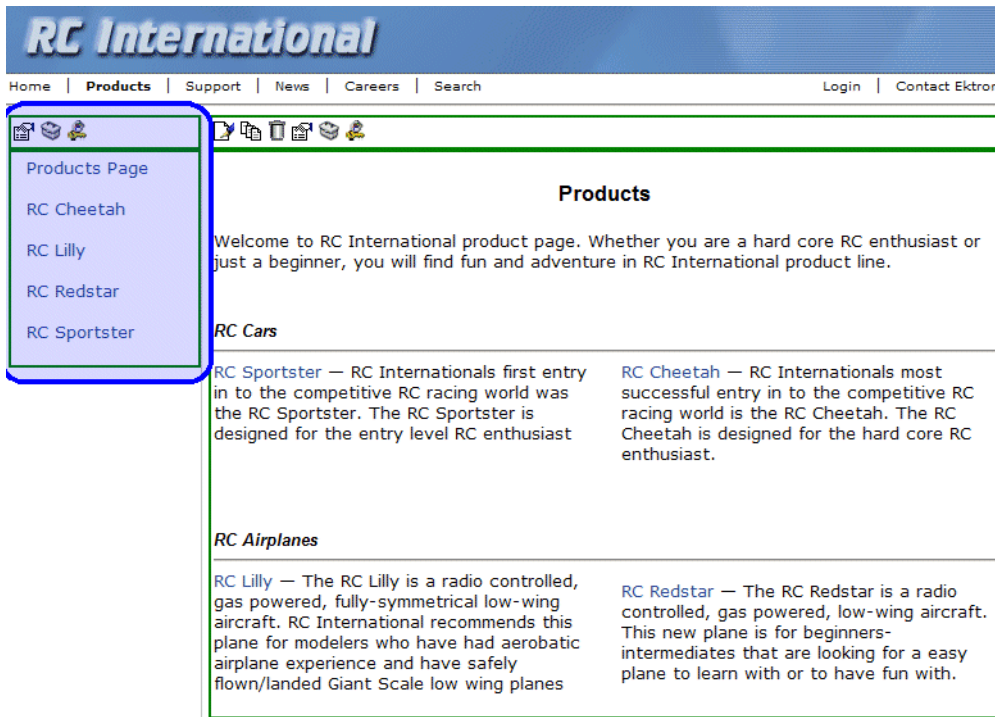
Example Using the ecmNavigation Display Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS200/300 sample Web site.

```
<?php ecmCollection(1, "ecmNavigation") ?>
```

In this example, the collection with an ID=1 is displayed, and the function “ecmNavigation” is used to define how the collection data is displayed on the Web page.

Shown here is how the collection would appear on the Web page.



As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used.

Shown below is the source code for the display function “ecmNavigation”

```
Function ecmNavigation($cInfo){
    global $html, $info;
    $html = "<table border=\"0\" cellspacing=\"0\" cellpadding=\"0\"
        width=\"100%\">";
    $mycount = $cInfo->count();
    for ($iloop=0; $iloop < $mycount; ++$iloop) {
        $info = $cInfo->Item($iloop+1);
```

```
$html = $html."<tr><td>&nbsp;&nbsp;&nbsp;<a href=\"";  
$html = $html.$info->Item("ContentLinks");  
$html = $html.">\"";  
$html = $html.$info->Item("ContentTitle");  
$html = $html."</a></td></tr><tr><td>&nbsp;&nbsp;&nbsp;</td></tr>\"";  
}  
$html = $html."</table>\"";  
return $html;  
}
```

As you can see, the `ecmNavigation` function is a simple PHP function that creates a table of the links that belong to the collection specified.

The `ecmNavigation` function also displays the title of the content blocks by using the “`info(“ContentTitle”)`” building block.

Example Using the `ecmTeaser Display` Function

The following is an example of a collection function being used as a navigation menu in the Ektron CMS200/300 sample Web site.

```
<?php ecmCollection (2, "MyDisplayFunction"); ?>
```

In this example, the collection with an ID=2 is displayed, and the function “`ecmTeaser`” is used to define how the collection data is displayed on the Web page.

Shown here is how the collection would appear on the Web page.



As stated earlier, the format of the collection on the Web page depends on the displayFunction that is used.

Shown below is the source code for the display function "ecmTeaser"

```
Function ecmTeaser($cInfo){
    global $html, $info;
    $html = "<table border=\"0\" cellspacing=\"0\" cellpadding=\"0\" width=\"100%\">";
    $mycount = $cInfo->count();
    for ($iloop=0; $iloop < $mycount; ++$iloop) {
        $info = $cInfo->Item($iloop+1);
        $html = $html."<tr><td><a href=\"";
        $html = $html.$info->Item("ContentLinks");
        $html = $html."\" >";
        $html = $html.$info->Item("ContentTitle");
        $html = $html."</a>&nbsp;(";
        $html = $html.date("m/d/Y g:i:s A", $info->Item("DateModified"));
        $html = $html.")</td></tr><tr><td>";
        $html = $html.$info->Item("ContentTeaser");
        $html = $html."</td></tr><tr><td>&nbsp;</td></tr>";
    }
    $html = $html."</table>";
    return $html;
}
```

As you can see, the ecmTeaser function is a simple ASP function that creates a table of the links that belong to the collection specified.

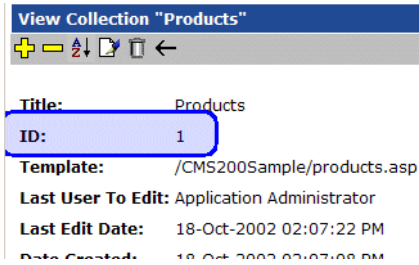
The ecmTeaser function also displays the:

- Content Title
- Date Modified
- Content Teaser

Collection Variables

ID

The collection ID is the ID number that is assigned to the collection when it is created in the workarea.



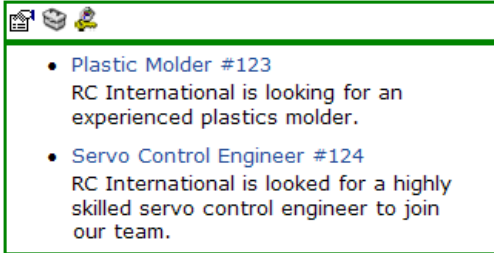

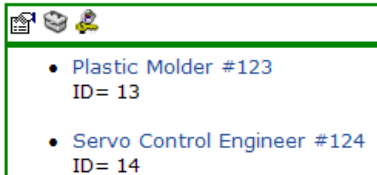

DisplayFunction

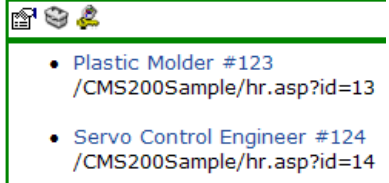



As explained earlier, the display function is a function that you create that will define how the collection data will be displayed on the Web page.


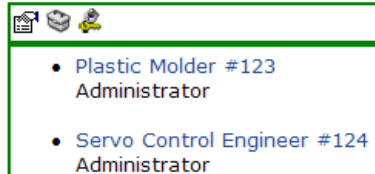

By using simple PHP scripting, you can create your own display functions, or use the three included functions.

Building Blocks

This section explains each of the possible building blocks that you may use when creating a custom display function.

Name	Description	Web Page View
info("ContentTeaser")	Displays the summary that has been created for the content block that is displayed in the collection data.	 <ul style="list-style-type: none"> • 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.
info("Comment")	Displays the comment added to the content block.	 <ul style="list-style-type: none"> • Plastic Molder #123 This is the historical comment for Plastics Molder #123 • Servo Control Engineer #124 This is the historical comment for Servo Control Engineer #124
info("ContentID")	Displays the ID number of the content block that is assigned to it by Ektron CMS200/300.	 <ul style="list-style-type: none"> • Plastic Molder #123 ID= 13 • Servo Control Engineer #124 ID= 14
info("ContentLinks")	Inserts the path of the content block with respect to your site root.	 <ul style="list-style-type: none"> • Plastic Molder #123 /CMS200Sample/hr.asp?id=13 • Servo Control Engineer #124 /CMS200Sample/hr.asp?id=14

Name	Description	Web Page View
info("ContentTitle")	Displays the title of the content block that is displayed with the collection data.	 <ul style="list-style-type: none"> Plastic Molder #123 /CMS200Sample/hr.asp?id=13 Servo Control Engineer #124 /CMS200Sample/hr.asp?id=14
info("DateModified")	Displays the date and time the content block was last modified.	 <ul style="list-style-type: none"> Plastic Molder #123 8/27/2002 2:38:34 PM Servo Control Engineer #124 8/27/2002 2:40:26 PM
info("DisplayDateModified")	Displays the date and time the content block was last modified using the date/time format specified in Setup.asp.	 <ul style="list-style-type: none"> Plastic Molder #123 27-Aug-2002 02:38:34 PM Servo Control Engineer #124 27-Aug-2002 02:40:26 PM
info("DisplayEndDate")	Displays the end date and time for the content block using the date/time format specified in Setup.asp.	
info("DisplayGoLive")	Displays the start date and time for the content block using the date/time format specified in Setup.asp.	 <ul style="list-style-type: none"> Plastic Molder #123 This content went live on: 30-Oct-2002 10:58 AM Servo Control Engineer #124 This content went live on: 29-Oct-2002 06:51 PM

Name	Description	Web Page View
info("EditorFname")	Displays the first name of the user who last edited the content block.	
info("EditorLname")	Displays the last name of the editor who last edited the content block.	
info("EndDate")	Displays the end date and time for the content block	
info("GoLive")	Displays the start date and time for the content block.	

Random Content

The random content function is used to randomly display the content of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

`<?php ShowRandomContent (id); ?>`

Where the "id" represents the ID number of the collection the content blocks, that will be displayed, belong to

View Collection "Products"



Title

RC Sportster
RC Cheetah
RC Lilly
RC Redstar

[More info](#)

Title:	Products
ID:	1
Template:	/CMS0000/sample/products.asp
Last User To Edit:	Application Administrator
Last Edit Date:	29-Oct-2002 09:36 PM
Date Created:	29-Oct-2002 09:34 PM
...	...

Random Summary

Similar to the random content function, the random summary function is used to randomly display the summary of a content block that belongs in a specified collection.

Displayed here is the format for the random content function

`<?php ShowRandomSummary (id); ?>`

Where the "id" represents the ID number of the collection the content blocks, that will be displayed, belong to



Title
RC Sportster
RC Cheetah
RC Lilly
RC Redstar

[More info](#)

Title	Products
ID:	1
Template:	/CMS00000sample/products.asp
Last User To Edit:	Application Administrator
Last Edit Date:	29-Oct-2002 09:36 PM
Date Created:	29-Oct-2002 09:34 PM
Created By:	Application Administrator

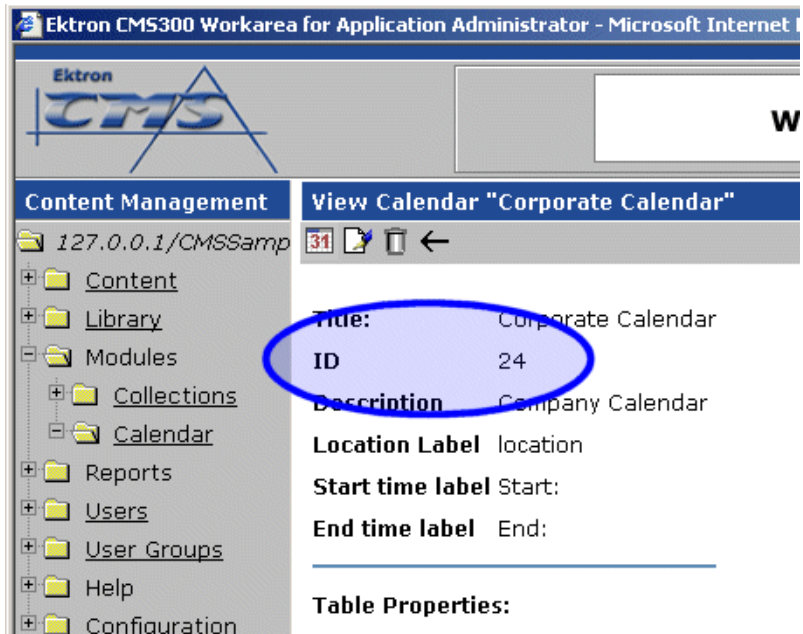
Event Calendar Function

The ecmEvtCalendar function is used to display an event calendar that has been created in the Ektron CMS200/300 Workarea.

Displayed here is the format for the event calendar custom function:

```
<?php ecmEvtCalendar(id); ?>
```

Where "id" represents the ID number of the event calendar in the Workarea.



Event Calendar Style Sheet

Before you insert an event Calendar into your Ektron CMS200/300 template, it is recommended that you make a reference to the style sheet, "\\CMS200/300Sample\\ekCalendar.css" in your template.

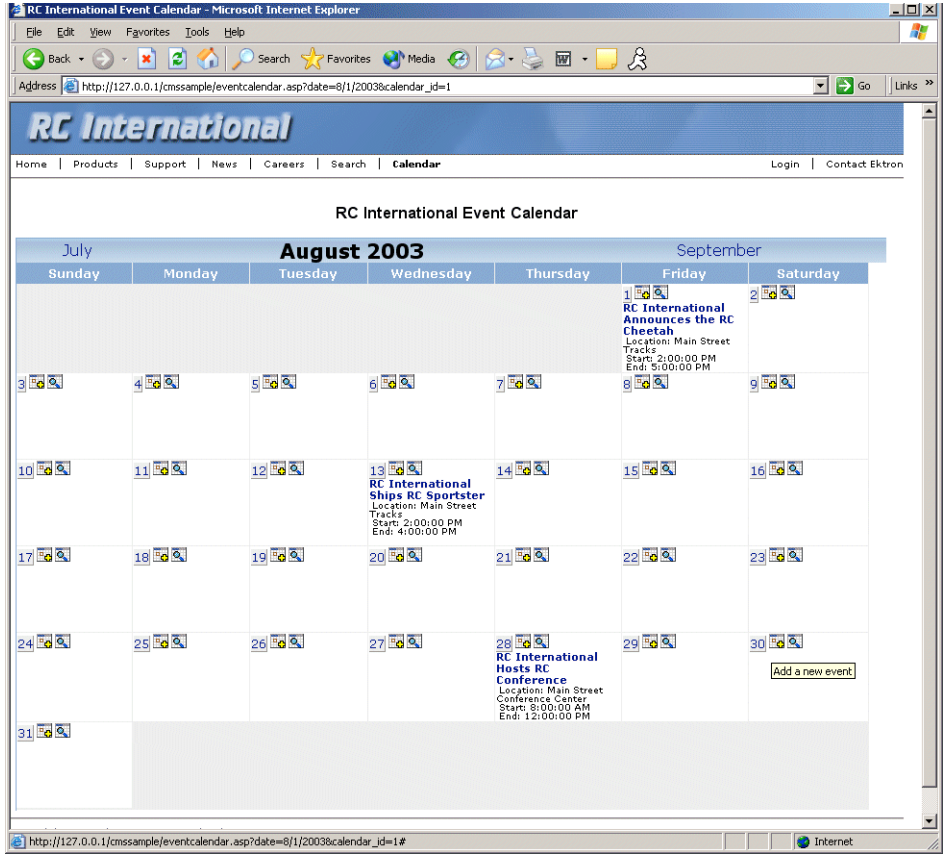
This style sheet contains styles that define the Calendar.

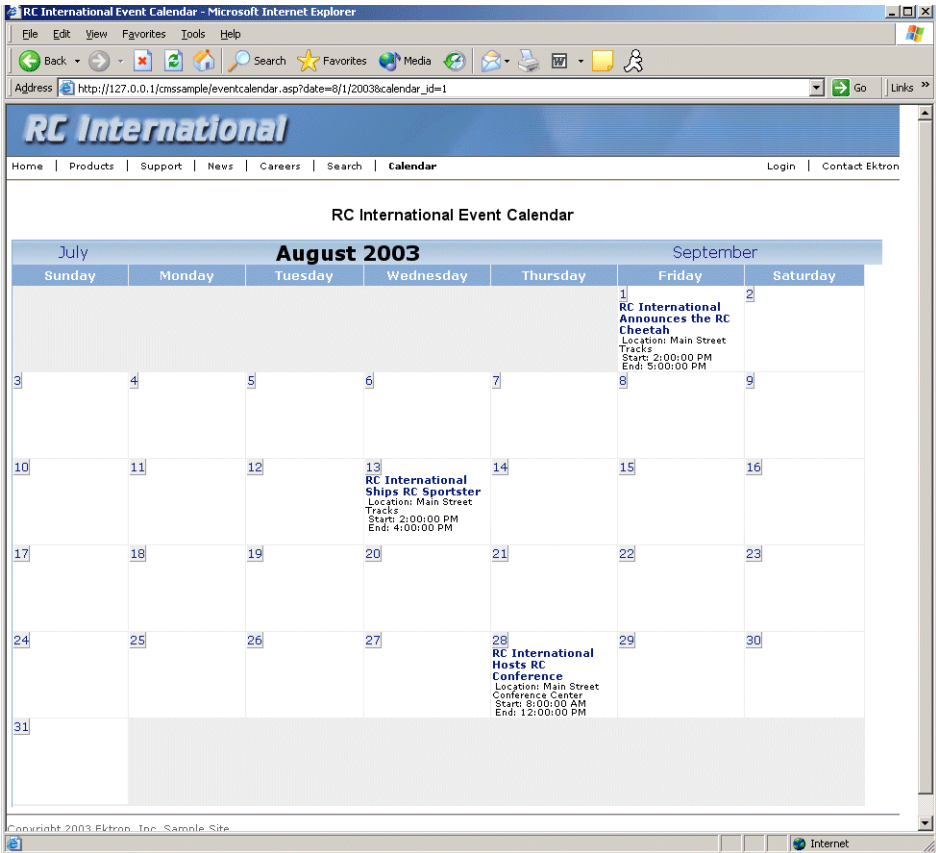
Event Calendar Sample

Displayed here is an example of the event Calendar in an Ektron CMS200/300 template:

```
<tr>
  <td>
    <?php ecmEvtCalendar(24); ?>
  </td>
</tr>
```

When this code is displayed in a browser, the Calendar with an ID-1 is displayed. Shown here is the Calendar displayed on the Web Page

If...	Calendar Display...
Logged in	

If...	Calendar Display...
Logged Out	

As you can see, when you are logged into the Web site, you are given options to add and view Calendar events. When logged out, only the calendar events are displayed.

Abbreviating Day Names

To further customize the event calendar function, you may specify in ApplicationAPI.php to display abbreviated day names (for example, Mon instead of Monday, Wed instead of Wednesday).

Displayed here is an example of the ecmEvtCalendar using abbreviated day names.

November		December 2003			Janu
Mon	Tue	Wed	Thu	Fri	
1	2	3	4	5	
3	9	10	11	12	

Configuring ApplicationAPI.php

To configure the ecmEvtCalendar function to display abbreviated day names, perform the following steps.

NOTE

Before editing ApplicationAPI.php, Ektron recommends to create a copy of it, referencing it in site_scripts_path.php, and making changes to the new file. See "Custom API" on page 371 for additional information.

1. Open the file that contains the ecmEvtCalendar function you want to edit.
2. Locate the ecmEvtCalendar function in the file.

```

333 Function ecmEvtCalendar($cID){
334     global $cookie_user_id,$MODULE_OBJ,$AppImgPath,$AppPath;
335     global $cmsPreview,$PopUpLoaded,$AppConfStr, $HTTP_SERVER_VARS, $cookie_si
336     //'check to see if we are logged in if we need some script
337     if ($cookie_user_id != 0){
338         if ($PopUpLoaded == 0) {
339             ecmGetPopupScript();
340         }
341     }
342     $CalendarObj = new COM($MODULE_OBJ);
343     $Arg = new COM("Scripting.Dictionary");
344
345     if((GetURL_param("ekM") != 0) And (GetURL_param("ekY") != 0) ){
346         $Arg->Add("Month" , GetURL_param("ekM"));
347         $Arg->Add("Year", GetURL_param("ekY"));
348     }
349     else{
350         $Arg->Add( "Month" , "");
351         $Arg->Add("Year", "");
352     }
353     $Arg->Add("ImagePath", $AppImgPath);
354     $Arg->Add("AppPath", $AppPath);
355     $Arg->Add("CallerID", $cookie_user_id);
356     $Arg->Add("Site", $cookie_site_id);
357     $Arg->Add("CalendarID", $cID);
358     $Arg->Add("WorkSpace", "");
359     $Arg->Add("QueryString", getkeyvalue($HTTP_SERVER_VARS, "QUERY_STRING"));
360     $Arg->Add("URLfile", getkeyvalue($HTTP_SERVER_VARS, "URL"));
361     $Arg->Add("PathInfo", getkeyvalue($HTTP_SERVER_VARS, "PATH_INFO"));
362     $Arg->Add("ServerName", getkeyvalue($HTTP_SERVER_VARS, "SERVER_NAME"));
363     $Arg->Add("PreviewMode", $cmsPreview);

```

3. Locate the line that reads

```

        "$Arg->Add("AbbreviateDayNames", false);"
51 $Arg->Add("PathInfo",getkeyvalue($HTTP_SERVER_VARS, "P
52 $Arg->Add("ServerName",getkeyvalue($HTTP_SERVER_VARS,
53 $Arg->Add("PreviewMode", $CMSPreview),
54 $Arg->Add("AbbreviateDayNames", false);
55 $CalView = $CalendarObj->ecmCalendar($AppCalFStr,$Arg)
56 echo($CalView);
57 //echo(ereg("eKM", getkeyvalue($HTTP_SERVER_VARS,"QUE

```

4. On that line, change “false” to “true”

This will force the ecmEvtCalendar to use abbreviated day names.

5. Save and close the file.
6. View the Web page that contains the updated function.

Form Function

The ecmFormBlock function displays an HTML form content block. For more information about HTML forms, see the Ektron CMS200/300 Administrator manual chapter “Working with HTML Forms.”

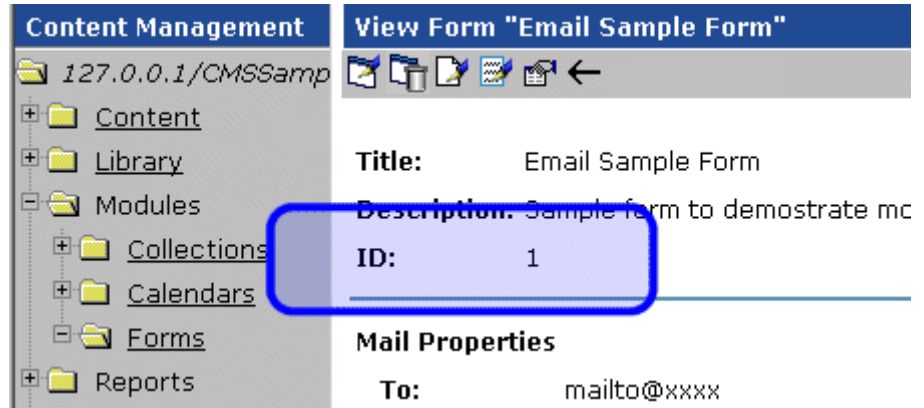
IMPORTANT!

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 CMS200/300 Administrator manual chapter “Managing Library Assets”, section “Updating Default Template for Multiple Quicklinks.”

Here is the format for the form custom function

`<?php ecmFormBlock(id); ?>`

The “id” argument represents the form’s ID number, visible when viewing the form from the View Form screen.



NOTE

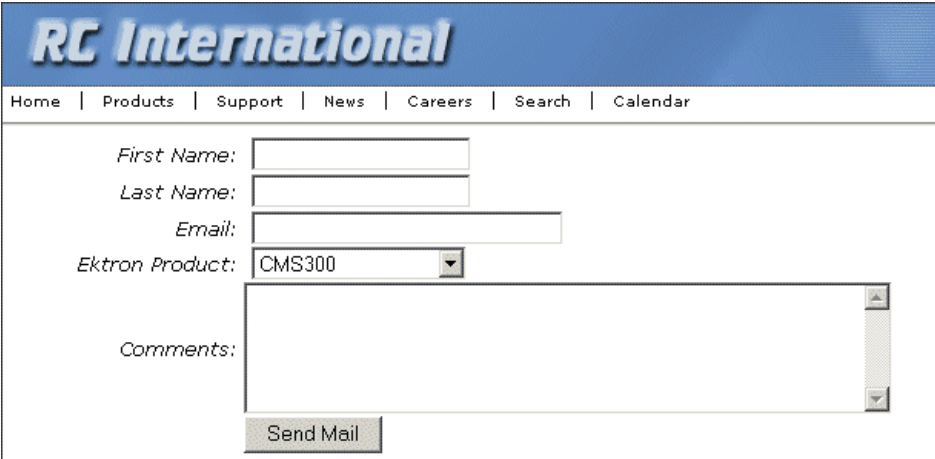
As an alternative, you can use the multipurpose function, which can display either a content block or a content block associated with a form. For more information, see “MultiPurpose Function” on page 363.

Here is an example of a form content block in an Ektron CMS200/300 template.

```
<tr>
    <td>
        <?php ecmFormBlock(1); ?>
    </td>
</tr>
```

```
</td>  
</tr>
```

When this code is read by a browser, the following is displayed.



The screenshot shows a web form for "RC International". The header is a blue bar with the company name in a stylized font. Below the header is a navigation menu with links: Home, Products, Support, News, Careers, Search, and Calendar. The form itself is on a white background and contains the following fields:

- First Name:** A text input field.
- Last Name:** A text input field.
- Email:** A text input field.
- Ektron Product:** A dropdown menu with "CMS300" selected.
- Comments:** A large text area with a vertical scrollbar.
- Send Mail:** A button located below the comments field.

When a visitor to your site submits a form, the form uses the form information to process the data.

MultiPurpose Function

The ecmMultiPurpose function displays either a content block or a content block associated with a form. As a result, you can use one template to display both types of content block.

Here is the format for the Multipurpose function

<?php ecmMultiPurpose(ID); ?>

To pass in a standard content block, pass the id value in the url parameter, as illustrated below.

`http://localhost/cms300sample/index.asp?id=2`

To pass in a Form content block, pass the ekfrm value in the url parameter, as illustrated below.

`http://localhost/cms300sample/index.asp?ekfrm=2`

NOTE For more information about passing the id value as a url parameter, see ["Dynamic Content Block"](#) on page 309.

If the id value or ekfrm value does not exist, a blank template appears.

If the url parameter is not passed in or recognized, the function calls ecmContentBlock and passes the DefaultID value as the id argument. If the DefaultID value is zero (0), ecmContentBlock is not called.

Metadata Tag

The metadata tag is used to display the metadata for one or more content blocks in the Ektron CMS200/300 Web site. Inserting this custom tag will allow you to generate the list of metadata that is edited by the authors, and insert it in the source of the Web page. Shown below is an example of what the metadata custom tag looks like.:

```
<?php ecmMetaData ("ItemList","Spare1"); ?>
```

A list and description of the variables are shown in this table:

Attribute	Description	Required?
ItemList	List of the content block ID numbers, and exclusions	No
Spare1	Not yet implemented	MUST BE ""

Creating an Item List

The ItemList for the ecmMetadata tag has the following format:

```
"[content block id; Exclusion list]...[content block id; Exclusion list]"
```

Attribute	Description	Required?
[(left bracket)	Delineates the beginning of a content block id/Exclusion List pair.	Yes
Content Block ID	A number that represents the content block you wish to retrieve the metadata from	Yes
; (semicolon)	Separator: Separates the content block id from the exclusion list	No (Yes if including an exclusion list)
Exclusion List	A comma delimited list of metadata you do not wish to be retrieved for this content block.	No

Attribute	Description	Required?
] (right bracket)	Delineates the end of a content block id/Exclusion List pair.	Yes

ecmMetadata Example

For this example, our administrator has defined the following metadata types for Ektron CMS200/300.

Name	Style	Type	Editable	Required	Remove Dups	Case Sensitive
Title	n/a	HTML	Yes	Yes	No	n/a
Keywords	Name	Meta	Yes	Yes	Yes	No
Content-Type	http-equiv	Meta	No	Yes	Yes	n/a

Our example Web page has two content blocks displayed. Content block 1, the company's home page introduction text, and content block 2, a sidebar containing the lead story for the week. The metadata entered by your content editors for these two content blocks are:

Content Block 1:

Title – "Welcome to Our Home Page"

Keywords – "Home; page; company; Welcome"

Content-Type -- "text/html; charset=iso-8859-1"

Content Block 2:

Title – "Announcing our New Product!"

Keywords – "Announce; company; product; widget"

Content-Type -- "text/html; charset=iso-8859-1"

To include both content blocks metadata in the web page our ecmMetadata call would look like:

```
<html>
  <head>
    <?php ecmMetadata ("[1][2;Title,Content-Type]", ""); ?>
    (rest of head)
  </head>
  <body>
    (rest of web page document)
```

```
</body>
</html>
```

In the above example the ecmMetadata call is contained in the <head> of the HTML document (since all HTML metadata is required to exist there). Once the Web page is viewed the Web page source will contain the following information:

```
<html>
  <head>
    <Title>Welcome to Our Home Page</Title>
    <meta name="Keywords" content= Home; page; company; Welcome; Announce;
    product; widget">
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
    (rest of head)
  </head>
  <body>
    (rest of web page document)
  </body>
</html>
```

The ecmMetadata constructed these three HTML tags based on the administrator defined parameters and the ecmMetadata call.

The metadata tag was constructed in the page because:

- **[1]** - Will display all the metadata content for the content block ID=1. In this example, the title, keywords, and content-type were defined, so they were all displayed as:

```
<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome; Announce;
product; widget">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

- **[2;title,content-type]** - Will display all the metadata content for content block ID=2, EXCEPT for its title and content-type definitions.

```
<Title>Welcome to Our Home Page</Title>
<meta name="Keywords" content= Home; page; company; Welcome; Announce;
product; widget">
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

The next section will provide you with a few more examples

ecmMetadata Variables

Listed below are all the different variables and descriptions about each of the them.

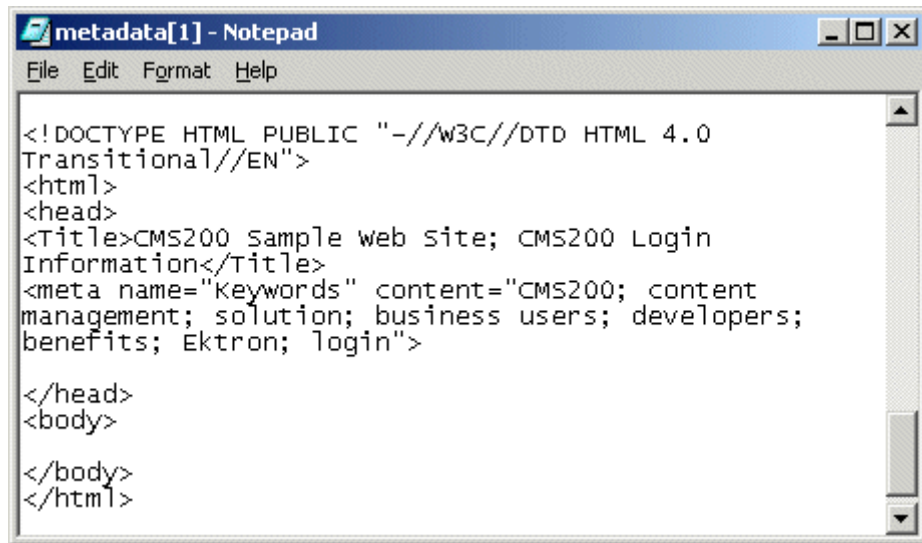
Item List

In this variable of the ecmMetadata tag, you will need to specify two values:

Value	Description
ID	Specify the content block ID to specify which content block will have the metadata displayed.
Exclusion List	After you specify the ID for the content block you want to display the metadata, you can also specify the metadata that you wish to be excluded.

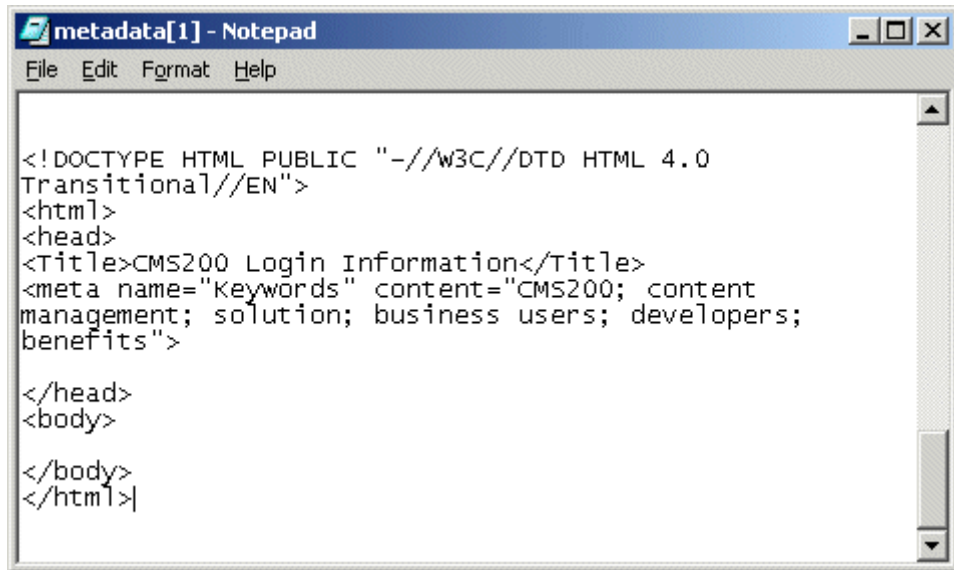
Some examples might be:

```
<?PHP ecmMetaData ("[1][2]","") ?>
```



In this example, all the metadata from the content block ID=1 and 2 will be displayed.

```
<?PHP ecmMetaData ("[1;title][2;keywords]","") ?>
```



```

metadata[1] - Notepad
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 Login Information</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

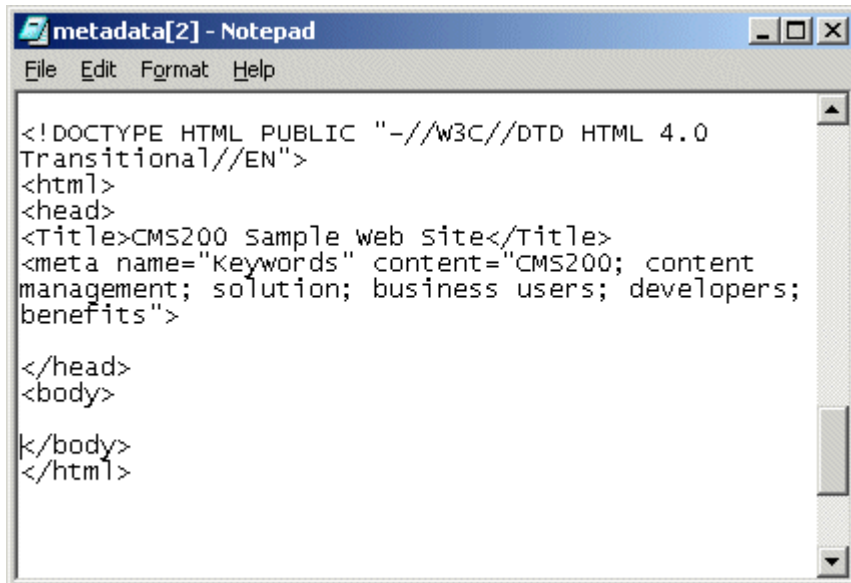
</head>
<body>

</body>
</html>

```

In this example, all the metadata from content block ID=1 will be displayed EXCEPT for the title. Also, all the metadata for content block ID=2 will be displayed Except for its keywords.

```
<?PHP ecmMetaData ("[1][2;title,keywords]","") ?>
```



```

metadata[2] - Notepad
File Edit Format Help

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<Title>CMS200 sample web site</Title>
<meta name="Keywords" content="CMS200; content
management; solution; business users; developers;
benefits">

</head>
<body>

</body>
</html>

```

In this example, all the metadata from content block ID=1 will be displayed, and all the metadata from content block ID=2 will be displayed except for its title and keywords.

Spare 1

There is one variable fields at the end that are currently not used. This is here for future scalability. When you insert a single metadata tag into your template, you must leave this variables as "".

Show Site Language Function

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



(For more information, see the “Multi-Language Support “ chapter of the **Ektron CMS200/300** Administrator manual.)

Below is an example of the Show Site Language custom tag.

<?php ecmShowSiteLanguage (FieldName); ?>

You can place this function in any location of any page on your site. By default, it appears in the lower right corner of the home page.

The variable is described in this table.

Variable	Description
FieldName	<p>An optional argument. You can leave it blank and allow the drop-down list to be populated by the available languages for your Web site. For example:</p> <pre><%=ecmShowSiteLanguage('')%></pre> <p>Or you can pass in an argument such as:</p> <pre><%=ecmShowSiteLanguage('mysitelanguage')%></pre>

Custom API

As you have seen, the formatting for the custom functions used in Ektron CMS200/300 for PHP are defined in the file `ApplicationAPI.php`. As a developer, you may make changes to this file to further enhance how the custom functions appear on your Web site.

However, Ektron recommends creating a copy of the file, and renaming it to an arbitrary name, for example, `MyCustomAPI.php`, and including that file when creating Ektron CMS200/300 templates. This ensures that your custom API will not be overwritten if you upgrade to another version of Ektron CMS200/300.

Dreamweaver Extension Support

Ektron CMS200/300 has a custom Macromedia® Extension Package which allows you to not only create your templates in Dreamweaver® or UltraDev™, but also to insert the custom functions through Dreamweaver®.

To do this, all you need is Ektron CMS200/300, Dreamweaver®/UltraDev™, and to keep reading.

Requirements

To use the Dreamweaver extension supplied by Ektron, you will need to have Dreamweaver 4.0 or higher installed, as well as Ektron CMS200/300.

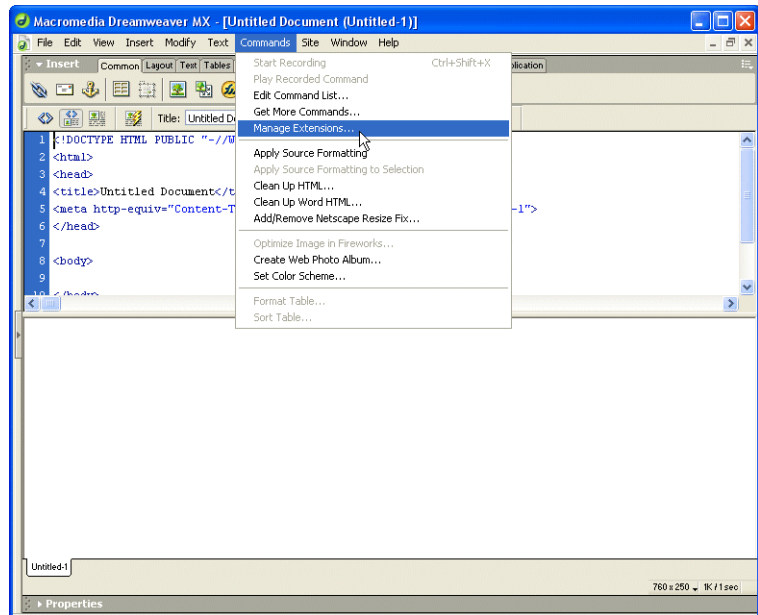
Installing the Extension

Before you can insert the Ektron CMS200/300 custom functions through Dreamweaver®, you must install the extension. Listed below are the steps to install the extension.

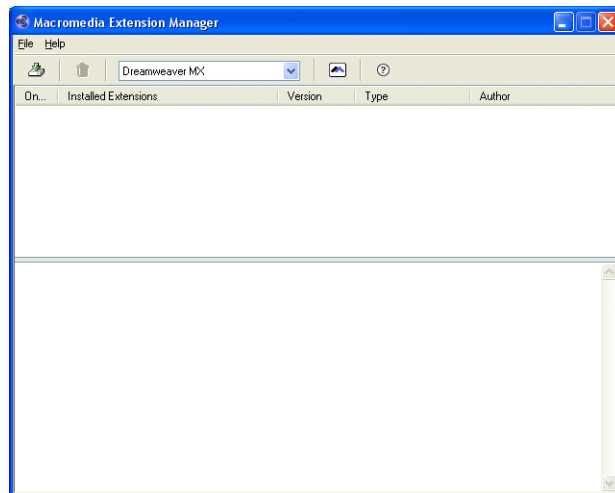
NOTE For more information about installing an extension, refer to your Dreamweaver® Manual.

To Install the Extension

1. Begin by following the path in Dreamweaver:
Commands > Manage Extensions...

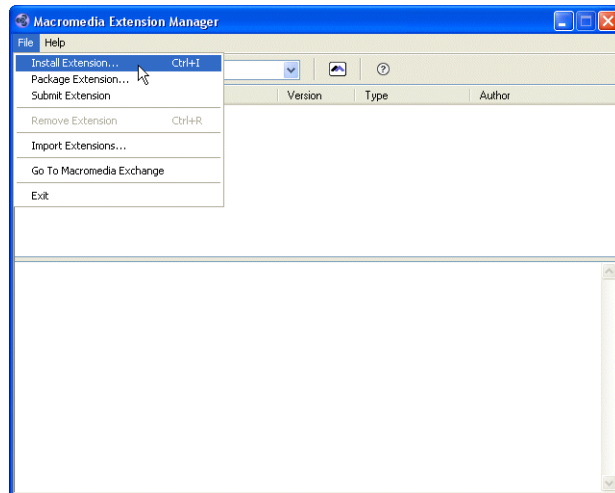


The Macromedia Extension Manager is displayed.

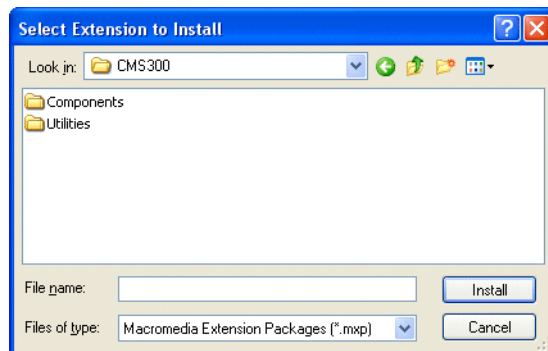


2. From this dialog box, follow the path:

File > Install Extension...

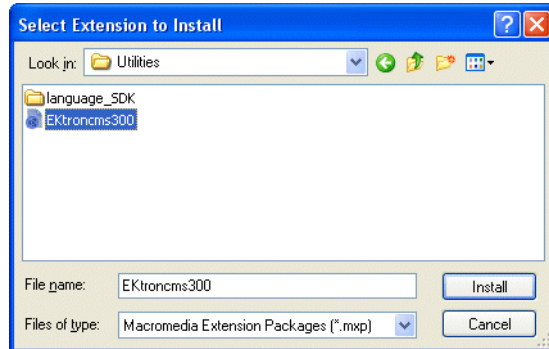


The Select Extension to Install window opens.



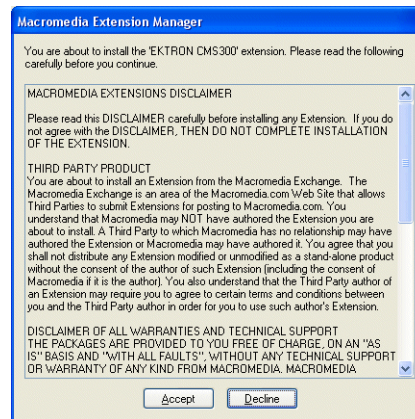
3. You will need to locate the Macromedia® Extension Package (.mxp) file that corresponds to Ektron CMS200/300. By default, the file gets installed to the following directory:

C:\Program Files\Ektron\CMS200/300\Utilities



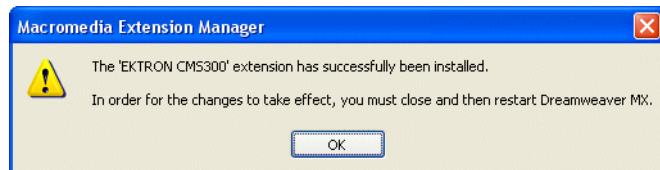
4. When you have selected the file, click the **Install** button.

The installation will begin with the Macromedia® Extensions Disclaimer displayed.



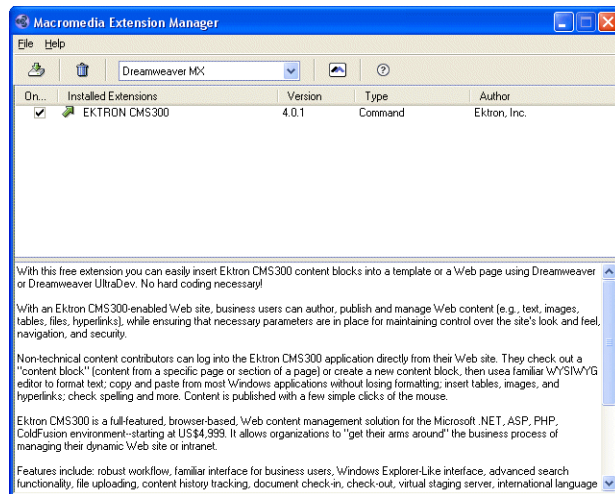
5. Click **Accept** to accept the disclaimer and install the extension.

The extension will then begin to install. Once installed, you will receive a confirmation box telling you that the extension was successfully installed.



6. Click **OK** to continue.

The Macromedia Extension Manager is again displayed, but with the added extension.



7. If needed, exit Dreamweaver®, then start it back up and you will then have the ability to use the Ektron Ektron CMS200/300 Extension Pack.

Refer to ["Using the Ektron CMS200/300 Extension"](#) on page 383 for more information on using Dreamweaver® to create and edit templates for Ektron CMS200/300.

Setup

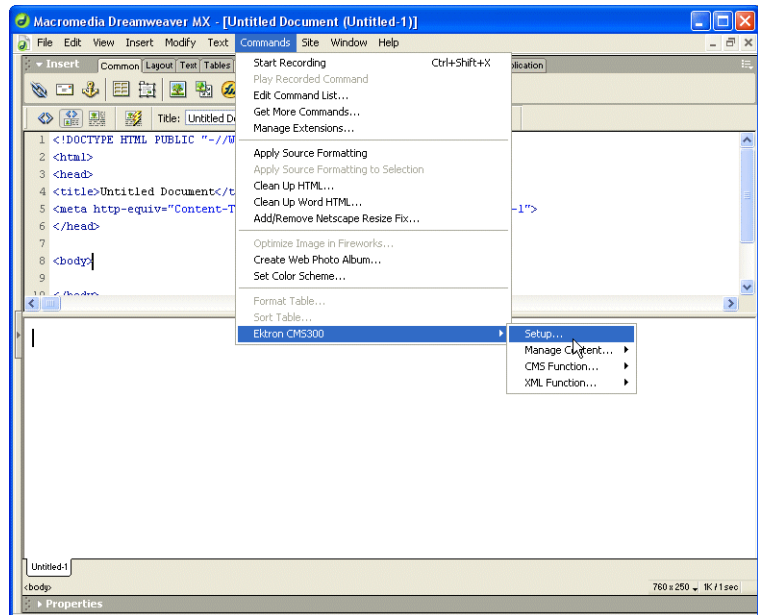
Before you can use Dreamweaver®/UltraDev™ to create and edit your templates for Ektron CMS200/300, you should configure the Ektron CMS200/300 Extension. Listed below are the configurative options for the extension in Dreamweaver:

- Username
- Password
- Domain
- Server File
- Server
- Work Online
- Include
- Check for Include

To change these options

1. Follow the path:

Commands > Ektron CMS200/300 > Setup...



The Ektron CMS200/300 Setup dialog box is displayed on the screen.

 A screenshot of the 'Ektron CMS200/300 Setup' dialog box. It contains the following fields and options:

- Username:
- Password:
- Domain:
- Server File:
- Server: ☒ ASP ☐ ASP.NET ☐ CF ☐ PHP
- ☒ Work Online (check is default)
- Include:
- ☒ Check for include (check is default)
- Buttons: Update Setup, Cancel, Help

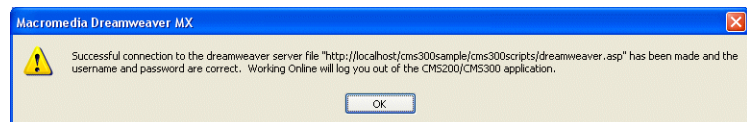
2. Use the following table to assist you with updating your setup information.

Field	Description
Username	Valid username of a user of your Ektron CMS200/300 Web site. When using the extension, the username will be used to log into the database to retrieve information about content blocks and other Web site information.
Password	Enter the password for the user into this field.

Field	Description
Domain	Only required in Active Directory has been enabled for your Web site. Enter the domain corresponding to the username entered.
Server File	Specify the location and file name for the dreamweaver.asp file needed for the extension to properly work and make connection with your Ektron CMS200/300 database.
Server	Select the type of application server you are creating templates for.
Work Online	By default, you will be checked off to work online. If you decide to work offline, you can no longer get data from the database in regards to the drop down list content block names for assigning content to a static content block.
Include	<p>Since each Ektron CMS200/300 template requires an include file to point to the API with the definitions for the custom functions, you may specify the include file in this text box.</p> <p>If you have selected a server other than the default ASP server, be sure that you update this text box to use the correct include file.</p> <hr/> <p>Note: If you are using ColdFusion as your application server, you will not need to include any file on the .cfm templates.</p> <hr/>
Check for Include	<p>To avoid having two include files on the same template which could cause the template to not properly display in the browser, the check for include option will check the template for the include file specified in the include field.</p> <p>If the template does not have the include file referenced, then upon insertion of a custom function in Dreamweaver, the include file will automatically be added to the template.</p>

- Once you have completed editing your setup information, click Update Setup to save and update the settings.

A message similar to the following is displayed;



NOTE

Receiving a message other than this is most like due to an incorrect server file, username, or password. Re check all your settings and try again.

4. Click **OK** to continue.

The setup dialog box closes, and you are now ready to create templates using Macromedia Dreamweaver.

Using the Ektron CMS200/300 Extension

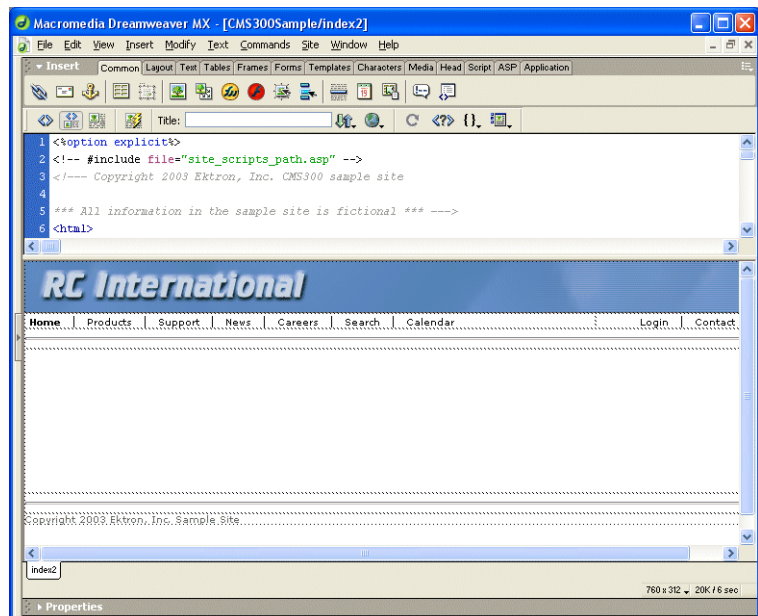
Now that the Macromedia® Extension Package has been installed for Dreamweaver, you can create and edit templates for Ektron CMS200/300.

To explain how to use this, we will begin by using a template that is in the Ektron CMS200/300 installation, index.asp, but with all of the custom tags stripped out.

NOTE You can also create your own templates instead of using the ones in the Ektron CMS200/300 installation.

In Dreamweaver, begin by opening the file:

C:\inetpub\wwwroot\CMS200/300Sample\index.asp
















NOTE





The custom ASP function that is inserted into this template by default has been stripped out.

Custom Function Symbols

During the process of inserting functions, 17 symbols are used to represent the custom display functions. The following table explains each symbol.

Symbol	Represents	More Information
	Login/Logout button	"Insert a Login Tag..." on page 403
	Partner Login	"Inserting a Partner Login Tag" on page 405
	Static content block.	"Inserting a Static Content Block" on page 406
	Dynamic content block	"Inserting a Dynamic Content Block" on page 408
	Search tag	"Inserting a Search Function" on page 415
	Search display function	"Inserting a Search Display Function" on page 418

Symbol	Represents	More Information
	List Summary	"Inserting a List Summary Function" on page 411
	Single Summary	"Inserting a Single Summary Function" on page 420
	Metadata	"Inserting a Metadata Function" on page 413
	Collection	"Inserting a Collection Tag" on page 410
	Collection XSLT <u>Not available with CMS200</u>	
	Event Calendar	"Inserting an Event Calendar Function" on page 425
	XML Content Block <i>Used to insert an XML content block with an XSLT other than the default applied to it.</i> <u>Not available with CMS200</u>	

Symbol	Represents	More Information
	Form Content Block	"Inserting a Form Content Function" on page 426
	Metadata XSLT <u>Not available with CMS200</u>	
	Random Content	"Inserting a Random Content Function" on page 422
	Random Summary	"Inserting a Random Summary Function" on page 423

Ektron CMS200/300 Commands Menu

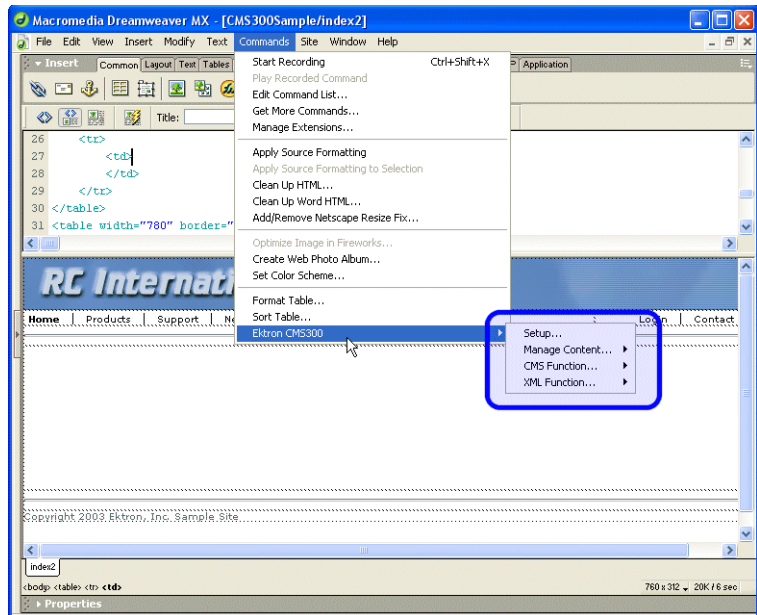
The following section explains how to locate Ektron CMS200/300 commands in Dreamweaver.

To locate the command menu in Dreamweaver, perform the following steps:

1. Open Dreamweaver.
2. In the top menu, follow the path:

Commands > Ektron CMS200/300

The Ektron CMS200/300 commands menu is displayed.



3. Click on the command for the task you would like to perform.
- The following table explains each of the commands:

Command	Description	More Information
Setup	Opens Ektron CMS200/300 Setup information dialog for Dreamweaver.	"Setup" on page 379
Manage Content...		
Edit Content...	Allows you to edit Ektron CMS200/300 content within Dreamweaver.	"Edit Content" on page 391
Save Content...	Allows you to save content created in Dreamweaver to Ektron CMS200/300	"Save Content" on page 393
Create Folder...	Allows you to create a new Ektron CMS200/300 content folder within Dreamweaver.	"Create Folder" on page 399

Command	Description	More Information
CMS Function...		
Insert Login Tag...	Inserts an Ektron CMS200/300 login function into the template.	"Insert a Login Tag..." on page 403
Insert Partner Login Tag...	Inserts an Ektron CMS200/300 partner login function into the template.	"Inserting a Partner Login Tag" on page 405
Insert Content Tag...	Inserts an Ektron CMS200/300 content block function into the template.	"Inserting a Static Content Block" on page 406
Insert Dynamic Tag...	Inserts an Ektron CMS200/300 dynamic content block placeholder function into the template.	"Inserting a Dynamic Content Block" on page 408
Insert Collection Tag...	Inserts an Ektron CMS200/300 collection function into the template.	"Inserting a Collection Tag" on page 410
Insert List Summary Tag...	Inserts an Ektron CMS200/300 list summary function into the template.	"Inserting a List Summary Function" on page 411
Insert Metadata Tag...	Inserts an Ektron CMS200/300 metadata function into the template.	"Inserting a Metadata Function" on page 413
Insert Search Tag...	Inserts an Ektron CMS200/300 search function into the template.	"Inserting a Search Function" on page 415
Insert Search Display Tag...	Inserts an Ektron CMS200/300 search display function into the template.	"Inserting a Search Display Function" on page 418
Insert Single Summary Tag...	Inserts an Ektron CMS200/300 single summary function into the template.	"Inserting a Single Summary Function" on page 420
Insert Random Content Tag...	Inserts an Ektron CMS200/300 random content function into the template.	"Inserting a Random Content Function" on page 422
Insert Random Summary Tag...	Inserts an Ektron CMS200/300 random summary function into the template.	"Inserting a Random Summary Function" on page 423
Insert Calendar Tag...	Inserts an Ektron CMS200/300 event calendar function into the template.	"Inserting an Event Calendar Function" on page 425

Command	Description	More Information
Insert Form Tag...	Inserts an Ektron CMS200/300 form content block into the template.	"Inserting a Form Content Function" on page 426
XML Function... (not available with CMS200)		
Insert Content XSLT Tag...	<p>Inserts an XML content block with your choice of XSLT in the template.</p> <hr/> <p>NOTE: Used to insert an XML content block with an XSLT other than the default applied to it.</p> <hr/>	
Insert Meta XSLT Tag...	Inserts an Ektron CMS200/300 content block with an XSLT applied to it forcing it to only display Metadata information.	
Insert Collection XSLT Tag...	Inserts an Ektron CMS200/300 content block with an XSLT applied to it forcing it to only display summary information.	

Each of these functions are explained in the following sections.

Inserting Custom Functions

The following section explains the steps required to insert each of Ektron CMS200/300's functions into your Web templates.

Manage Content...

Making the process of setting up and managing your Web site, the Ektron CMS200/300 Dreamweaver extension allows you to perform content related tasks directly from Macromedia Dreamweaver saving you time and effort.

Through Dreamweaver, you can perform the following tasks:

- Add new content blocks to Ektron CMS200/300
- Edit existing Ektron CMS200/300 content block
- Create new content folders

The following sections explain how to perform these tasks.

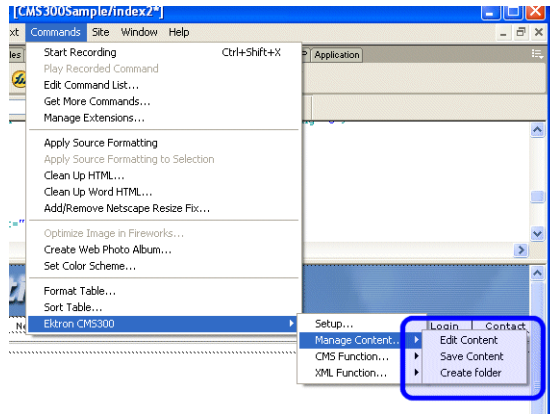
NOTE The user specified in the setup information for the extension will only be able to perform tasks they have permissions to in Ektron CMS200/300.

Accessing Manage Content Commands...

To access the CMS Manage commands menu, follow this path:

Commands > Ektron CMS200/300... > Manage Content...

A list of all CMS Manage Content commands that can be performed is displayed.



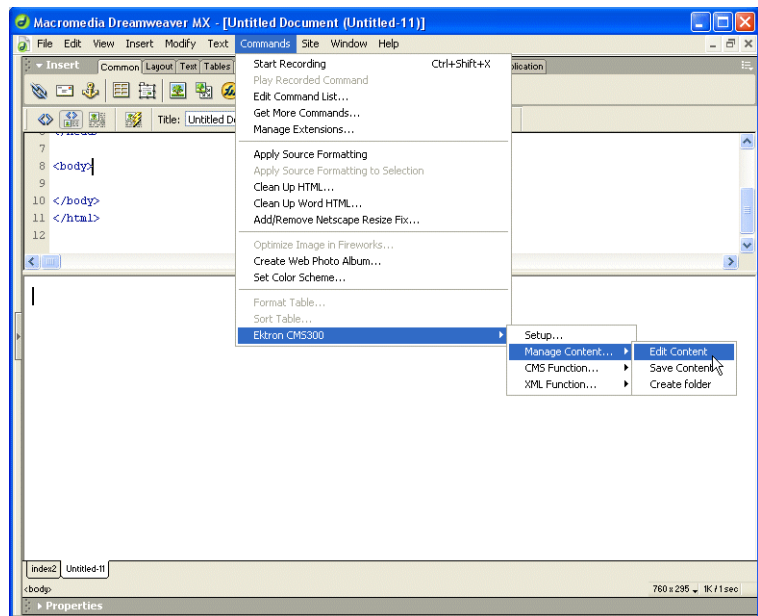
The following sections explain how to use each of the available manage content commands.

Edit Content

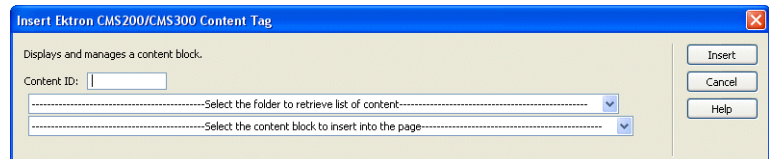
The edit content command allows you to get a content block from the Ektron CMS200/300 Web site, and edit it within Dreamweaver.

To edit a content block within Dreamweaver, perform the following steps:

1. Create a new HTML document in Dreamweaver.
2. In the Ektron CMS200/300 Manage Content menu, click on Edit Content.

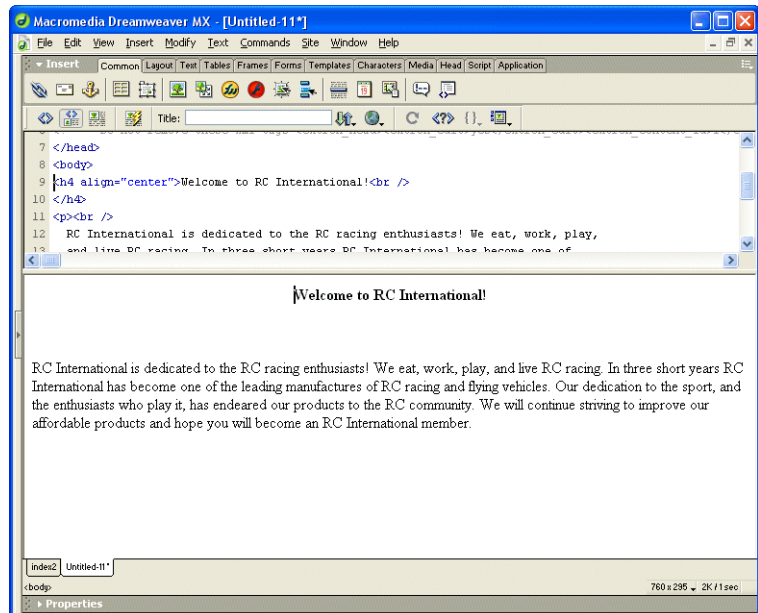


The Insert Ektron CMS200/300 Content tag dialog box is displayed.



3. Choose the Ektron CMS200/300 content block you would like to edit by:
 - Typing the ID number of the content
 - Using the drop down lists to locate the content folder and content block

4. Click **Insert** when you have chosen the content block.
The content block is inserted into Dreamweaver.



5. Make the necessary changes to the content block using Dreamweaver's editing capabilities.
See ["Saving Existing Ektron CMS200/300 Content" on page 396](#) for information about saving the content back to Ektron CMS200/300.

Save Content

Content that has been created, or edited, in Dreamweaver can be saved to your Ektron CMS200/300 Web site by using the Save Content command.

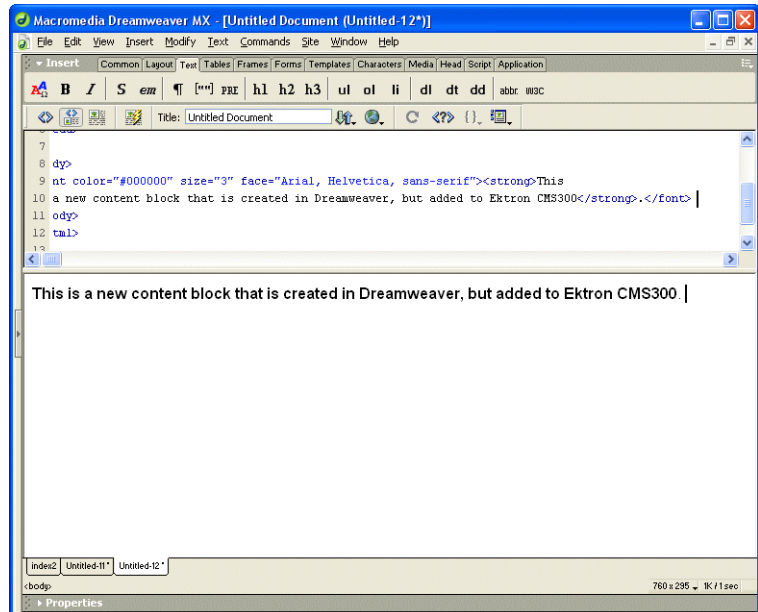
The following section explains how to save content created, and edited, to Ektron CMS200/300.

Saving New Content

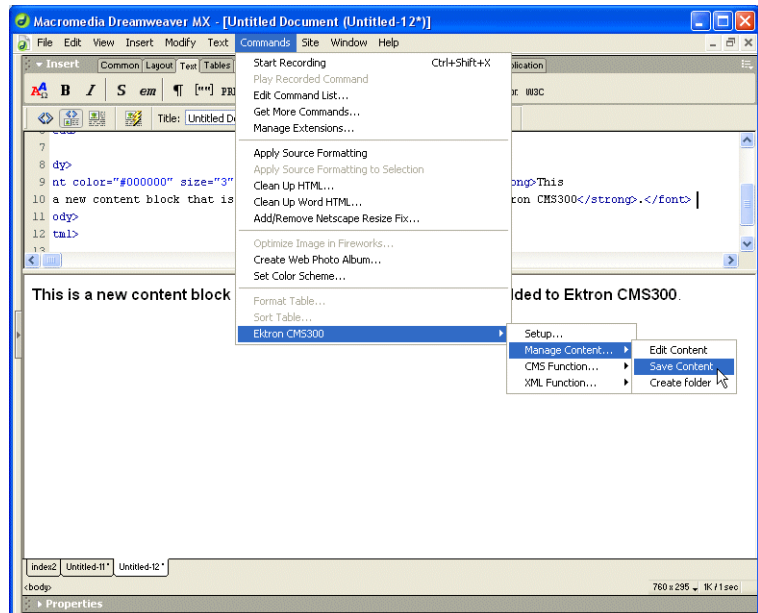
Content can be created in Dreamweaver, then saved to Ektron CMS200/300.

To save a content block that was created in Dreamweaver, perform the following steps:

1. If you haven't already, create the new content in Dreamweaver.



2. In the Ektron CMS200/300 Manage Content menu, click on **Save Content**.



The Save Ektron CMS200/300 Content dialog box is displayed.

- Using the following table, enter the necessary information to successfully fill out the Save Content form.

Field	Description
Title	Type a unique title for the content block.
Comment	Enter a comment for the content block.
Start Date	<p>If desired, type in a start date for the content block.</p> <hr/> <p>NOTE: Date format MUST match what is set in /cms300scripts/setup.asp. By default, this is dd-mmm-yyyy hh:mm AM/PM (for example, 24-Dec-2005 10:15 AM)</p> <hr/>
End Date	<p>If desired, type in an end date for the content block.</p> <hr/> <p>NOTE: Date format MUST match what is set in /cms300scripts/setup.asp. By default, this is dd-mmm-yyyy hh:mm AM/PM (for example, 24-Dec-2005 10:15 AM)</p> <hr/>
Content Folder	Select the content folder you would like to save the new content block to.

Displayed here is an example of this form filled out.

4. Click **Save** to save and add the new content block the selected content folder in Ektron CMS200/300.

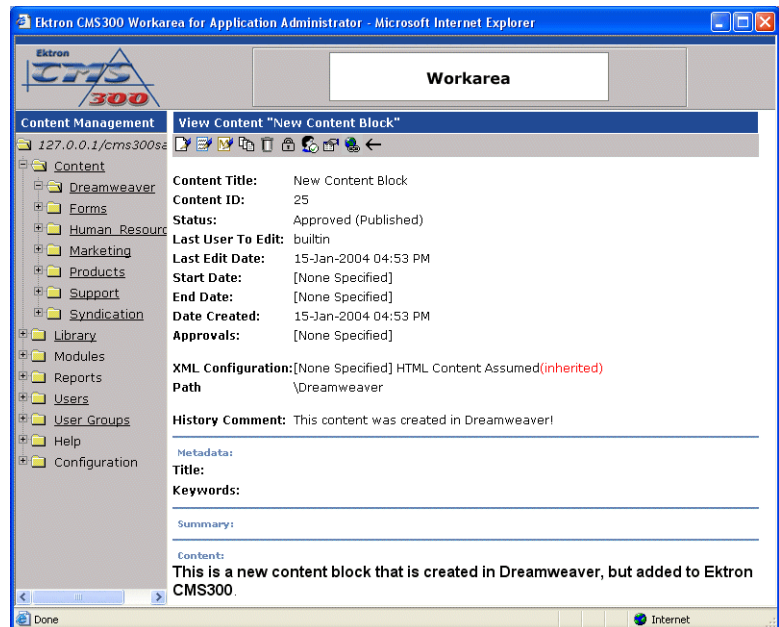
The content block is added, and the following confirmation message is displayed.



NOTE

If you receive a message other than the one displayed above, check your settings and try again.

Displayed here is the new content block in the Ektron CMS200/300 Workarea.

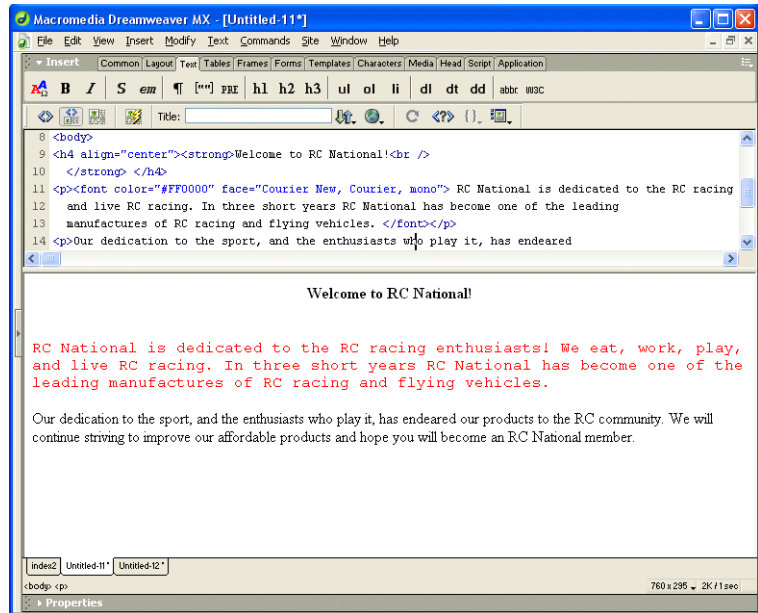


Saving Existing Ektron CMS200/300 Content

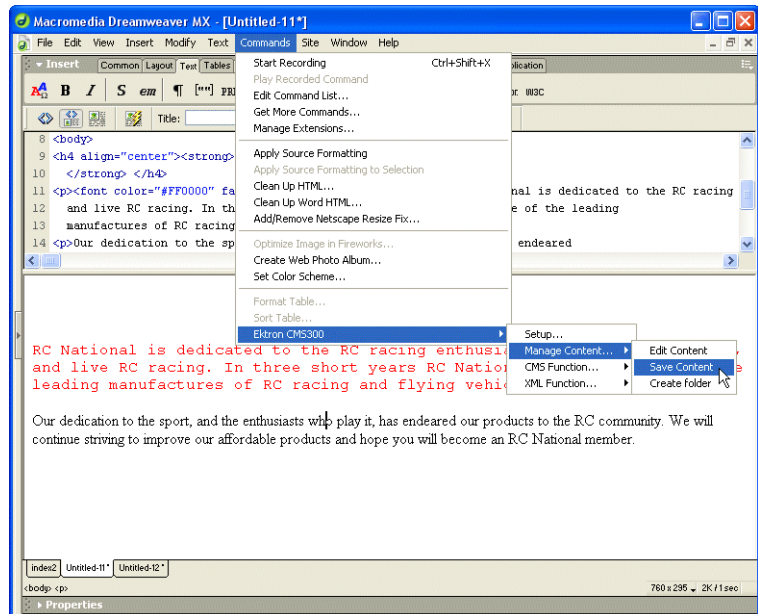
When Ektron CMS200/300 content has been edited in Dreamweaver, you can use the save content command to save your changes back to Ektron CMS200/300.

To save existing Ektron CMS200/300 content, perform the following steps.

1. Make changes to an existing Ektron CMS200/300 content block as described in "Edit Content" on page 391.



2. In the Ektron CMS200/300 Manage Content Menu, click on **Save Content**.



The Save Ektron CMS200/300 Content dialog box is displayed.

3. Use the following table to complete the form.

Field	Description
Title	Editing this field will rename the content block in Ektron CMS200/300.
Comment	Enter a comment for the content block.
Start Date	<p>If desired, type in a start date for the content block.</p> <hr/> <p>NOTE: Date format MUST match what is set in /cms300scripts/setup.asp. By default, this is dd-mmm-yyyy hh:mm AM/PM (for example, 24-Dec-2005 10:15 AM)</p> <hr/>
End Date	<p>If desired, type in an end date for the content block.</p> <hr/> <p>NOTE: Date format MUST match what is set in /cms300scripts/setup.asp. By default, this is dd-mmm-yyyy hh:mm AM/PM (for example, 24-Dec-2005 10:15 AM)</p> <hr/>
Content Folder	Disabled.

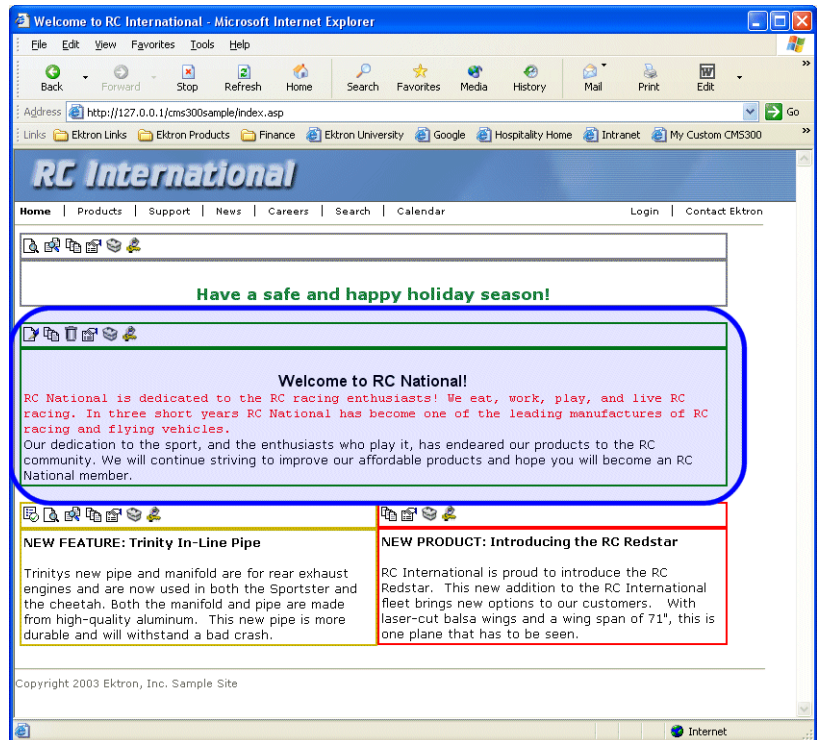
4. Click **Save** to save the changes made to the content.

The content is saved, and the following confirmation message is displayed.



NOTE If you receive a message other than the one displayed above, check your settings and try again.

Displayed here is the updated content displayed on the Web browser.

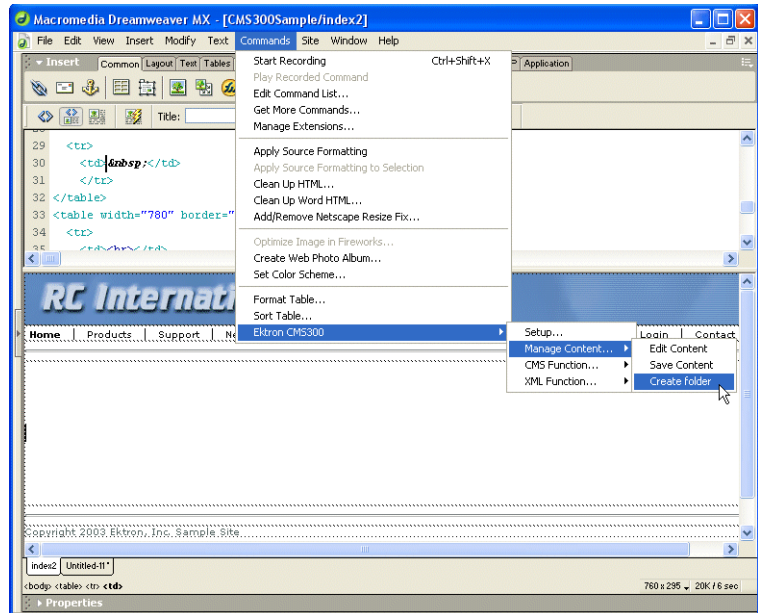


Create Folder

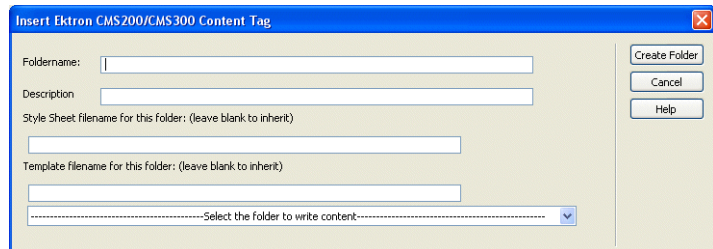
To allow you to easily organize your Web content from Dreamweaver, the Ektron CMS200/300 extension allows you to create new content folders.

To create a new content folder from Dreamweaver, perform the following steps:

1. In the Ektron CMS200/300 Manage Content command menu, click on **Create Folder**.



The Create Ektron CMS200/300 Content Folder dialog box is displayed.



2. Using the following table, enter the necessary information to create the content folder.

Field	Description
Foldername	Type in a name for the content folder to be created.
Description	Enter a description for the new content folder.
Style Sheet	Type in the path and filename, relative to your Ektron CMS200/300 site root, of a style sheet to be applied to content that will belong to the folder. <i>NOTE: Leave blank to inherit the settings of the parent folder.</i>
Template	Type in the path and filename, relative to your Ektron CMS200/300 site root, of a template to be applied to content that will belong to the folder. <i>NOTE: Leave blank to inherit the settings of the parent folder.</i>
Select Folder	From the drop down list, select the existing content folder the new folder will belong to.

Displayed here is an example of a completed form.

The screenshot shows a dialog box titled "Insert Ektron CMS200/CMS300 Content Tag". It contains the following fields and values:

- Foldername:** Dreamweaver
- Description:** Content folder for content created in Dreamweaver
- Style Sheet filename for this folder: (leave blank to inherit):** (blank)
- Template filename for this folder: (leave blank to inherit):** index.asp
- Select Folder:** \

On the right side of the dialog, there are three buttons: "Create Folder", "Cancel", and "Help".

3. Click **Create Folder** to add the new content folder to Ektron CMS200/300.

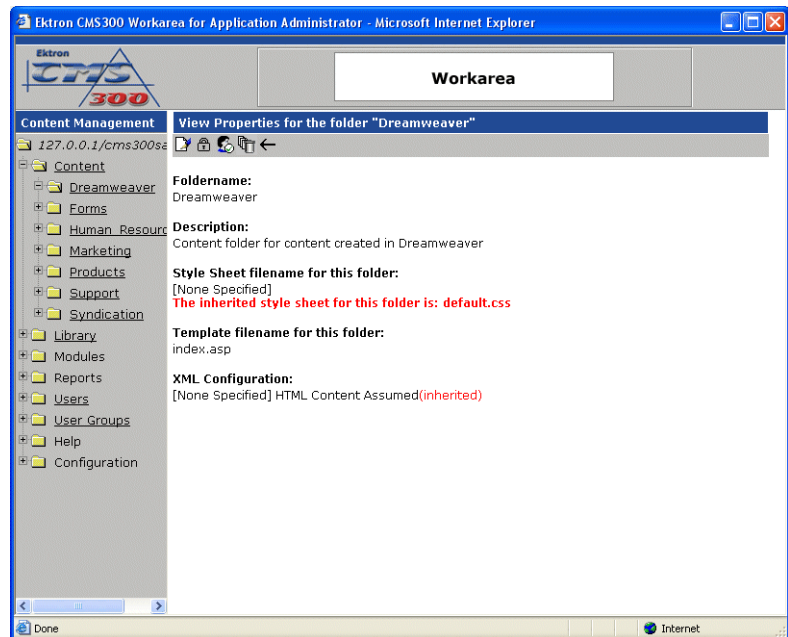
The content folder is added and a confirmation message is displayed.



NOTE

If a message other than this is displayed, check your settings and try again.

Displayed here is the new content folder in the Ektron CMS200/300 Workarea.



CMS Content...

The CMS Content commands allow you to insert standard Ektron CMS200/300 display functions into your template.

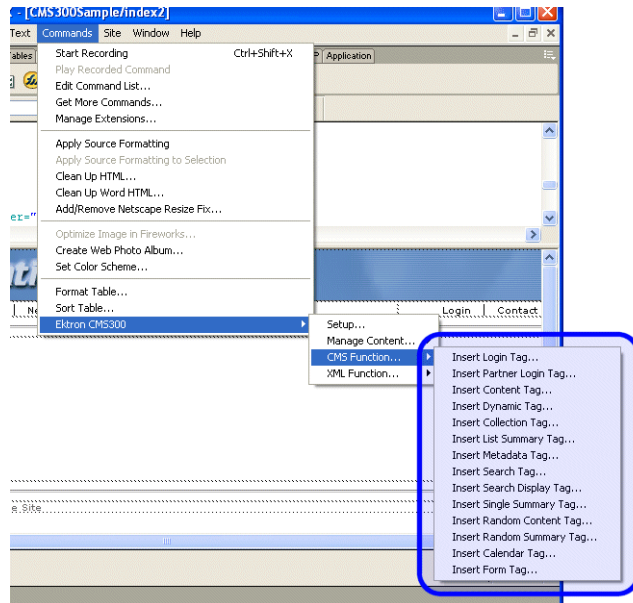
The following sections explain how to insert each of these functions.

Accessing CMS Content Commands...

To access the CMS Content commands menu, follow this path:

Commands > Ektron CMS200/300... > CMS Content...

A list of all CMS Content functions that can be inserted is displayed.



The following sections explain how to insert each of the available functions.

Insert a Login Tag...

To insert a login tag:

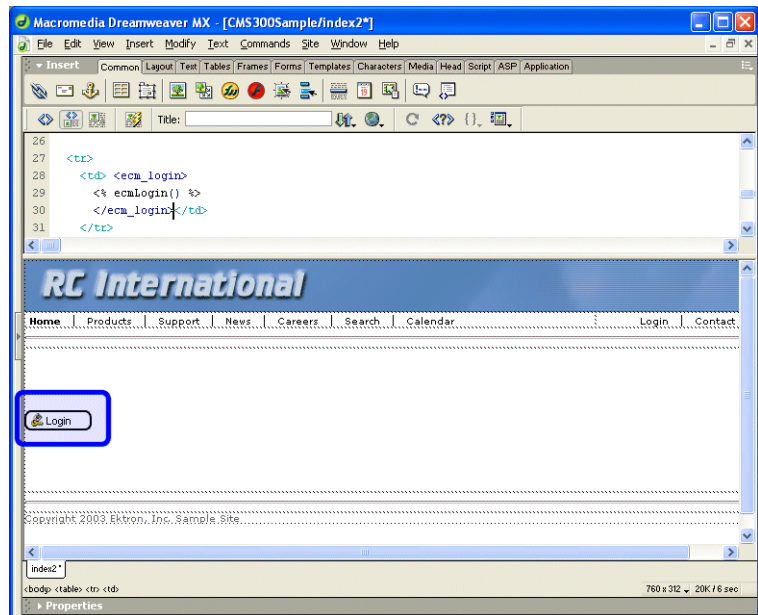
1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content commands, click on **Insert Login Tag...**

You will see the Insert Ektron CMS200/300 Login Tag box.



3. Click **insert**.

There will now be a login icon showing you where the login button will be placed on the template.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<% ecmlogin %>
ColdFusion	<cfmodule template="#request.ecm.AppPath#ecmLogin.cfm">
.NET	<% ecmlogin() %>

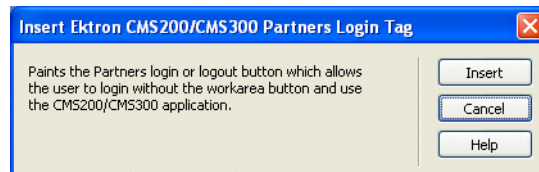
Platform	Code View
PHP	<?php ecmlogin() ?>

Inserting a Partner Login Tag

To insert a partner login tag, which only displays a login and logout button on the Web page, perform the following steps:

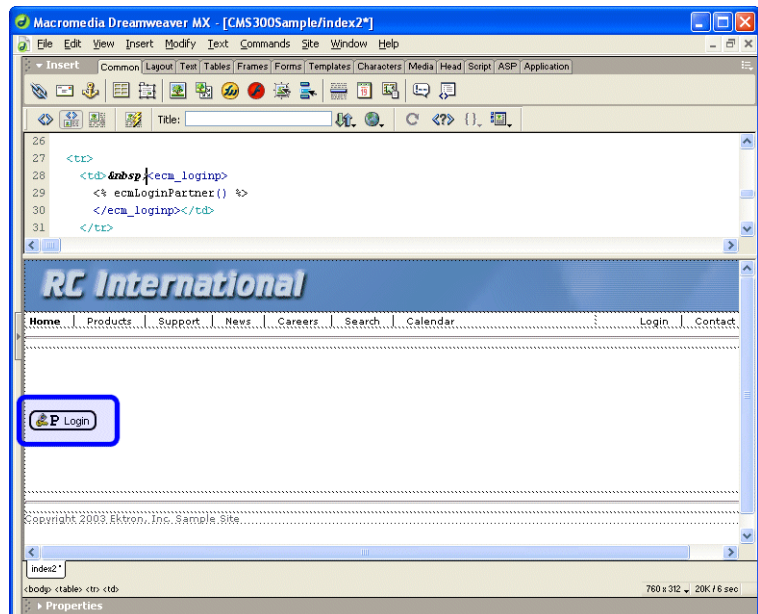
1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content commands, click on **Insert Partner Login Tag...**

You will see the Insert Ektron CMS200/300 Partners Login Tag box.



3. Click **insert**.

There will now be a login icon showing you where the partner login button will be placed on the template.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

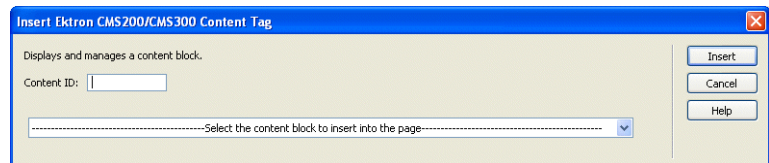
Platform	Code View
ASP	<% ecmLoginPartner %>
ColdFusion	<cfmodule template="#request.ecm.AppPath#ecmLoginPartner.cfm">
.NET	<% ecmLoginPartner() %>
PHP	<?php ecmLoginPartner() ?>

Inserting a Static Content Block

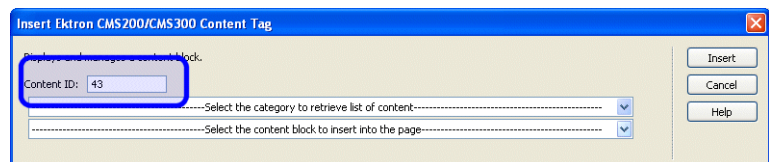
To insert a static content block tag:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Content Tag...**

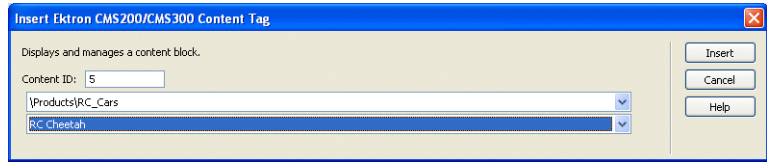
The Insert Ektron CMS200/300 Content Tag box is displayed



- If you know the ID of the content or you want to insert a static content block for a document that doesn't exist yet, enter the ID in the content ID field.



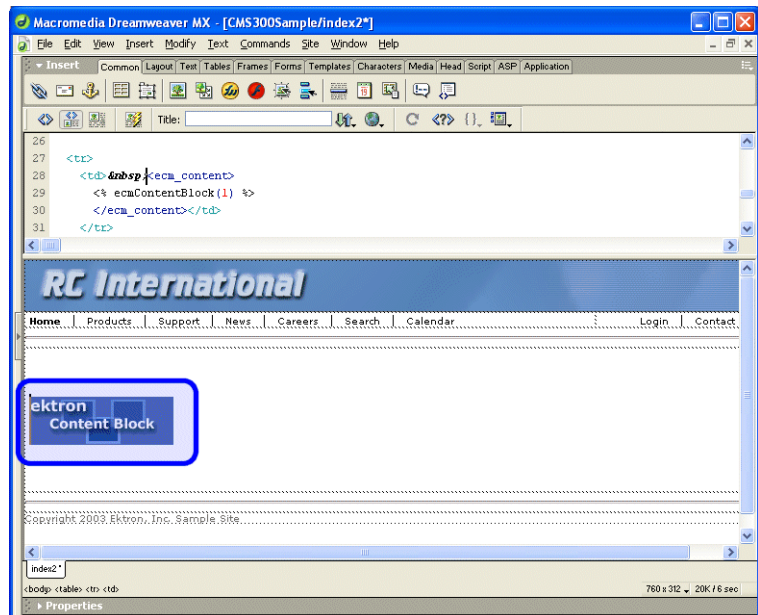
- Click on the field below the Content ID box, and select a content block from the drop down box:



NOTE You will only see the list of Ektron CMS200/300 content blocks if you are working online.

- Click **insert**.

A graphic showing you where the static content block will be placed on the template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<% ecmContentBlock(1) %>
ColdFusion	<cfmodule id="1" template="#request.ecm.AppPath#ecmContentBlock.cfm">

Platform	Code View
.NET	<% ecmContentBlock(1) %>
PHP	<?php ecmContentBlock(1); ?>

Inserting a Dynamic Content Block

To insert a dynamic content block tag:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Dynamic Tag...**

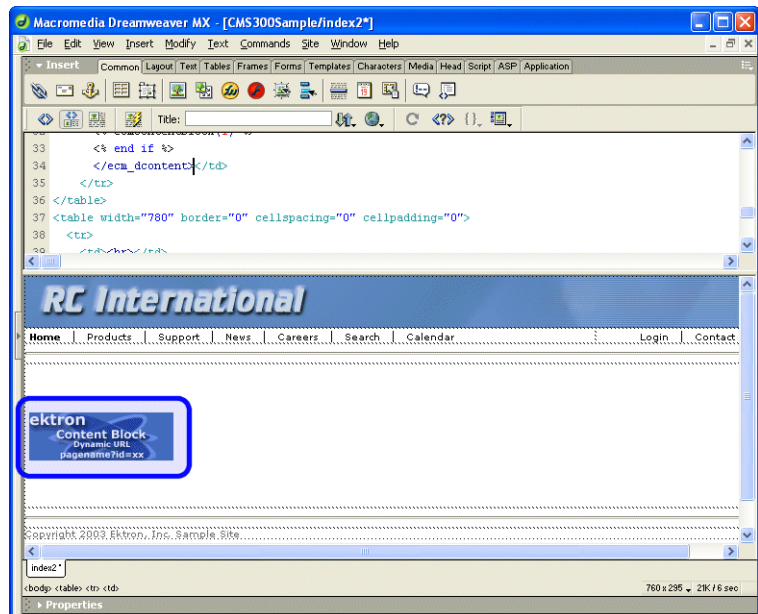
The Insert Ektron Dynamic Tag box is displayed

3. Use the following table to assist you will inserting a dynamic content block function.

Field	Description
Insert Default Content	If you want a static content block to be displayed if no ID is passed through the URL, check off this box.
Default Content ID	Specify the ID number of the content block that you wish to use as a default content block on the Web page.
Select Content	If you are working online, you may select the content block to use as a default content block from the list.

4. When completed, click **insert**.

A graphic to define where the dynamic content block has been placed on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<pre><% if request.QueryString("id") <> "" then ecmContentBlock(request.QueryString("id")) else ecmContentBlock(1) end if %></pre>
ColdFusion	<pre><cfif isdefined("url.id")> <cfmodule id="#url.id#" template="#request.ecm.AppPath#ecmContentBlock.cfm"> <cfelse> <cfmodule id="1" template="#request.ecm.AppPath#ecmContentBlock.cfm"> </cfif></pre>
.NET	<pre><% if request.QueryString("id") <> "" then ecmContentBlock(request.QueryString("id")) else ecmContentBlock(1) end if %></pre>
PHP	<pre><?php \$id =GetURL_Id(1); ecmContentBlock(\$id); ?></pre>

Inserting a Collection Tag

To insert a collection tag:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Collection Tag...**

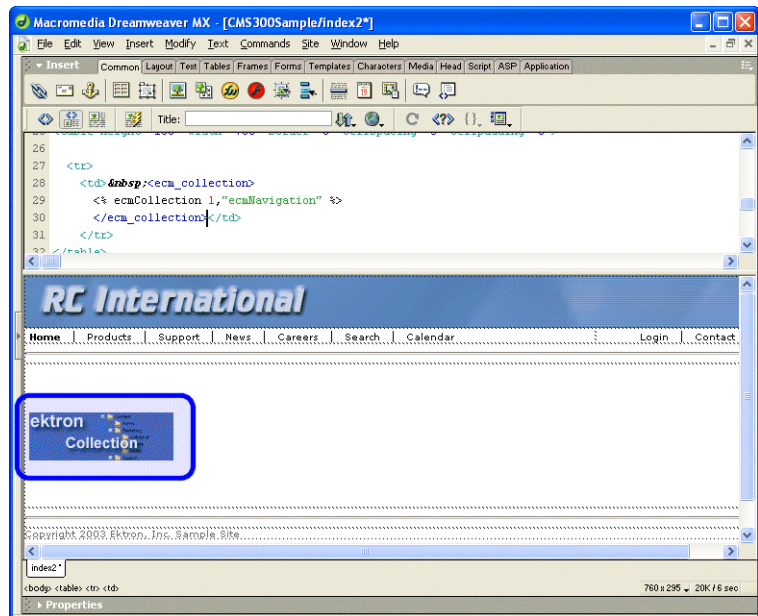
The Insert Ektron Collection Tag box is displayed.

3. Use the following table to assist you will inserting a navigation tag.

Field	Description
Collection ID	<ul style="list-style-type: none"> • Working Offline - If you are working offline, you can enter a Collection ID into the textbox. • Working Online - If you are working online, the top drop-down list will be filled with a list of categories that you have privileges to. After choosing a category, the bottom drop-down box will be filled with a list of all the Collection titles within that category. When a title is selected, the Collection ID textbox will be filled with that Collection's id number.
Display Function	Either choose from the provided list of display functions or fill in one of your own

4. When completed, click **insert**.

A graphic to define where the collection tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<code><% ecmCollection 1,"ecmNavigation" %></code>
ColdFusion	<code><cfmodule id="1" displayFunction="ecmNavigation" template="#request.ecm.AppPath#ecmCollection.cfm"></code>
.NET	<code><% ecmCollection(1,"ecmNavigation") %></code>
PHP	<code><?php ecmCollection(1,"ecmNavigation"); ?></code>

Inserting a List Summary Function

To insert a list summary function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert List Summary Tag...**

The Insert Ektron List Summary Tag box is displayed.

Insert Ektron CMS200/CMS300 List Summary Tag

Used to display a list of content block summaries.

Folder:

Recursive: ☐

Show Summary: ☒

Style Info:

Order By:

Show Info:

Max Number of Summaries:

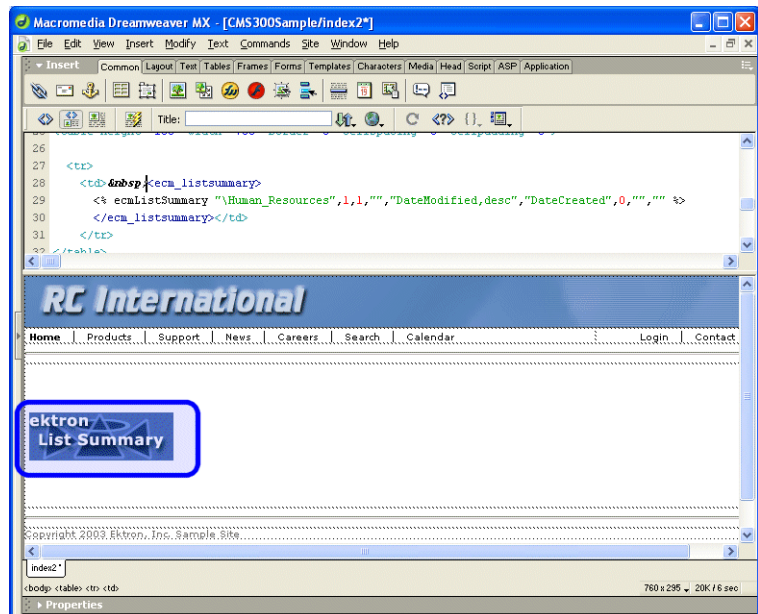
Insert
Cancel
Help

3. Use the following table to assist you will inserting a metadata tag.

Field	Description
Folder	Select the folder the summary information will come from. If you are working online, the folder drop-down contains folders from your CMS200/300 site to choose from. Either enter the folder name manually or select it from the list. The default is the root or "\".
Recursive	If checked, summary information will be gathered from the folder selected and all of it's child folders.
ShowSummary	If checked, content summaries will be displayed.
StyleInfo	Insert any inline style information, which will be applied to the generated text hyperlink, here.
OrderBy	Specify how the summaries are ordered.
ShowInfo	Select the type of information that will be displayed next to the generated hyperlinks.
Max Number of Summaries	Enter a number to specify how may summaries will be displayed on the page (0=unlimited).

- When completed, click **insert**.

A graphic to define where the list summary tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<code><% ecmListSummary "\Human Resources",1,1,"","DateModified,desc", "DateCreated",0,"","" %></code>
ColdFusion	<code><cfmodule template="#request.ecm.AppPath#ecmListSummary.cfm" Folder="\Human Resources" MaxNumber="0" Recursive="1" ShowSummary="1" StyleInfo="" OrderBy="DateModified,desc" ShowInfo="DateCreated"></code>
.NET	<code><% ecmListSummary("\Human Resources",1,1,"","DateModified,desc", "DateCreated",0,"","")%></code>
PHP	<code><?php ecmListSummary("\Human Resources",1,1,"", "DateModified,desc","DateCreated",0,"",""); ?></code>

Inserting a Metadata Function

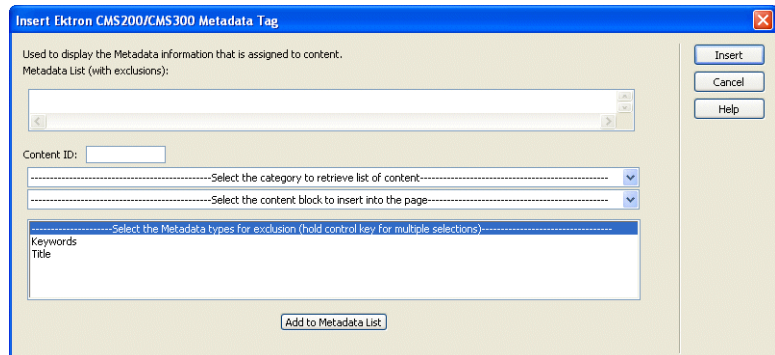
To insert a metadata function:

1. Move the blinking cursor to the area that you want to place it.

NOTE The Ektron CMS200/300 Metadata function should be inserted within the <head> tags of your template to work properly.

2. From the list of Ektron CMS200/300 Content functions, click on **Insert Metadata Tag...**

The Insert Ektron Metadata Tag box is displayed.

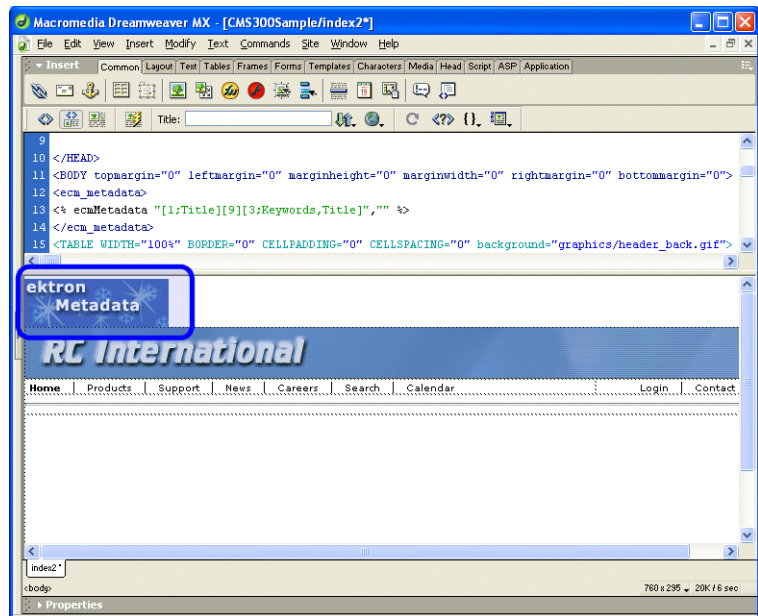


3. Use the following table to assist you will inserting a metadata tag.

Field	Description
Metadata List (with exclusions)	This is the list of content ids which you want to show the assigned Metadata of. The syntax is [id, exclusion list][id, exclusion list]. An example would be [1][2;title, keywords]. The exclusion list is a comma delimited list.
Content ID	<ul style="list-style-type: none"> Working Offline - Enter a valid content ID in the Content ID text-box. Working Online - The top drop-down list contains a list of content folders that you have privileges to. After choosing a content folder, the bottom drop-down box is populated with a list of all the content block titles within that folder. When a title is selected, the content ID textbox will be filled with that content block's id number.
Exclusion List	This is the list of Metadata tags that will be displayed with for the content id. By selecting one or many of the types in this list, you can exclude them from being displayed. An example of an exclusion would be if you wanted to display meta information for 2 content blocks and they both had the same title. You would want to exclude the title for 1 of the content blocks so only 1 title would be displayed.

4. When completed, click **insert**.

A graphic to define where the metadata tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<code><% ecmMetadata "[1;Title][9][3;Keywords,Title]", "" %></code>
ColdFusion	<code><cfmodule template="#request.ecm.AppPath#ecmMetadata.cfm" ItemList="[1;Title][9][3;Keywords,Title]"></code>
.NET	<code><% ecmMetadata("[1;Title][9][3;Keywords,Title]", "")%></code>
PHP	<code><?php ecmMetadata("[1;Title][9][3;Keywords,Title]", ""); ?></code>

Inserting a Search Function

To insert a search function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Search Tag...**

The Insert Ektron Search Tag box is displayed.

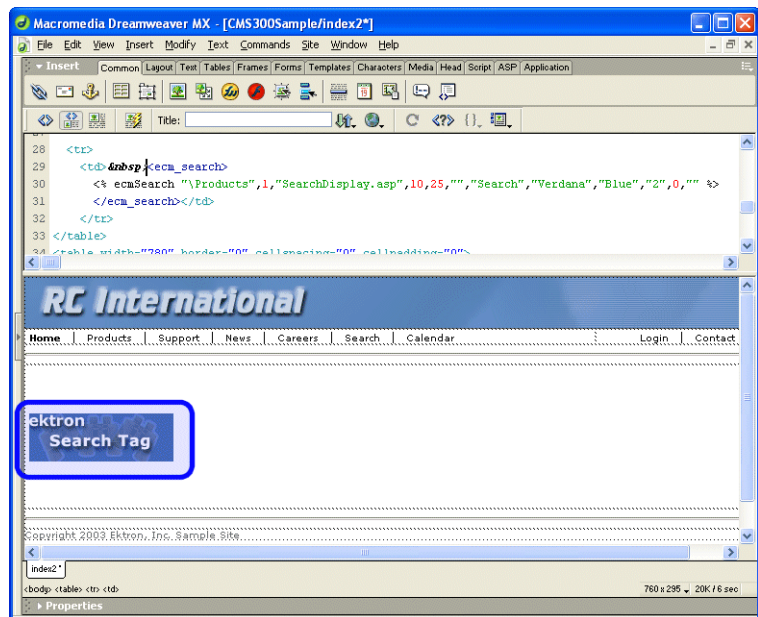
3. Use the following table to assist you will inserting a search tag.

Field	Description
Order By	Specify how you want to order the search results that are generated when a search is performed on you Web site. Also specify if you want the results in ascending or descending order.
Target Page	Specify the Web page where the search results will be displayed. <u>Note: The target page MUST include the search display function</u>
Text Box Size	Specify the size of the text box that will be displayed on the Web page for users to enter the search keywords.
Text Box Max Char	Specify the maximum amount of characters that will accepted in the text box.
Button Image Source	If you wish to use an image for the search button, insert the location of the image in this field. If no image is specified, a standard submit button will be used.
Button Text	The button text is the text that will be displayed on the standard submit button, or the alt text of a search button image.

Field	Description
Font Face	Enter the font face that you would like the search options to appear in on your Web page.
Font Color	Enter the font color that you would like the search options to appear in on your Web page.
Font Size	Enter the font size that you would like the search options to appear in on your Web page.
Display Horizontally	The search tag can be displayed either vertically or horizontally on your Web page. Check off this box if you want it to appear horizontally.

4. When completed, click **insert**.

A graphic to define where the search tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<code><% ecmSearch "\Products",1,"SearchDisplay.asp",10,25,"","Search", "Verdana","Blue","2",0,"" %></code>
ColdFusion	<code><CFMODULE template="#request.ecm.AppPath#ecmSearch.cfm" StartingFolder="\Products" Recursive="1" TargetPage="SearchDisplay.cfm" TextBoxSize="10" MaxCharacters="25" ButtonImageSrc="" ButtonText="Search" FontFace="Verdana" FontColor="Blue" FontSize="2" Horizontal="0"></code>
.NET	<code><% ecmSearch("\Products",1,"SearchDisplay.aspx",10,25,"","Search", "Verdana","Blue","2",0,"") %></code>
PHP	<code><?php ecmSearch("\Products",1,"SearchDisplay.php",10,25,"","Search", "Verdana","Blue","","0",""); ?></code>

Inserting a Search Display Function

To insert a search display function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Search Display Tag...**

The Insert Ektron Search Display Tag box is displayed.

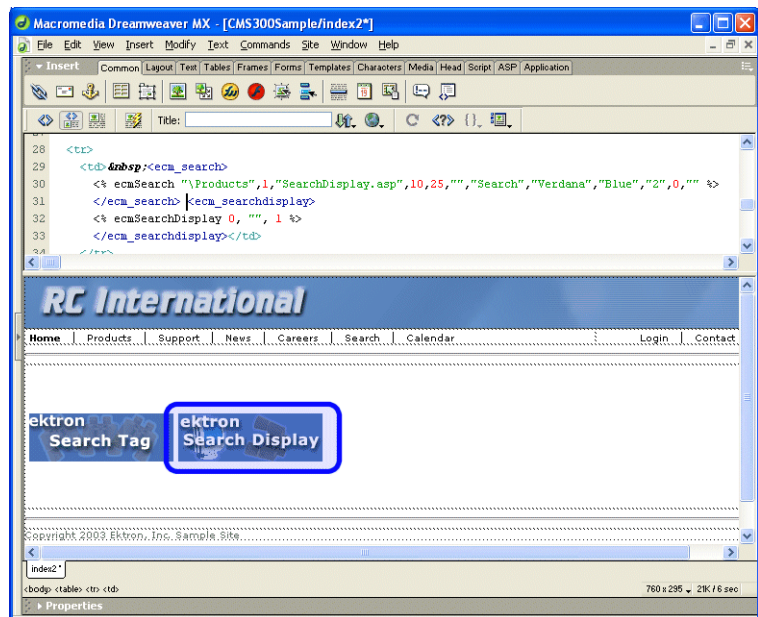
3. Use the following table to assist you will inserting a search display function.

Field	Description
Maximum Number of Documents	Specify the maximum amount of results that will be returned to the user when a search is performed.

Field	Description
Style Info	Insert an inline style sheet that will be used to format all the hyperlinks that are returned.
Show Last Modified Date	Check off this box to show the last modified date with the search result. This allows users to distinguish between updated and old content.
Insert Default Content	Specify the ID number of the content block that you wish to use as a default content block on the Web page when no search is executed.
Default Content ID	If you are working online, you may select the content block to use as a default content block from the list.

4. When completed, click **insert**.

A graphic to define where the search display tag has been placed on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View (with no default content block set)
ASP	<% ecmSearchDisplay 0, " ", 1 %>

Platform	Code View (with no default content block set)
ColdFusion	<pre><cfmodule template="#request.ecm.AppPath#ecmSearchDisplay.cfm" MaxNumber="0" StyleInfo=" " ShowDate="1"></pre>
.NET	<pre><% ecmSearchDisplay(0, " ", 1) %></pre>
PHP	<pre><?php ecmSearchDisplay(0, " ", 1)?></pre>

Inserting a Single Summary Function

To insert a single summary function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Single Summary Tag...**

The Insert Ektron Single Summary Tag box is displayed.

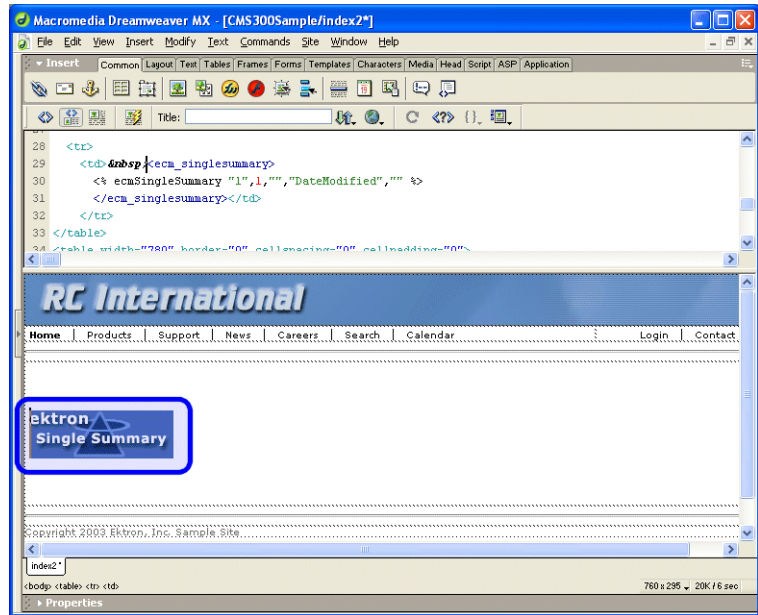
3. Use the following table to assist you will inserting a single summary tag.

Field	Description
Content ID	<ul style="list-style-type: none"> • Working Offline - Enter a valid content ID into the Content ID textbox. • Working Online - The top drop-down list will be filled with a list of content folders that you have privileges to. After choosing a content folder, the bottom drop-down box will be filled with a list of all the content block titles within that category. When a title is selected, the content ID textbox will be filled with that content block's id number.
Show Summary	If checked, content summaries will be displayed.
Style Info	Insert any inline style information, which will be applied to the generated text hyperlink, here.

Field	Description
Show Info	Select the type of information that will be displayed next to the generated hyperlinks.

- When completed, click **insert**.

A graphic to define where the single summary tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

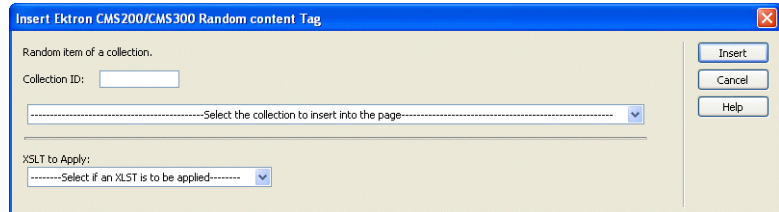
Platform	Code View
ASP	<code><% ecmSingleSummary "1",1,"","DateModified","" %></code>
ColdFusion	<code><cfmodule template="#request.ecm.AppPath#ecmSingleSummary.cfm" ContentID="1" ShowSummary="1" StyleInfo="" ShowInfo="DateModified"></code>
.NET	<code><% ecmSingleSummary("1",1,"","DateModified","")%></code>
PHP	<code><?php ecmSingleSummary("1",1,"","DateModified","") ?></code>

Inserting a Random Content Function

To insert a random content function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Random Content Tag...**

The Insert Ektron CMS200/CMS300 Random Content Tag box is displayed.

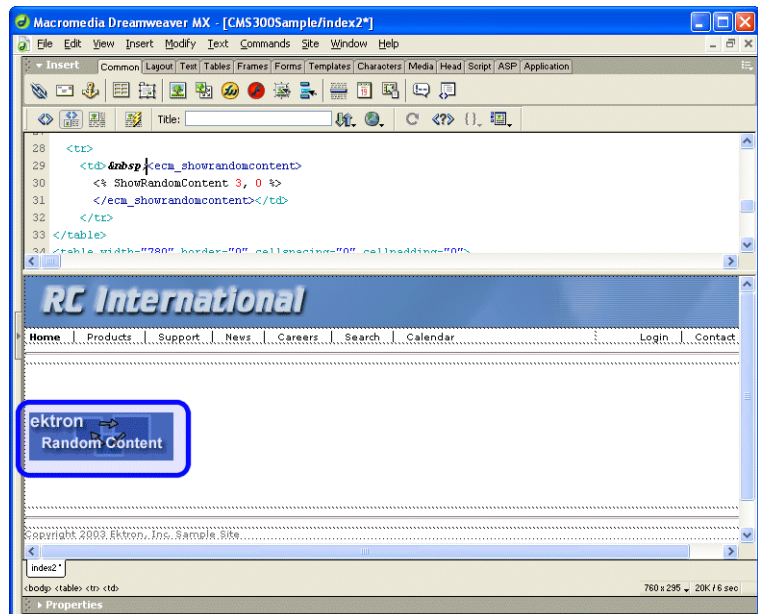


3. Use the following table to assist you will inserting a random content tag.

Field	Description
Collection ID	Type in the collection ID, or select it from the drop down list, of the collection where the random content function will retrieve content from.
XSLT to apply <i>XML Content only</i>	<p>If desired, you may specify an XSLT to apply to the content that gets displayed.</p> <hr/> <p>Not available with CMS200</p> <hr/>

4. When completed, click **insert**.

A graphic to define where the random content tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<code><% ShowRandomContent 3, 0 %></code>
ColdFusion	<code><cfmodule id="3" displayFunction="0" template="#request.ecm.AppPath#ecmShowRandomContent.cfm"></code>
.NET	<code><% ShowRandomContent (3, 0) %></code>
PHP	<code><?php ShowRandomContent(3, 0); ?></code>

Inserting a Random Summary Function

To insert a random summary function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Random Summary Tag...**

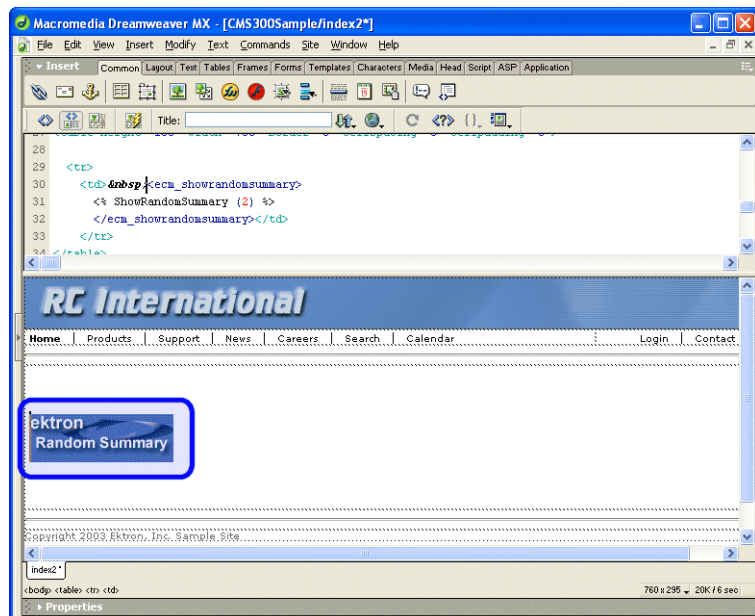
The Insert Ektron CMS200/300 Random Summary Tag box is displayed.

3. Use the following table to assist you will inserting a random content tag.

Field	Description
Collection ID	Type in the collection ID, or select it from the drop down list, of the collection where the random summary function will retrieve summaries from.

4. When completed, click **insert**.

A graphic to define where the random summary tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<% ShowRandomSummary (2) %>
ColdFusion	<cfmodule id="2" template="#request.ecm.AppPath#ecmShowRandomSummary.cfm">
.NET	<% ShowRandomSummary (2) %>
PHP	<?php ShowRandomSummary (2); ?>

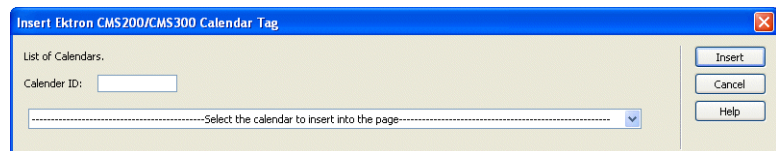
Inserting an Event Calendar Function

To insert an event calendar function:

NOTE Inserting a calendar function does not add a reference to a calendar style sheet. This step must be performed manually.

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Calendar Tag...**

The Insert Ektron CMS200/CMS300 Calendar Tag box is displayed.

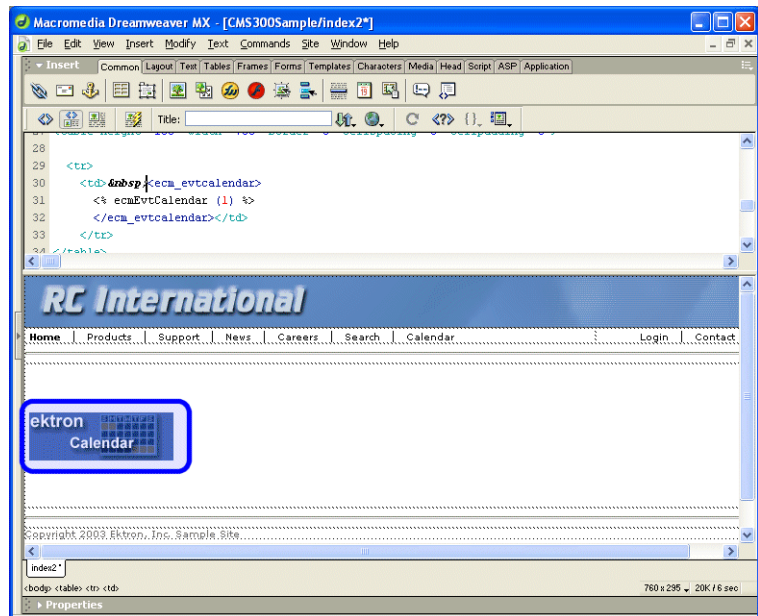


3. Use the following table to assist you will inserting a random content tag.

Field	Description
Calendar ID	Type in the calendar ID, or select it from the drop down list, of the calendar that you want to insert.

4. When completed, click **insert**.

A graphic to define where the event calendar will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<code><% ecmEvtCalendar (1) %></code>
ColdFusion	<code><cfmodule Calendar_id="1" template="#request.ecm.AppPath#ecmEvtCalendar.cfm"></code>
.NET	<code><% ecmEvtCalendar (1) %></code>
PHP	<code><?php ecmEvtCalendar (1); ?></code>

Inserting a Form Content Function

To insert a form content function:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 Content functions, click on **Insert Form Content Tag...**

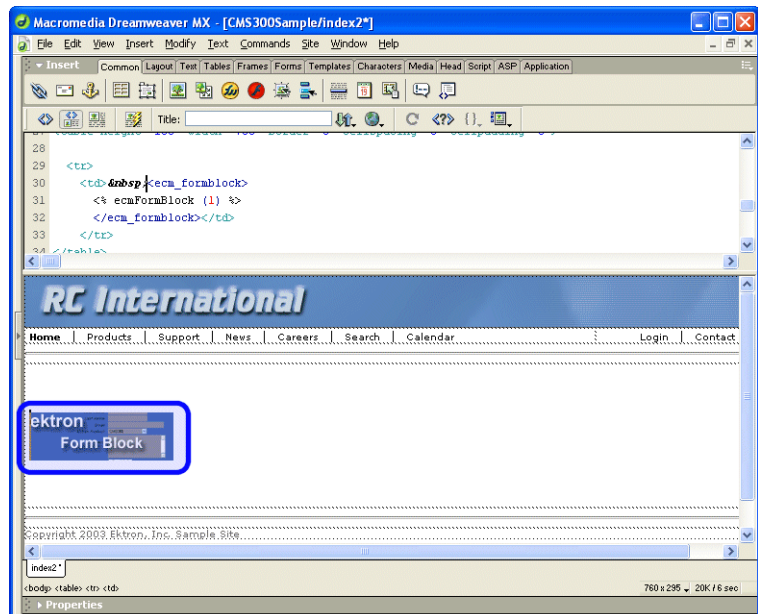
The Insert Ektron CMS200/CMS300 Form Tag box is displayed.

3. Use the following table to assist you will inserting a form tag.

Field	Description
Form ID	Type in the form ID, or select it from the drop down list, of the Ektron CMS200/300 form you would like to insert.

4. When completed, click **insert**.

A graphic to define where the form content tag will appear on your template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<code><% ecmFormBlock (1) %></code>
ColdFusion	<code><cfmodule id="1" template="#request.ecm.AppPath#ecmFormBlock.cfm"></code>
.NET	<code><% ecmFormBlock (1) %></code>
PHP	<code><?php ecmFormBlock (1); ?></code>

XML Functions...

NOTE

This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

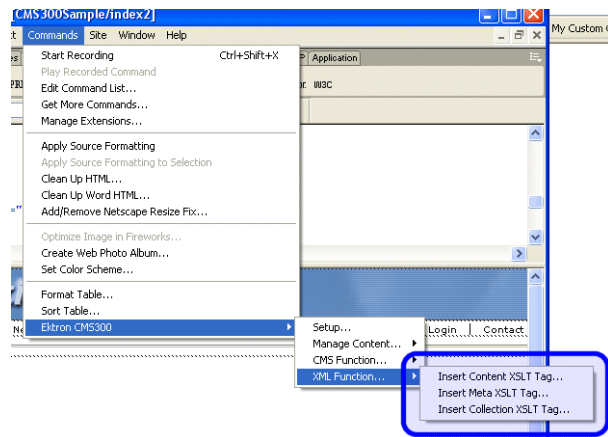
In addition to the many CMS functions, three additional XML related functions that may be inserted into your Web site templates. The following section explains each of these functions.

Accessing XML Function Commands

To access the CMS XML Functions command menu, follow this path:

Commands > Ektron CMS200/300... > XML Functions...

A list of all CMS XML Functions that can be inserted is displayed.



The following sections explain how to insert each of the available XML functions.

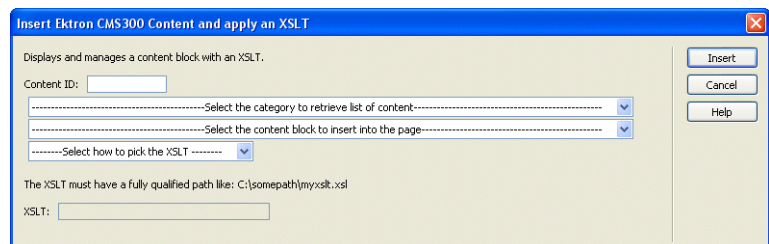
Insert Content XSLT Tag...

The insert content XSLT command allows you to insert an XML content block, and specify an XSLT (other than the default) to be applied to it.

To insert a content XSLT tag, perform the following steps:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of Ektron CMS200/300 XML functions, click on **Insert Content XSLT Tag...**

The Insert Ektron CMS200/300 Content and Apply an XSLT Tag box is displayed.

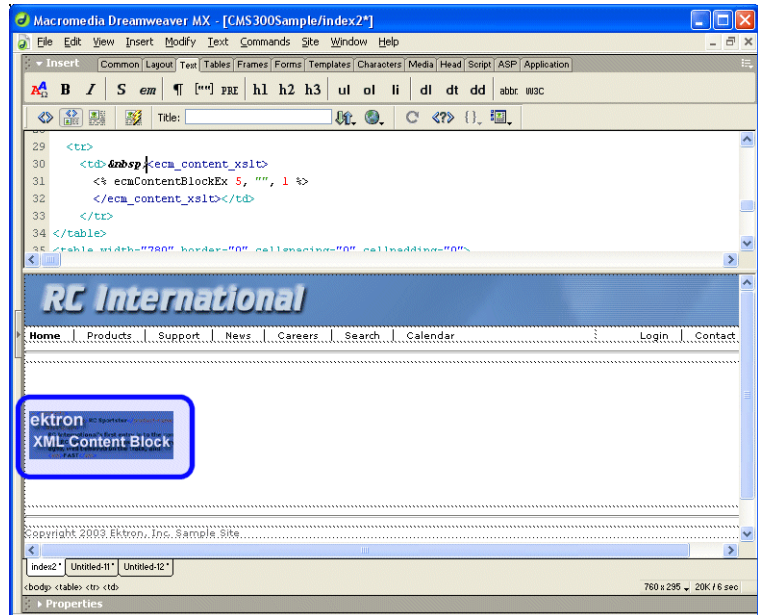


3. Use the following table to assist you with this form.

Field	Description
Content ID	Either type in, or select from the drop down lists, the ID number of the XML content to insert.
Select XSLT	From the drop down list, select the XSLT you would like to apply to the XML content block.
Enter XSLT	If "Enter XSLT" is selected, you can insert the path and file name of the XSLT to apply to the content.

4. Click **insert**.

A graphic showing you where the XML content block will be placed on the template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<% ecmContentBlockEx 5, "", 1 %>

Platform	Code View
ColdFusion	<code><cfmodule id="5" xsltpath="" xslttype="1" template="#request.ecm.AppPath#ecmContentBlockEx.cfm"></code>
.NET	<code><% ecmContentBlockEx (5, "", 1) %></code>
PHP	<code><?php ecmContentBlockEx (5, "", 1); ?></code>

Insert Meta XSLT Tag...

NOTE This feature is not available with Ektron CMS200. If you are using CMS200 and want to upgrade to CMS300, please contact your Ektron sales representative at sales@ektron.com.

The sample multi-language XML configuration in the Ektron CMS200/300 RC International sample Web site contains options to edit metadata for the content block within the editor (instead of the usual metadata area).

A sample XSLT strips out all XML content, with the exception of the metadata information.

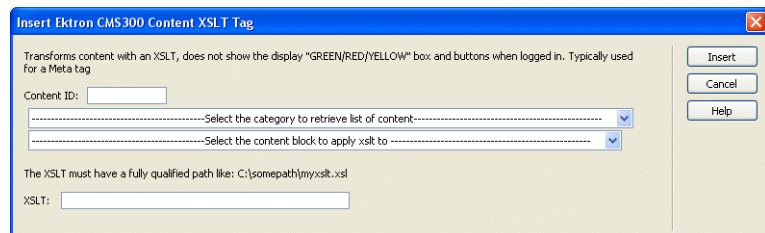
To insert a meta XSLT tag, perform the following steps:

1. Move the blinking cursor to the area that you want to place it.

NOTE Since the inserted content is metadata, the location of the function should be in the `<head>` tags of the template.

2. From the list of Ektron CMS200/300 XML functions, click on **Insert Meta XSLT Tag...**

The Insert Ektron CMS200/300 Meta XSLT Tag box is displayed.

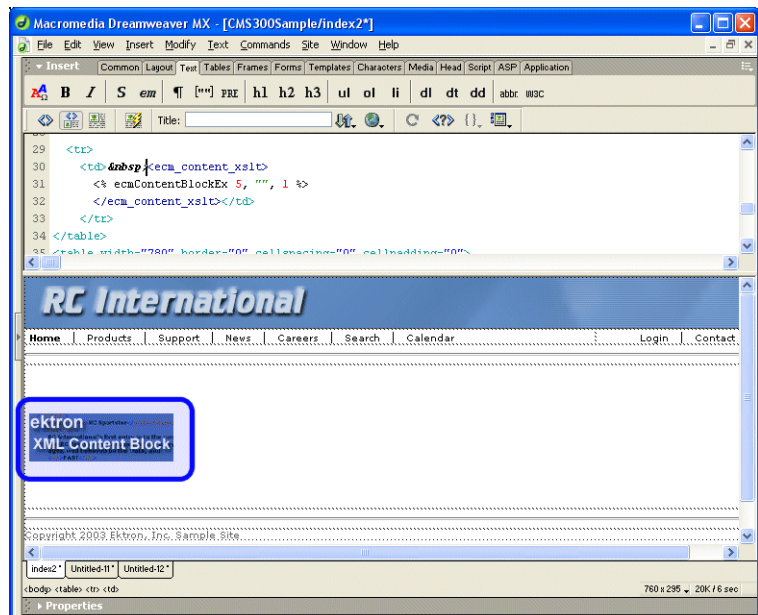


3. Use the following table to assist you with this form.

Field	Description
Content ID	Either type in, or select from the drop down lists, the ID number of the XML content to insert.
XSLT File	Enter a fully qualified path to the XSLT that contains the information to strip out all XML content except for metadata information. For example, C:\Inetpub\wwwroot\CMS300Sample\XMLFILES\xslt\i18lcontent_meta_en.xsl

4. Click **insert**.

A graphic showing you where the XML content block will be placed on the template is inserted.



Shown below is the code that will be inserted according to the platform that has been specified in the setup:

Platform	Code View
ASP	<% ecmMetaEx 19, "C:\Inetpub\wwwroot\CMS300Sample\XMLFILES\xslt\i18lcontent_meta_en.xsl" %>

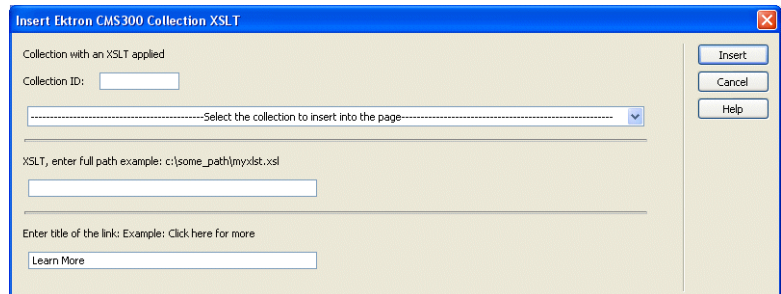
Platform	Code View
ColdFusion	<pre><cfmodule id="19" xsltpath=" C:\Inetpub\wwwroot\CMS300Sample\XMLFILES\ xslt\i18lcontent_meta_en.xsl" template="#request.ecm.AppPath#ecmMetaEx.cfm"></pre>
.NET	<pre><% ecmMetaEx (19, "C:\Inetpub\wwwroot\CMS300Sample\XMLFILES \xslt\i18lcontent_meta_en.xsl") %></pre>
PHP	<pre><?php ecmMetaEx (19, "C:\Inetpub\wwwroot\CMS300Sample\XMLFILES \xslt\i18lcontent_meta_en.xsl"); ?></pre>

Insert Collection XSLT Tag...

The collection XSLT allows you to insert a collection into your Web template, then assign an external XSLT to it.

To insert a Collection XSLT tag, perform the following steps:

1. Move the blinking cursor to the area that you want to place it.
2. From the list of XML functions, click on **Insert Collection XSLT Tag...**
3. The Insert Ektron CMS200/300 Collection XSLT Tag box is displayed.

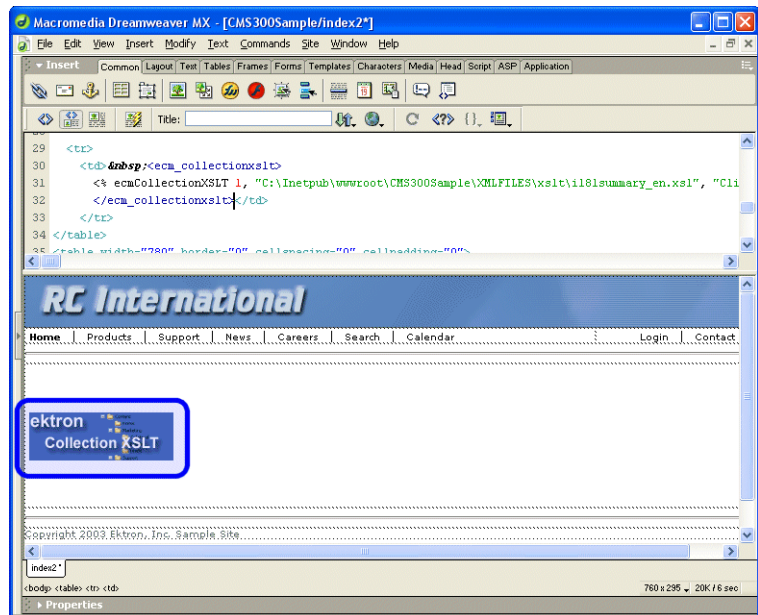


4. Use the following table to assist you with this form.

Field	Description
Collection ID	Either type in, or select from the drop down lists, the ID number of the XML collection to insert.

Field	Description
XSLT File	Enter a fully qualified path to the XSLT that contains the information to strip out all XML content except for metadata information. For example, C:\Inetpub\wwwroot\CMS300Sample\XMLFILES\xslt\i18lsummary_en.xml
Link Title	Enter a title for a link which is used as a "Click Here to Read Full Text" type of hyperlink.

5. Click **insert**.
6. A graphic indicating the XML content block's location on the template is inserted.



Shown below is the code that will be inserted according to the platform specified in the setup:

Platform	Code View
ASP	<% ecmCollectionXSLT 1, "C:\Inetpub\wwwroot\CMS300Sample\XMLFILES\xslt\i18lsummary_en.xml", "Click here to learn more..." %>
ColdFusion	<cfmodule id="1" xslt="C:\Inetpub\wwwroot\CMS300Sample\XMLFILES\xslt\i18lsummary_en.xml" anchorTitle="Click here to learn more..." template="#request.ecm.AppPath#ecmCollectionXSLT.cfm">

Platform	Code View
.NET	<code><% ecmCollectionXSLT (1, "C:\Inetpub\wwwroot\CMS300Sample\XMLFILES\xslt\i18lsummary_en.xml", "Click here to learn more...") %></code>
PHP	<code><?php ecmCollectionXSLT (1, "C:\Inetpub\wwwroot\CMS300Sample\XMLFILES\xslt\i18lsummary_en.xml", "Click here to learn more..."); ?></code>

Additional Information

Supporting Documents

You can find more detailed information about Ektron CMS200/300 in the following manuals:

- *Ektron CMS200/300 Administrator Manual, Version 4.7, Revision 1*
- *Ektron CMS200/300 User Manual, Version 4.7, Revision 1*
- *Ektron CMS200/300 Administrator Quick Start Manual, Version 4.7, Revision 1*
- *Ektron CMS200/300 Setup Manual, Version 4.7, Revision 1*

Support

If you are experiencing trouble with any aspect of Ektron CMS200/300, please contact the Ektron Support Team via our Web Site at

<http://www.ektron.com/support/>

Sales

For questions and answers about purchasing Ektron CMS200/300 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