

webMethods Designer Composite Application Framework Help

Versions 7.1.1 and 7.1.2

October 2008



Contents

	About This Guide Document Conventions Additional Information	17
Part I.	webMethods CAF	21
	Chapter 1. Getting Started with Web Application Development	23
	Introduction to the webMethods UI Developer Perspective	24
	Creating a Web Application	25
	Creating Web Applications for Tomcat	26
	Creating Web Applications for JBoss	27
	Creating Web Applications for Multiple Server Types	29
	Creating Web Applications for My webMethods Server	30
	Adding a View File to a Web Application	32
	Adding a Control to a View File	33
	Associating Web Applications with an Application Server	34
	Publishing a Web Application to the Server	34
	Chapter 2. Developing Web Applications	37
	Component Libraries	38
	Adding Component Libraries to a Web Application	38
	Removing Component Libraries from a Web Application	39
	Modifying How Component Libraries are Deployed	39
	Project Templates	40
	Style Sheets in Web Applications	41
	Applying CSS to an Individual Control	41
	Applying CSS to a View File	42
	Importing a View	43
	Page Templates	43
	Page Templates and Content Parameters	45
	Page Templates and Control Parameters	46
	Navigation in Web Applications	47
	The Application Navigation Configuration Editor	47
	Adding a Home Link to a Web Application	49
	Hiding Breadcrumbs on the Home Page	49
	Application Initialization Parameters	50
	Environment Variables	51

Chapter 3. Working with Application Servers	53
Installing and Configuring an Application Server	54
Configuring the Apache Tomcat Servlet Container	54
Configuring the JBoss Application Server	57
Web Application Security	58
Security on Tomcat	58
Security on JBoss	60
Security Roles for Web Applications	61
Deploying Component Libraries on an Application Server	62
Making the Tomcat Server Re-render the Page Automatically	63
Running Servers in Debug Mode	64
Chapter 4. Getting Started with Portlet Application Development	65
Introduction to the My webMethods Perspective	66
Creating a Portlet Application	68
Adding a Portlet to a Portlet Application	70
Adding a View File to a Portlet	72
Adding a Control to a View File	73
Connecting to My webMethods Server	74
Associating Portlet Applications with My webMethods Server	76
Running Servers in Debug Mode	76
Publishing a Portlet Application to the Server	77
Adding a Portlet to a My webMethods Server Page	77
Uninstalling Portlet Applications from My webMethods Server	78
Troubleshooting Portlet Applications	79
Troubleshooting Fortiet Applications	,,
Chapter 5. webMethods CAF Concepts	81
Portlet Applications	82
JSR 168	82
JavaServer Faces	82
Ajax	83
JSF Scopes	83
Chapter 6. Views in webMethods CAF	85
My webMethods Perspective Overview	86
Using the Portlet Application Configuration Editor	88
Creating Portlet Preferences	89
Using Preferences to Perform Wiring	90
Storing a Preference	92
Using Security Roles	94
Creating a Security Role	94

Binding a Security Role to a Control	94
Using an Access Control Panel Control	95
User Attributes	95
Using the Design Canvas	. 97
The Design Canvas Toolbar	. 99
Manipulating Controls on the Design Canvas	100
The Instant Preview	103
Live Preview	104
Debugging from the Design Canvas	106
Using the Bindings View	106
Creating a New Method	107
Adding a New Property to a Page Bean	109
Customizing Initial Values for a Page	110
Showing All Managed Beans	111
Adding a Reference to Another Page	112
Using the Data Source Explorer View	
Connecting to a Database	
Creating a Database Connector	114
Using the IS Browser View	
Connecting to Integration Server	116
The IS Browser View Toolbar	. 117
Authentication Credentials on the IS Browser View	118
Modifying the Authentication Information for Integration Server	. 119
Providing Integration Server Authentication at Run Time	120
Creating a Web Service Connector	121
Deleting a Web Service Connector	
Renaming Integration Server on the IS Browser View	123
Changing Properties on the IS Browser View	124
Displaying Integration Server in a Browser	
Using the MWS Admin View	125
Connecting to My webMethods Server	127
The MWS Admin View Toolbar	128
Authentication Credentials on the MWS Admin View	129
Deploying Portlets to My webMethods Server	129
Exporting from My webMethods Server	130
Importing to My webMethods Server	132
Renaming My webMethods Server on the MWS Admin View	133
Changing Properties on the MWS Admin View	
Other Actions You Can Perform on the MWS Admin View	135
Displaying My webMethods Server in a Browser	126

Using the Outline View	137
Using the Palette View	138
Moving Objects from the Palette View	140
Managing Favorites in the Palette View	141
Customizing the Palette View	142
The Palette View Toolbar	143
Displaying Deprecated Items	143
Filtering Controls	144
Using the Properties View	145
The Properties View Toolbar	145
Actions in the Properties View	146
Adding Converters or Validators to a Control	147
Binding Data to a Control	148
Using the Solutions View	149
Portlet Applications in the Solutions View	150
Creating a Portlet Application on the Solutions View	150
Creating a Portlet on the Solutions View	150
Creating a Portlet View File on the Solutions View	152
Web Applications in the Solutions View	153
Creating a Web Application on the Solutions View	153
Creating a Web Application View File on the Solutions View	154
Database Connectors in the Solutions View	
Specifying an External Web Browser	155
Chapter 7. Samples for the My webMethods Perspective	
Importing Samples into the My webMethods Perspective	158
Common Sample Issues	159
Access Control Sample	160
Core Controls Sample	161
Core Provider Sample	161
File Export Sample	162
Locale Order Sample	
Localization Sample	163
Northwind Sample	164
Portlet Links Sample	164
Search Sample	165
Wired Config Sample	165
Wizard Sample	166
Chapter 8. A Tutorial for Portlet Application Samples	169
Using the Portlet Tabs Tutorial	170

	Portlet Tabs Tutorial	170
	Create a Portlet Application Project	171
	Create a Portlet	171
	Create a List of Views	172
	Create the Tabs Control	173
	Create Some Content	174
	Create the Other Pages	174
	Publish and View Portlets on My webMethods Server	175
Part II.	User Interface Controls Reference	179
	Chapter 9. Controls Listed by Component Library	181
	Chapter 10. User Interface Controls Concepts	191
	Control ID Reference	192
	Hideable Controls	194
	Toggle Controls	197
	Scriptaculous Effects	198
	Client-Side Model	201
	CAF.Model	
	CAF.Output.Model	
	CAF.Link.Model	
	CAF.Command.Model	
	CAF.Input.Model	
	CAF.Checkbox.Model	
	CAF.Select.Model	
	CAF.Table.Model	
	CAF.Table.Row.Model	
	Template Row	
	Paging	
	Listeners	
	CAF.Tree.Model	
	Image URLs	
	Skinning	
	Table Row Tools	
	Adding and Removing Rows	
	Moving Rows	
	Selecting Rows	
	Client-Side Libraries	
	CSV	217

String	. 218
Logger	. 218
Logger Bookmarklet	. 218
Logger and Firebug	. 219
Library	. 219
CAF.Dialog Class	. 220
CAF.Progress Class	. 221
CAF.Tooltip Class	. 222
CAF.Request Class	. 222
CAF.Updater Class	. 223
CAF.Draggable Class	. 223
CAF.Droppable Class	. 225
Filterable Controls	. 226
CAF.Select.Model	. 228
Examples of Filtering	. 229
Filtered Listbox (Medium List)	. 229
Filtered Listbox (Large List)	. 230
Filtered Swapbox (Medium List)	. 230
Filtered Swapbox (Large List)	. 231
Filtered Dropdown (Medium List)	. 232
Filtered Combobox (Medium List)	. 232
Autocomplete Text Field (Large List)	. 233
Filtered Table (Current Page)	. 233
Filtered Table (Entire Table)	. 233
Chapter 11. Behavior Controls	235
Custom Drag	
Custom Drop	
Drag To Move Rows	
Drag To Reorder Columns	
Drag To Resize	
Drag To Resize Columns	
Raise On Change	
Synchronize Values	
Chapter 12. Command Controls	249
Async Command Button	
Async Command Icon	
Async Command Interval	
Async Command Link	
Async Hidden Command	. 259

	Command Button	261
	Command Icon	263
	Command Link	265
	Form	268
	Hidden Command	271
	Servlet Command Form	272
Cha	pter 13. Dialog Controls	
	Modal Dialog	
	Modeless Dialog	
	Portal Resource Picker Dialog	
	Principal Picker Dialog	
	Progress Dialog	286
Cha	apter 14. Input Controls	200
Ulla	·	
	Autocomplete Text Input	
	Checkbox	
	Checkbox Group	
	Combobox	
	Date Input	
	Date Range Input	
	Dropdown	
	Extended Select-Many Listbox	
	Extended Select-One Listbox	
	File Input	
	Filter Input	
	Hidden Input	
	HTML Input	
	In Place Text Input	
	Option	
	•	
	Option Group	
	Radio Button Group	
	Secret Input	
	Select-Many Listbox	
	Select-One Button	
	Select-One Link	340
	Select-One Listbox	
	Select-One Tabs	
	Simple Schedule Input	347

	Swapbox Text Input	
	pter 15. Output Controls	
	Applet	
	Button	
	Control Label	
	Custom Element	
	Flash	
	Formatted Message	
	Formatted Messages	
	Formatted Text	
	Frame	
	Header	
	Horizontal Rule	
	Icon	
	Image	374
	Include HTML	
	Include Resource Bundle	376
	Include Script	377
	Include Stylesheet	378
	Link	379
	Message	381
	Messages	383
	Parameterized Text	384
	Refresh Button	385
	Refresh Icon	386
	Refresh Interval	387
	Refresh Link	388
	Script Block	389
	Select-One Output Text	389
	Specific Formatted Messages	390
	Specific Messages	391
	Text	
	Truncated Text	394
Cha	pter 16. List Controls	397
	Async List	398
	Async Listbox	401
	Async Tabs	403
	Listbox	406

Simple List	
Chapter 17. Logic Controls	413
Async Iterator	
Content Parameter	
Control Parameter	
Else	
lf	418
Import View	
Iterator	421
Load Resource Bundle	423
Naming Container	
Chapter 18. Map Controls	425
Dynamic Map Marker List	426
Google Map	431
Google Map Key	435
Invoke Map Script	435
Map Marker	438
Map Marker Group	443
Map Navigation Control	444
Return Map Coords Script	445
Return Map Value Script	446
Chapter 19. Panel Controls	
Access Control Panel	450
Attachments Panel	451
Block Edge Panel	
Block Panel	453
Disableable Panel	454
Grid Panel	
Hideable Panel	457
Inline Hideable Panel	460
Inline Panel	462
Overlay Panel	463
Page Group	465
Popup Panel	466
Property Group	467
Property Line	469
Property Sub-Group	469

Scrolling Panel	471
Stack Panel	472
Static Cell	473
Static Row	474
Submit Group	475
Titlebar Tabs Wrapper	476
Tooltip	476
Chapter 20. Portlet Controls	479
Extended Portlet Parameter	
Extended Portlet URL	
Portlet Include	
Portlet Parameter	
Portlet Simple Link	
Portlet URL	
FOILIEL ORL	407
Chapter 21. Script Controls	489
Control Reference	490
Custom Script	490
Invoke Script	491
Parameter	494
Portlet Url Script	496
Portlet Url Script Parameter	498
Return Value Script	500
Chapter 22. Search Controls	503
Async Search Result Table	
Async Search Result Tree	
Search Bar	
Search Result Table	
Search Result Tree	
Sedicii Result 11ee	
Chapter 23. Table Controls	525
Async Categorized Table	526
Async Table	531
Async Tree	536
Atom Feed Icon	541
Basic Column	547
Calendar	549
Categorized Table	
Column Sort Link	557

	Data Pages	558
	Data Prev/Next	559
	Data Total	560
	Data Total Selected	562
	Export Table Button	562
	Hidden Column	565
	Newsfeed	565
	Row Headers Column	566
	Scrollbar Column	568
	Standard Column	568
	Table	570
	Tree	575
	Tree Toggle	580
	Truncating Column	582
Ob a	when 24. Table Day, Cambrela	- 0-
Cna	pter 24. Table Row Controls	
	Add Row Button	
	Add Row Icon	
	Add Row Link	
	Move Row Down Button	
	Move Row Down Icon	
	Move Row Down Link	
	Move Row Up Button	
	Move Row Up Icon	
	Move Row Up Link	
	Remove Row Button	
	Remove Row Icon	
	Remove Row Link	
	Select All Rows Checkbox	
	Select All Visible Rows Checkbox	
	Select Row Checkbox	
	Select Row Column	
	Select Row Link	
	Select Row On Click	
	Select Row Quad-State Checkbox	
	Select Row Tri-State Checkbox	609
Cha	pter 25. Toggle Controls	611
	Initiate Toggle	612
	One Way Toggle Button	
	One Way Toggle Checkbox	

	One Way Toggle Link	. 615
	Toggle Button	. 616
	Toggle Dropdown	. 618
	Toggle Link	. 620
	Toggle Radio Button Group	. 621
	Toggle Tabs	623
	Chapter 26. Webapp Controls	625
	Breadcrumbs	
	Popup Menus	
	Static Menus	
	Toggle Menus	
Part III.	Converters Reference	633
	Chapter 27. Converter Concepts	635
	Using Converters	. 636
	Creating a Custom Converter	636
	Chapter 28. Converters	639
	Array Converter	. 640
	Big Decimal Converter	. 640
	Big Integer Converter	. 64
	Boolean Converter	. 64
	Byte Converter	. 64
	Character Converter	. 642
	Collection Converter	. 642
	Date and Time Converter	. 643
	Double Converter	. 644
	Float Converter	. 644
	Integer Converter	. 645
	Iterator Converter	. 645
	Long Converter	. 646
	Map Converter	. 646
	Number Converter	. 647
	Short Converter	. 648
	String Array Converter	. 649
	WmPortal IURI Converter	. 649
	Chapter 29. Currency Converters	651
	Currency Big Decimal Converter	652
	Currency Big Integer Converter	653

	Currency Double Converter	654
	Currency Float Converter	655
	Currency Integer Converter	656
	Currency Long Converter	657
	Currency Number Converter	658
	Currency Short Converter	659
	Chapter 30. Formatted Converters	661
	Formatted Big Decimal Converter	662
	Formatted Big Integer Converter	662
	Formatted Boolean Converter	663
	Formatted Date Time Converter	664
	Formatted Double Converter	665
	Formatted Float Converter	665
	Formatted Integer Converter	666
	Formatted Long Converter	667
	Formatted Number Converter	667
	Formatted Short Converter	668
Part IV.	Validators Reference	669
	Chapter 31. Validator Concepts	671
	Using Validators	
	Creating a Custom Validator	672
	Chapter 32. Validators	675
	Credit Card Validator	676
	Double Range Validator	677
	Email Validator	677
	Equal Validator	678
	Exact RegExp Validator	678
	Length Validator	679
	Long Range Validator	680
	One Of Validator	681
	RegExpr Validator	681
	Chapter 33. Formatted Validators	683
	Formatted Date Range Validator	684
	Formatted Double Range Validator	685
	Formatted Long Range Validator	686

About This Guide

This guide contains the Composite Application Framework online help in PDF book format. The information in this guide is the same information that you can view via the webMethods Designer online help.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Italic	Identifies variable information that you must supply or change based on your specific situation or environment. Identifies terms the first time they are defined in text. Also identifies service input and output variables.
Narrow font	Identifies storage locations for services on the webMethods Integration Server using the convention <i>folder.subfolder:service</i> .
Typewriter font	Identifies characters and values that you must type exactly or messages that the system displays on the console.
UPPERCASE	Identifies keyboard keys. Keys that you must press simultaneously are joined with the "+" symbol.
\	Directory paths use the "\" directory delimiter unless the subject is UNIX-specific.
[]	Optional keywords or values are enclosed in []. Do not type the [] symbols in your own code.

Additional Information

The webMethods Advantage Web site at http://advantage.webmethods.com provides you with important sources of information about webMethods products:

- Troubleshooting Information. The webMethods Knowledge Base provides troubleshooting information for many webMethods products.
- Documentation Feedback. To provide feedback on webMethods documentation, go to the Documentation Feedback Form on the webMethods Bookshelf.
- Additional Documentation. You have the option of downloading the documentation during product installation to a single directory called "_documentation," located by

default under the webMethods installation directory. In addition, you can find documentation for all webMethods products on the webMethods Bookshelf.

Part I. webMethods CAF

webMethods CAF allows you to create Web application projects for use with application servers and portlet application projects for use with My webMethods Server. A Web application is an application that a user accesses over the Web on either the Internet or an intranet. As created in the webMethods CAF, a Web application is made up of one or more views (pages) intended to be displayed by an application server, such as Apache Tomcat or JBoss. A portlet application is composed of one or more portlets, which in turn are made up of one or more views (pages) intended to be displayed by My webMethods Server. webMethods CAF is covered in the following online help topics:

- "Getting Started with Web Application Development" on page 23
- ▶ "Developing Web Applications" on page 37
- "Working with Application Servers" on page 53
- "Getting Started with Portlet Application Development" on page 65
- "webMethods CAF Concepts" on page 81
- "Views in webMethods CAF" on page 85
- "Samples for the My webMethods Perspective" on page 157
- "A Tutorial for Portlet Application Samples" on page 169

You will also find the following reference topics:

- "User Interface Controls Reference" on page 179
- ▶ "Converters" on page 639
- ▶ "Validators" on page 675

webMethods Designer - Composite Application Framework Help Version 7.1.1	

Chapter 1. Getting Started with Web Application Development

Introduction to the webMethods UI Developer Perspective	. 24
Creating a Web Application	. 25
Adding a View File to a Web Application	. 32
Adding a Control to a View File	. 33
Associating Web Applications with an Application Server	. 34
Publishing a Web Application to the Server	. 34

Introduction to the webMethods UI Developer Perspective

A Web application is an application that a user accesses over the Web on either the Internet or an intranet. As created in the webMethods CAF, a Web application is made up of one or more views (pages) intended to be displayed by an application server, such as Apache Tomcat or JBoss. Do not use this perspective to create portlet applications for My webMethods Server (see "Getting Started with Portlet Application Development" on page 65). To create Web applications, you use the webMethods UI Developer perspective.



To open the webMethods UI Developer perspective

- 1 To start Designer: Start > Programs > webMethods7 > Tools > Designer.
- In the Workspace Launcher window, accept the default workspace or click **Browse** and navigate to the folder where you want the Eclipse workspace to reside, and then click **OK**.

If you have installed only webMethods CAF, Designer opens in the webMethods UI Developer perspective. If you have installed the BPMS Suite, Designer opens in the Business Analyst perspective. To create Web application projects, change to the webMethods UI Developer perspective.

- 3 If needed, in Designer: Window > Open Perspective > Other > webMethods UI Developer.
- 4 Click OK.

The webMethods UI Developer perspective contains the following views. For more information on views, see "Views in webMethods CAF" on page 85:

View	Description
Bindings	A tree view that contains all data that can be used within this view or within the entire Web application.
Data Source Explorer (Eclipse)	A tree view of data sources. Using this view you can connect to, navigate, and interact with resources associated with a selected connection profile, such as a database.
Outline (Eclipse)	An outline of a view that is currently open in the design canvas. This view is useful in locating specific controls within a complex Web application and for moving controls from one location in the Web application to another.
Palette	A tree view of controls you can use to build Web applications. You can drag controls from this view directly to the design canvas.

View	Description
Problems (Eclipse)	A display of system-generated errors, warnings, or information associated with a resource. For example, if you save a Java source file that contains syntax errors, the errors are automatically are logged in this view.
Project Explorer (Eclipse)	A tree view of the resources in the workspace. From here, you can open files for editing or select resources for operations such as exporting.
Properties (Eclipse)	A view of display names and values of properties for a resource selected elsewhere in Designer. This view is useful in changing the values of properties of controls in the design canvas.
Servers (Eclipse)	A table of runtime environments associated with Designer. Once configured, you can publish Web applications to the server.
Solutions	A tree view that represents a logical representation of the resources in the workspace which is simpler than the one provided in the Project Explorer. You can perform selected operations on resources shown in this view.

In addition to the views, there are various editors:

Editor	Description
Design canvas	A JSF graphical editor in which you build portlets by adding controls from the Palette view. This editor provides a logical view of the Web application.
Java editor	A source code editor that provides specialized features for editing Java code.

Creating a Web Application

If you have the webMethods UI Developer perspective displayed ("Introduction to the webMethods UI Developer Perspective" on page 24) and have defined an application server on the local host ("Installing and Configuring an Application Server" on page 54), you are ready to create a Web application project.

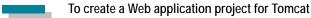
The configuration settings depend on what kind of application server you intend to use for development. The following procedures describe a way to get started in creating Web application projects. As you gain experience with webMethods CAF, you can modify your the procedures as appropriate:

- ▶ "Creating Web Applications for Tomcat" on page 26
- "Creating Web Applications for JBoss" on page 27

- "Creating Web Applications for Multiple Server Types" on page 29
- "Creating Web Applications for My webMethods Server" on page 30

Creating Web Applications for Tomcat

The following procedure describes how to create a Web application for use with the Tomcat servlet container. This procedure causes component libraries to be deployed as part of the Web application when you publish it to the server. Before you can perform this procedure, you need to have Tomcat configured on the local machine and defined as a server in webMethods CAF, as described in "Configuring the Apache Tomcat Servlet Container" on page 54.



1 In Designer:
File > New > Web Application Project.

If you do not see this menu item, try New > Other.

- 2 On the Select a Wizard page, expand webMethods > Composite Applications > Web Application Project, and click Next.
- 3 In the Project Name field of the New Dynamic Web Project page, type a name for the Web application.

The project name cannot have spaces or any of the following characters:

- 4 From the Target Runtime list, choose Apache Tomcat.
- 5 From the Configuration list, choose the CAF Web Application.
- 6 From the Project Template list, choose a project template to be used for the Web application and click Next.

For information on project templates, see "Project Templates" on page 40.

- 7 On the Project Facets page, click Next.
- 8 On the Web Module page, click Next.
- 9 For JSF Libraries on the JSF Capabilities page, choose the CAF Base JSF Runtime option and select the Deploy check box.

This choice causes the CAF Base JSF Runtime component library to be deployed with the Web application when you publish it to Tomcat. To deploy the component libraries to the Tomcat servlet container, see "Deploying Component Libraries on an Application Server" on page 62. To stop deploying the libraries with the Web application, see "Modifying How Component Libraries are Deployed" on page 39.

10 Under Component Libraries, move CAF Base Controls to the right box and select the Deploy check box.



Note: Do not include the MwS Controls or CAF Portlet Controls libraries. They are suitable only for portlet application development.

- 11 Click Finish.
- 12 To associate the Web application you have created with the Tomcat servlet container, see "Associating Web Applications with an Application Server" on page 34.

The Web application project is created. As you develop the Web application, any additional libraries you need are automatically added. To add or remove libraries manually, see "Modifying How Component Libraries are Deployed" on page 39.

Related Topics

- "Configuring the Apache Tomcat Servlet Container" on page 54
- "Modifying How Component Libraries are Deployed" on page 39
- ▶ "Deploying Component Libraries on an Application Server" on page 62
- ▶ "Associating Web Applications with an Application Server" on page 34

Creating Web Applications for JBoss

The following procedure describes how to create a Web application for use with the JBoss application server. This procedure causes component libraries to be deployed as part of the Web application when you publish it to the server. Before you can perform this procedure, you need to have JBoss configured on the local machine and defined as a server in webMethods CAF, as described in "Configuring the JBoss Application Server" on page 57.



To create a Web application project for JBoss

- 1 In Designer:
 File > New > Web Application Project.
 - If you do not see this menu item, try New > Other.
- 2 On the Select a Wizard page, expand webMethods > Composite Applications > Web Application Project, and click Next.
- 3 In the Project Name field of the New Dynamic Web Project page, type a name for the Web application.

The project name cannot have spaces or any of the following characters:

* | \ : " <> . / ?

- 4 From the Target Runtime list, choose JBoss.
- 5 From the Configuration list, choose the CAF Web Application.
- 6 From the Project Template list, choose a project template to be used for the Web application and click Next.
 - For information on project templates, see "Project Templates" on page 40.
- 7 On the Project Facets page, click Next.
- 8 On the Web Module page, click Next.
- 9 For JSF Libraries on the JSF Capabilities page, choose the CAF Base JSF Runtime option and clear the Deploy check box.

This choice causes the CAF Base JSF Runtime component library to be associated with the Web application but not deployed with the application when you publish it to JBoss. The libraries are already deployed on the server.



Note: The content of this step applies to JBoss version 4.2 and later. For earlier versions of JBoss, you need to deploy CAF Base JSF Runtime with the Web application or as a shared library (described in "Creating Web Applications for Tomcat" on page 26).

10 Under Component Libraries, move CAF Base Controls to the right box and select the Deploy check box.



Note: Do not include the MwS Controls or CAF Portlet Controls libraries. They are suitable only for portlet application development.

The links at the bottom of this page show how to deploy the component libraries to the JBoss application server and how to stop deploying the libraries with the Web application.

- 11 Click Finish.
- 12 To associate the Web application you have created with the JBoss application server, see "Associating Web Applications with an Application Server" on page 34.

The Web application project is created. As you develop the Web application, any additional libraries you need are automatically added. To add or remove libraries manually, see "Modifying How Component Libraries are Deployed" on page 39.

Related Topics

- "Configuring the JBoss Application Server" on page 57
- "Modifying How Component Libraries are Deployed" on page 39

- "Deploying Component Libraries on an Application Server" on page 62
- "Associating Web Applications with an Application Server" on page 34

Creating Web Applications for Multiple Server Types

If you intend to create a Web application for use with multiple servers, one way to approach it is to act as though you are creating the project for Tomcat, but deploying all libraries to the server. Before you can perform this procedure, you need to have the application servers configured on the local machine and defined as a servers in webMethods CAF, as described in "Installing and Configuring an Application Server" on page 54.



To create a Web application project for multiple server types

1 In Designer:

File > New > Web Application Project.

If you do not see this menu item, try New > Other.

- On the Select a Wizard page, expand webMethods > Composite Applications > Web Application Project, and click Next.
- 3 In the Project Name field of the New Dynamic Web Project page, type a name for the Web application.

The project name cannot have spaces or any of the following characters:

- 4 From the Target Runtime list, choose Tomcat.
- 5 From the Configuration list, choose the CAF Web Application.
- From the Project Template list, choose a project template to be used for the Web application and click Next.

For information on project templates, see "Project Templates" on page 40.

- 7 On the Project Facets page, click Next.
- 8 On the Web Module page, click Next.
- 9 For JSF Libraries on the JSF Capabilities page, choose the CAF Base JSF Runtime option and clear the Deploy check box.

This choice causes the CAF Base JSF Runtime component library to be associated with the Web application but not deployed with the application when you publish it to the server.

10 Under Component Libraries, move CAF Base Controls to the right box and clear the Deploy check box.



Note: Do not include the MwS Controls or CAF Portlet Controls libraries. They are suitable only for portlet application development.

This choice causes the CAF Base Controls component library to be associated with the Web application but not deployed with the application when you publish it to the server.

- 11 Click Finish.
- 12 To associate the Web application you have created with the application server, see "Associating Web Applications with an Application Server" on page 34.
- 13 Deploy the component libraries on each of the application servers as described in "Deploying Component Libraries on an Application Server" on page 62.

You do not need to perform this step for My webMethods Server, as all component libraries are already deployed to it.

The Web application project is created. As you develop the Web application, any additional libraries you need are automatically added. To add or remove libraries manually, see "Modifying How Component Libraries are Deployed" on page 39.

Related Topics

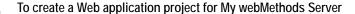
- ▶ "Installing and Configuring an Application Server" on page 54
- "Modifying How Component Libraries are Deployed" on page 39
- "Deploying Component Libraries on an Application Server" on page 62
- "Associating Web Applications with an Application Server" on page 34

Creating Web Applications for My webMethods Server

The following procedure describes how to create a Web application for use with My webMethods Server. You do not need to deploy component libraries as part of the Web application because they already exist on the server.



Note: To use My webMethods Server with webMethods CAF, you need to have selected My webMethods Server Support at the time you installed Designer.



1 In Designer:

File > New > Web Application Project.

If you do not see this menu item, try New > Other.

- 2 On the Select a Wizard page, expand webMethods > Composite Applications > Web Application Project, and click Next.
- 3 In the Project Name field of the New Dynamic Web Project page, type a name for the Web application.

The project name cannot have spaces or any of the following characters:

- 4 From the Target Runtime list, choose My webMethods Server.
- 5 From the Configuration list, choose the CAF Web Application.
- From the Project Template list, choose a project template to be used for the Web application and click Next.

For information on project templates, see "Project Templates" on page 40.

- 7 On the Project Facets page, click Next.
- 8 On the Web Module page, click Next.
- 9 For JSF Libraries on the JSF Capabilities page, choose the CAF Base JSF Runtime option and clear the Deploy check box.
- 10 Under Component Libraries, move CAF Base Controls to the right box and clear the Deploy check box.

This choice causes the CAF Base Controls component library to be associated with the Web application but not deployed with the application when you publish it to My webMethods Server. The libraries are already deployed on the server.

- 11 Click Finish.
- 12 To associate the Web application you have created with My webMethods Server, see "Associating Web Applications with an Application Server" on page 34.

The Web application project is created. As you develop the Web application, any additional libraries you need are automatically added. To add or remove libraries manually, see "Modifying How Component Libraries are Deployed" on page 39.

Related Topics

- "Modifying How Component Libraries are Deployed" on page 39
- "Associating Web Applications with an Application Server" on page 34

Adding a View File to a Web Application

A view file represents a page in a Web application. To add a view file, follow these steps.



To add a view file to a Web application

1 In Designer: File > New > Web Application View.

If you do not see this menu item, try New > Other.

2 On the JSF Application View File page, locate the Web application project and click the WebContent node to make sure the view file is created inside it.

You can create subfolders within WebContent to hold the view file (right-click WebContent and the click New > Folder) but do not place any files within the META-INF or WEB-INF nodes.

- 3 In the File Name field, type the name of the view file, which must have an extension of .view, and click Next.
- 4 On the Portlet View Options page, choose the Form template from the Template list.
- 5 Leave the Support Application Navigation option selected.

This option is required to make the view file able to participate in the Web application navigation model.

- 6 Click Next.
- From the Managed Bean Scope list on the Managed Bean panel, choose the scope for the managed bean. The default is session.

For descriptions of JSF scopes, see "JSF Scopes" on page 83.

8 Click Finish.

The design canvas, a JSF graphical editor, is displayed.

Adding a Control to a View File

In the webMethods CAF, you use the design canvas to place the elements that make up the page. The page elements are JSF controls that you drop from the Palette view.

When you drop a control onto the design canvas, the canvas ("Using the Design Canvas" on page 97) provides clues to valid drop points. Some controls have facets into which you can drop other controls and some controls require child controls to function properly.



To add controls to a view file

- 1 If you do not have the view file open in the design canvas, in the Solutions view double-click the view file.
- 2 Click the Palette view ("Using the Palette View" on page 138) to bring it to the front.
- 3 On the Palette view, expand the nodes to display the controls they contain. For example, click the Output node to display the Text control.
- 4 Drag a control until it highlights the view root (or a Form control if you have used the Form template).
 - For example, drag the Text control and drop it into the target parent control in the view file.
- With the new control selected in the view file, in the Properties view ("Using the Properties View" on page 145) make changes to the properties of the control.

For example, with the Text control in the form selected, locate the Value property in the Properties view and type this text:

```
Hello World
```

When you click away from the Value property, the text appears inside the Text control in the form.

To test the result, publish the portlet to the application server, described in "Publishing a Web Application to the Server" on page 34.

Related Topics

- "Using the Design Canvas" on page 97
- ▶ "Using the Palette View" on page 138
- ▶ "Using the Properties View" on page 145

Associating Web Applications with an Application Server

During Web application development, you publish the project to an application server to perform a live preview. Before you can publish a Web application, you must associate it with (add it to) the server by doing the following:



To add a Web application to an application server

- 1 In the Servers view, right-click the server and then click Add and Remove Projects.
- 2 To add the portlet application to the server, move it to the Configured projects box and click Finish.

If the server is running, Designer publishes the portlet application.

When you publish to the server, and have multiple portlet applications in the Configured projects box, all portlet applications are published. To improve performance, you can add and remove portlet applications as needed.

Related Topics

▶ "Publishing a Web Application to the Server" on page 34

Publishing a Web Application to the Server

When you publish a Web application to an application server, you make the Web application available for use on the server. When you make changes to a Web application in the webMethods CAF, you need to republish the Web application to the server. Before you can use the Servers view to publish a Web application, you must associate it with the server, described in "Associating Web Applications with an Application Server" on page 34.



To publish a Web application to an application server using the Servers view

- In the Servers view, if the server shows a status of Stopped, select the server instance and, in the toolbar of the Servers view, click Start the server to begin communication with the server.
- 2 On the toolbar of the design canvas, click 🚺 .
- If a Define a New Server page appears, specify the server instance on which to run the application.

If you select the Always use this server when running this project option, webMethods CAF will always use this server. To change this setting, on the Project Explorer view, right-

click the Web application and click **Properties**. Click **Server** properties, choose a new server or <None>, and click **OK**.

The Web application is now available for use on the server. If you already have the Web application displayed on a server page, you may need to refresh the page to see changes to application.

Chapter 2. Developing Web Applications

Component Libraries	. 38
Project Templates	. 40
Style Sheets in Web Applications	. 41
Importing a View	. 43
Page Templates	. 43
Navigation in Web Applications	. 47
Application Initialization Parameters	. 50
Environment Variables	. 51

Component Libraries

The controls that are displayed on the Palette view ("Using the Palette View" on page 138) are made up of component libraries. The webMethods UI Developer perspective uses the controls in these component libraries:

- ▶ "CAF JSF Base Runtime Control Library" on page 181
- ▶ "CAF Base Control Library" on page 182

The libraries need to be associated with the Web application at design time and present on the application server at run time. There are two ways to make these component libraries available on the server, include them in the WAR file for the Web application or deploy them to the server itself. By deploying component libraries to an application server, one set of libraries can be shared by all Web applications using the server (see "Deploying Component Libraries on an Application Server" on page 62).

Related Topics

- ▶ "Adding Component Libraries to a Web Application" on page 38
- "Removing Component Libraries from a Web Application" on page 39
- "Modifying How Component Libraries are Deployed" on page 39

Adding Component Libraries to a Web Application

Designer automatically adds component libraries to a Web application in certain cases. For example, if you add a Web service client to a view file, Designer adds a Web service library to the application. If a component library has not been added previously, you can add it manually.



To add a component library to a Web application manually

- 1 On the Project Explorer view, right-click the top node of the Web application project and click Properties.
- On the Properties page, choose Java Build Path, click the Libraries tab to bring it to the front, and click Add Library.
- 3 On the Add Library wizard, click JSF Libraries and click Next.
- 4 Select the check box for any component library to be added to the Web application, and click Finish.



Note: Do not include the MwS Controls or CAF Portlet Controls libraries. They are suitable only for portlet application development.

Removing Component Libraries from a Web Application

To remove a component library from a Web application, do the following:



To remove a component library from a Web application

- 1 On the Project Explorer view, right-click the top node of the Web application project and click **Properties**.
- 2 On the Properties page, choose Java Build Path and click the Libraries tab to bring it to the front.
 - If a component library is listed, that library is associated with the application.
- 3 Select one or more component libraries to be removed from the Web application, and click Remove.
- 4 Click OK.

Modifying How Component Libraries are Deployed

If you deploy the libraries to the application server as shared components, you can clear the Deploy check boxes for the component libraries when you create the Web application project, as described in "Creating a Web Application" on page 25. After a Web application project has been created, you can still specify whether or not to deploy the component libraries in the WAR file.



To modify how component libraries are deployed for a web application library

- 1 On the Project Explorer view, right-click the top node of the Web application project and click **Properties**.
- 2 On the Properties page, choose J2EE Module Dependencies.
- 3 On the J2EE Module Dependencies page, to modify whether or not a component library is deployed in a Web application WAR file, do one of the following:
 - To stop the inclusion, clear the check box at the left edge.
 - To include the component library, select the check box at the left edge.



Note: Do not include the MwS Controls or CAF Portlet Controls libraries. They are suitable only for portlet application development.

4 Click OK.

Project Templates

A project template is a preconfigured Web application that provides a starting point from which to develop a full application. While a project template is not intended as a sample, it can give you an idea of how a Web application goes together.

You apply a project template at the time you create a new Web application project, as described in "Creating a Web Application" on page 25. On the Project Template list, select Starter Web Application.

The Starter Web Application project template contains the following views:

- default.view The home page for the Web application. This page imports the header.view file
- header.view The view that contains navigation and other elements that should appear on each page in a Web application. This view is imported into default.view and would also be imported into any new views created for the Web application.
- login_error.view The page to which users are redirected if they fail login authentication.
- login.view The login page to which users are redirected if they attempt to open the default.view file.

The Starter Web Application project template also includes the styles.css. For more information, see "Style Sheets in Web Applications" on page 41.



To demonstrate this project template

- 1 Make sure you have security configured for the application server you are running, as described in "Web Application Security" on page 58.
- 2 Make sure the Web application that contains the project template is associated with the application server, as described in "Associating Web Applications with an Application Server" on page 34.
- 3 On the Solutions view, double-click the default.view file to display it on the design canvas.
- 4 On the toolbar of the design canvas, click 🚺 .
 - The default view is published to the server, but you are redirected to the login page.
- 5 Type the default user name and password for the application server and click **OK**. The default page is displayed, including anything that exists on the header.view file.

Style Sheets in Web Applications

When you create a Web application, you have to provide control over the appearance of the page being displayed by the application server. You can control the appearance of individual controls through the Properties view or you can provide overall control for the entire application using an external stylesheet. In both cases, you are using a Cascading Style Sheet (CSS).

Related Topics

- ▶ "Applying CSS to an Individual Control" on page 41
- "Applying CSS to a View File" on page 42

Applying CSS to an Individual Control

You can apply styles to an individual control in either of two ways, apply styles directly from the design canvas or from within the Properties view, as described below. CSS values applied to an individual control have precedence over a CSS applied to the entire page ("Applying CSS to a View File" on page 42).

Changing Styes from the Design Canvas

You can edit CSS values for controls without having to know CSS syntax. On the design canvas, right-click a control and then click Style. The resulting menu offers these choices:

Menu item	Description
Edit Style	Displays the CSS Style Definition editor. Use this editor to set values for a variety of CSS settings.
Reset Style	Click to reset the control to default CSS settings.
Style Classes	Choose a preset CSS style class. This feature requires that you have created an external CSS and used an Include Stylesheet control to associate it with the view. See "Include Stylesheet" on page 378.
Color	From the menu, choose a color to be used for foreground objects. The choices are a subset of colors available in the CSS Style Definition editor.
Background Color	From the menu, choose a color to be used for the background. The choices are a subset of colors available in the CSS Style Definition editor.

Changing Styes from the Properties View

You can edit CSS values for controls from the Properties view. With this approach, you need to have some knowledge of CSS to proceed.



- 1 On the design canvas, select the control you want to modify.
- 2 On the Properties view, bring the Display tab to the front.
- 3 On the CSS Style property, click ... to open the CSS editor.
- 4 In the Edit CSS dialog, type CSS entries using valid syntax and click **OK**.
- 5 Click the Preview tab on the bottom of the design canvas to check the result of your modifications.

Applying CSS to a View File

You can apply an external stylesheet to be used by all controls on a page. If you apply the stylesheet to header, or other view that is imported into all pages of a Web application, you can provide a common style to all pages in the application. See "Importing a View" on page 43. For an example, see the Starter Web Application project template you can create as part of a new Web application project. By default, the login.view includes the external stylesheet styles.css.

The external CSS can located at a URL where it can be accessed by the application server at runtime, or you can include it in the Web application project.



- 1 If you are going to include the external stylesheet in the Web application project, do so in either of these two ways:
 - Copy the external CSS file:
 - 1 Copy the CSS file into the Eclipse workspace, somewhere inside the WebContent directory for the Web application project.
 - 2 On the Project Explorer view, right-click the WebContent node and click Refresh.
 - Create a file in the Web application project:
 - 1 On the Project Explorer view, right-click WebContent and New > File.
 - In the File name field, type a file name with a .css extension and click Finish.

 The file opens in a CSS editor.
 - 3 Type the contents in the CSS editor or copy them in from another file.
- 2 On the Project Explorer or Solutions view, double-click the view file to display it on the design canvas.

- 3 On the Project Explorer view, drag the CSS file to the design canvas. webMethods CAF automatically adds an "Include Stylesheet" on page 378 control and binds it to the CSS file.
- 4 Click the Preview tab on the bottom of the design canvas to check the result of your modifications.

Importing a View

The "Import View" on page 418 control allows you to import one view into another view. By doing so, you can interact with controls in the imported view as if they are present in the current view. A common use for this capability is the creation of a header to be used for all pages in a Web application. For an example, see the Starter Web Application project template you can create as part of a new Web application project. By default, the header view is imported into the default view.

To import a view into another view, do the following:



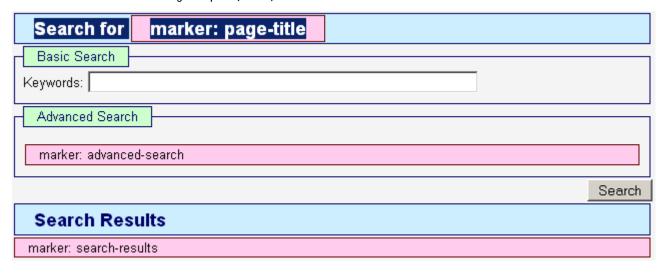
To import a view

- 1 On the Project Explorer or Solutions view, double-click the target view file to display it on the design canvas.
- On the Project Explorer view, drag the view to be imported to the design canvas. webMethods CAF automatically adds an "Import View" on page 418control and binds it to the target view file.

Page Templates

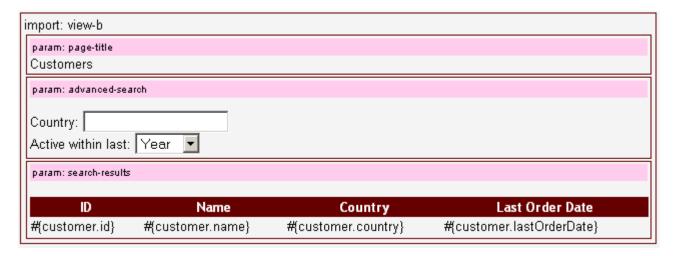
You can use the Import View control to import a template view (see "Importing a View" on page 43), where the imported view contains incomplete sections (markers) into which the import control injects content specified on the importing view. A template view is a regular view used for the purpose of templating, and a marker control is any control (often a Panel control) used to mark a section to be replaced. You can specify the content to be injected by means of "Content Parameter" on page 416 children of the Import View control. For example, the following template view (View B) contains markers for the page title, the advanced-search property group, and the search-results table:

Search Page Template (View B)



The top-level instance view (View A) imports the template view, and specifies the content for the page title, advanced-search, and search-results sections:

Search Page Results (View A)



The name of a "Content Parameter" on page 416 in the Import View control should be the ID (or "Control ID Reference" on page 192) of the control in the imported view the Content Parameter's content will replace. If there is no Content Parameter present in the Import View control for a marker in the imported view (for example, if View A above did not have a search-results Content Parameter), the marker's content is rendered as is. In other words, the marker's content is the default content displayed if no Content Parameter for the marker has been specified.

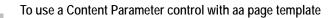
You can use UIParameter controls (Control Parameters) in an Import View control to set property values on the imported view's page bean. The name of the UIParameter specifies the name of the page bean property, and the value specifies the value to which the property is set. In the above example, if View B's page bean had a property called pageTitle, a UIParameter with the name pageTitle could be used by View A to set that property (in the above example, it might set the property's value to "Customers"). View B could then use the page bean property to display the pageTitle, or use it for some other purpose, such as the input to a Web service.

Related Topics

- "Page Templates and Content Parameters" on page 45
- ▶ "Page Templates and Control Parameters" on page 46

Page Templates and Content Parameters

As described in "Page Templates" on page 43, you can use one or more "Content Parameter" on page 416 controls as children of an "Import View" on page 418 control to inject content to from the importing view to a page template when the page is displayed.



- On the Solutions view, double-click the page template view to display it on the design canvas.
- 2 On the design canyas, select the control that is to be used as a marker.
- 3 On the General tab of the Properties view, type a unique ID in the ID field.
- 4 On the Solutions view, double-click the target view to display it on the design canvas.
- 5 Place an Import View control on the canvas, as described in "Importing a View" on page 43.
- 6 From the Palette view, drag a Content Parameter control (Control > Logic > Content Parameter inside the Import View.
- With the Content Parameter control selected on the design canvas, in the Name field of the Properties view, type the same value you used for the ID in step 3.



Page Templates and Control Parameters

As described in "Page Templates" on page 43, you can control parameters to set property values on the imported view's page bean when the page is displayed. The following example uses the Starter Web Application project template do demonstrate how to use a control parameter to create a page title that differs, depending on which page in the Web application is displayed.



To use a Control Parameter control with a page template

- 1 Create a new Web application project using the Starter Web Application project template, as described in "Creating a Web Application" on page 25.
- 2 On the Solutions view, double-click the header view to display it on the design canvas.
- 3 From the Palette view, drag a Header control (Control > Output > Header) to the design canvas, after the Static Menus control.
- 4 From the Palette view, drag a String (Data > Primitives > String) to the Bindings view, onto the header managed bean.
- 5 On the Add New Property dialog, type pageTitle in the Property Name field.
- 6 Select the Property is writable option and click Finish.
 - The Page Title property is added to the Bindings view.
- 7 From the Bindings view, drag the Page Title property to the Header control on the design canvas.
- With the Header control selected, the **Value** field of the Value tab should contain the expression #{HeaderView.pageTitle}.
- 9 On the Solutions view, double-click the default view to display it on the design canvas.
- 10 Select the Import View control, and on the General tab of the Properties view add a new control parameter by doing the following:
 - a In the Control Params area, click 💠 Add.
 - Double-click the control parameter you have just added.
 The Properties view now displays properties for the control parameter.
 - c In the Name field, type the name of the property you created in step 5, in this example the name is pageTitle.
 - d In the Value field, type a tile to appear whenever the default view is displayed.
- 11 Repeat steps 9 and 10 for each view file that uses the Import View control to import the contents of the header view file.

Navigation in Web Applications

In Web applications, the developer is responsible for providing all page navigation within the application. webMethods CAF provides these controls:

This control	Displays Web application
"Breadcrumbs" on page 626	Breadcrumbs navigation as a list of links, starting with the home page. See "Hiding Breadcrumbs on the Home Page" on page 49.
"Popup Menus" on page 627	Top-level navigation as a list of links. Clicking on or hovering over one of the top-level links displays the children of that page as a popup menu of second-level links. These menus do not include the home node.
"Static Menus" on page 629	Top-level navigation (and optionally second- and third-level pages) as a list of links. These menus do not include the home node.
"Toggle Menus" on page 631	Top-level navigation as a list of links. Clicking or hovering over one of the top-level links displays the children of that page as a list of second-level links. These menus do not include the home node.

The hierarchy for each of these controls is provided by an application navigation (appNav) bean. Once established, all navigation controls in the Web application use the same appNav bean. You use the Application Navigation Configuration Editor to associate the menus with the pages in the application.

Related Topics

- ▶ "The Application Navigation Configuration Editor" on page 47
- "Adding a Home Link to a Web Application" on page 49
- "Hiding Breadcrumbs on the Home Page" on page 49

The Application Navigation Configuration Editor

You use the Application Navigation Configuration Editor to associate the page navigation controls with the pages in a Web application. When you design a Web application, you usually have a hierarchy in mind. For example, you may have a tree hierarchy with a home page and child pages, which in turn have their own children. You use the Application Navigation Configuration Editor to design this structure, which is exposed though the AppNavBean. The various navigation controls use the AppNavBean to draw all or portions of the navigation.

The following procedure assumes you have already created multiple pages that need to be linked.



To create an application navigation configuration

- 1 On the Solutions view, under the node for the Web application, double-click the Web Navigation node.
- On the Web Application Navigation editor, click the Configuration tab at the bottom. By default, the Home node is provided. The ID field shows which is the Home node.
- 3 To add a navigation node, do the following:
 - a With the parent node selected (Home in the initial case), click Add.
 - The editor adds a new node under the parent node, and gives it an arbitrary title. You can also drag a view from the Solutions view to the node in the editor that is to be its parent.
 - b Select the new navigation node and make changes in Navigation Page Details as necessary:
 - In the Title field, type a new title for the node.
 - In the ID field, click Browse, select the view for this node, and click OK.
 - (Optional) In the URL field, click Browse, select the view for this node, and click OK. Values take the form /project/view. You can also type an external URL.
 - (Optional) In the Resource Key field, if you have set a resource key in the view file, type the resource key. The value of the resource key is used in place of the text in the Title field.

There are some other actions you can perform on the Application Navigation Configuration Editor:

- To delete a node, select that node and click Delete.
- To move a node within a parent node, select the node and click Up or Down.
- To move a node from one parent node to another, drag the node. When you drag a node, all of its child nodes move with it.

Adding a Home Link to a Web Application

By adding a link pointing to the home page to a header file that is imported into all pages in the Web application, you can provide navigation back to the home page from anywhere in the application. You can use the appNav bean for this purpose. The following procedure assumes you have already created the application navigation configuration as described in "The Application Navigation Configuration Editor" on page 47.



To add a link to the home page on all pages in a Web application

- 1 On the Solutions view, double-click the header view to display it on the design canvas.
 - In the Starter Web Application project template, this is the header.view file.
- 2 From the Palette view, drag a Link control (Control > Output > Link) into the header view file.
- With the Link control selected on the design canvas, on the Value tab of the Properties view click #{...} for the Value field.
- 4 In the Expression Binding dialog, select App Nav Bean > Root > Url and click OK.

 The control is now bound to the root view as defined in the appNav bean.

Hiding Breadcrumbs on the Home Page

In conjunction with the appNav bean, the "Breadcrumbs" on page 626 control provides three levels of navigation, starting from the home page. Before you can take advantage of breadcrumbs, you need to create the application navigation configuration, as described in "The Application Navigation Configuration Editor" on page 47.

While breadcrumbs are useful for navigation when the user is somewhere inside the hierarchy of Web pages, they are useless at the top level. The following procedure describes how to place a Breadcrumbs control on a page and then hide the control when the home page is displayed.



To hide the breadcrumbs control when the home page is displayed

1 On the Solutions view, double-click the header view to display it on the design canvas.

In the Starter Web Application project template, this is the header.view file.

- 2 From the Palette view, drag a Link control (Control > Webapp > Breadcrumbs) into the header view file.
 - If you have a valid application navigation configuration, breadcrumbs will be displayed on any page that imports the header view.
- If you don't already know it, determine the home page by doing the following:
 - a On the Solutions view, under the node for the Web application, double-click the Web Navigation node.
 - b On the Web Application Navigation editor, click the Configuration tab at the bottom and select the home page.
 - Under Navigation Page Details, locate the ID field.
 This is the home page for the Web application. If you copy the contents of this field, you can paste it in a later step.
- With the Breadcrumbs control selected on the design canvas, on the General tab of the Properties view click #{...} for the Rendered field.
- 5 In the Expression Binding dialog, select App Nav Bean > Current > ID.

The resulting binding expression looks like this:

```
#{header_view.appNavBean.current.id}
```

6 Edit the binding expression to state that the Breadcrumbs control is to be rendered if the ID of the current view is not that of the home page:

```
#{header_view.appNavBean.current.id !="/home.view"}
```

7 Click OK.

Application Initialization Parameters

Application initialization parameters are variables that can be added as default values in a Web application. The application initialization parameters are read-only values that are used at run time, but the server administrator can alter them as needed. You can use application initialization parameters to customize the behavior of a Web application, either to display the value or use it in a binding expression. Application initialization parameters can only be strings and are valid only within a Web application. For variables that can span Web applications, see "Environment Variables" on page 51.

Some application initialization parameters are included in webMethods CAF, and are based on J2EE standards. For an example of a pre-configured application initialization parameter, see "Making the Tomcat Server Re-render the Page Automatically" on page 63.

To create an application initialization parameter for a Web application

- 1 With a view from the Web application displayed on the design canvas, on the Bindings view expand Implicit Variables > Application Initialization Parameters.
- 2 Right-click Application Initialization Parameters and click Add > Data.
- In the Property Name field of the Add New Property dialog, type a name for the initialization parameter and click Finish.
 - The new application initialization parameter appears in the Bindings view under the Application Initialization Parameters node.
- 4 Double-click the new application initialization parameter and then click the Properties view to bring it to the front.
- 5 On the Properties view, click the Data Binding tab to bring it to the front and, in the Value field type a value for the parameter.

You can now drag this parameter from the Bindings view to a control on the design canvas to bind the control to the parameter.

Environment Variables

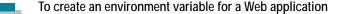
Environment variables are variables that can be added as default values in a Web application. The environment variable values are used at run time, but the server administrator can alter them as needed. You can use environment variables to customize the behavior of a Web application, either to display the value or use it in a binding expression. Unlike application initialization parameters ("Application Initialization Parameters" on page 50), environment variables can be typed (primitive and object data types) and can be shared among Web applications.



Note: For primitive data types, you can specify a default value, and it can be changed at run time either programatically or by an administrator. Object data types must be programatically put into the JNDI tree.

Environment variables operate in the Java Naming and Directory Interface (JNDI) environment. While use of JNDI is widespread in application servers, you should make sure a specific application server supports JNDI before attempting to deploy Web applications that use environment variables. Environment variables can be programmatically updated by the current Web application, or by another application, providing a way to share services between Web applications in the same container.

Common configuration parameters to put into environment variables are anything that may differ for an application deployed into different environments (for example between Development, Staging and Production). For example when adding a Web service connector, it automatically adds common Web service configuration values like endpoint, username, and password to the application environment variables.



- 1 With a view from the Web application displayed on the design canvas, on the Bindings view expand Implicit Variables > Environment Variables.
- 2 Right-click Environment Variables and click Add > Data.
- 3 In the Property Name field of the Add New Property dialog, type a name for the environment variable.
- In the Property Type field, type a primitive or object data type and click Finish.

 The new environment variable appears in the Bindings view under the Environment Variables node.
- 5 Double-click the new environment variable and then click the Properties view to bring it to the front.
- 6 On the Properties view, click the Data Binding tab to bring it to the front and, in the Value field type a value for the variable.

You can now drag this parameter from the Bindings view to a control on the design canvas to bind the control to the parameter.

Chapter 3. Working with Application Servers

Installing and Configuring an Application Server	. 54
Web Application Security	. 58
Deploying Component Libraries on an Application Server	. 62
Making the Tomcat Server Re-render the Page Automatically	. 63
Running Servers in Debug Mode	. 64

Installing and Configuring an Application Server

Software AG does not include any external application servers in its product installations. You are responsible for downloading and installing the application server yourself. In addition, it is assumed that you are responsible for, and knowledgeable about, the configuration of the server in your own environment. The application servers supported for this release are:

- Apache Tomcat Servlet Container 5.5 (version 5.5.23 or later)
- JBoss Application Server 4.2

Related Topics

- ▶ "Configuring the Apache Tomcat Servlet Container" on page 54
- "Configuring the JBoss Application Server" on page 57

Configuring the Apache Tomcat Servlet Container

Before you can create a Web application, you need to configure Tomcat and define it as a server on which applications can be published. For development purposes, the server needs to reside on the local host.

Configuring the Tomcat servlet Container

The following procedure assumes that you have already installed Tomcat on your local host, using your Tomcat documentation for guidance.



- 1 Right-click inside the Servers view, and click New > Server.
- 2 In the Server's host name field of the New Server wizard, type localhost.
- 3 Under Select the server type, expand the Apache server type, select the version of the installed server, and click Next.
- 4 Click Browse and navigate to the installation directory of the server.
- 5 If necessary, choose the JRE to be used by the server.
- 6 If you have an existing Web application to be added to the server, move it to the Configured projects box.

You can perform this action at a later time, as described in "Associating Web Applications with an Application Server" on page 34.

- 7 On the Select Tasks page, make sure the Deploy wm_cafshared.war option is selected and click Finish.
 - The wm_cafshared.war file contains JavaScripts, cascading stylesheets, images, and other resources needed for CAF controls to function. This option verifies that the WAR file is deployed to the server.
- 8 If you plan to use security constraints on your Web application, do the following:



Note: If you perform the following steps, you allow a loose interpretation of the auth-constraint element as described in the Java Servlet Specification. While this action makes it easier to begin development activities, you may want to adopt a strict interpretation for a production environment.

- a On the Project Explorer view, expand Servers > Tomcat... and double-click the server.xml file to open the editor.
- b On the server.xml editor, expand Server > Service > Engine > Realm.
- c Right-click the Realm node and Add Attribute > New Attribute.
- d In the Name field of the New Attribute dialog, type allRolesMode.
- e In the Value field, type strictAuthOnly.
- 9 If you want the component libraries to be shared on the Tomcat server, as opposed to deploying the libraries with each Web application, deploy the libraries to the server, as described in "Deploying Component Libraries on an Application Server" on page 62.

Deploying Shared Libraries to the Tomcat Installation Directory

By default, webMethods CAF creates an instance of the Tomcat servlet container in the Eclipse workspace. This practice means the Tomcat (standalone) installation is not affected by changes made in webMethods CAF. If you want to deploy shared component libraries to the Tomcat installation directory, do the following:



To manually deploy shared libraries to the Tomcat installation directory

- 1 On the Servers view, right-click the Tomcat server and then click CAF Tools > Deploy Shared Libraries.
- 2 On the Deploy Shared Libraries dialog, select these two component libraries:
 - CAF JSF Base Runtime Control Library CAF Base Control Library

3 For Destination, click Browse, move to the Tomcat installation directory shown here, and click OK:

Tomcat_install_dir\shared\lib

If this directory does not exist, you need to create it.

4 Click OK.

Running Tomcat in the Installation Directory

Be default, webMethods CAF places an instance of the Tomcat server in the Eclipse workspace and uses that instance for development. Doing so leaves the Tomcat installation unaltered. If you prefer to run the actual Tomcat installation, you can do so.



To cause webMethods CAF to run Tomcat in the installation directory

- 1 If you have any Web applications associated with Tomcat, you need to remove them by doing the following:
 - a On the Servers view, right-click the Tomcat server and click **Add and Remove Projects**.
 - b Click Remove All and click Finish.
 - On the Servers view, the value in the **Status** column for the Tomcat server now shows **Restart**.
 - c On the toolbar of the Servers view, click **()** Start the server.
- 2 On the Servers view, double-click the Tomcat server.
 - The Tomcat overview is displayed.
- 3 In the Server Locations area, choose the Use Tomcat installation option.

Related Topics

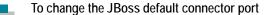
- "Security on Tomcat" on page 58
- "Deploying Component Libraries on an Application Server" on page 62
- "Making the Tomcat Server Re-render the Page Automatically" on page 63
- "Running Servers in Debug Mode" on page 64

Configuring the JBoss Application Server

Before you can create a Web application, you need to configure JBOSS and define it as a server on which applications can be published. For development purposes, the server needs to reside on the local host.

The following procedures assume that you have already installed JBoss on your local host, using your JBoss documentation for guidance.

Before you configure the JBoss application server for use with webMethods CAF, you may want to change the connector port. By default, JBoss uses the common port number 8080, which may conflict with another server on your machine.



- 1 Locate the following file in the JBoss installation directory:
 [Boss_install_dir\server\default\deploy\jboss-web.deployer\server.xml
- Open the server.xml file in a text editor and locate this line: <Connector port="8080" address="\${jboss.bind.address}"</p>
- 3 Edit the port number as appropriate and save the file.

To configure and define a JBoss application server in Eclipse

- 1 Right-click inside the Servers view, and click New > Server.
- 2 In the Server's host name field of the New Server wizard, type localhost.
- 3 Under Select the server type, expand the JBoss server type, select the version of the installed server, and click Next.
- 4 Click **Browse** and navigate to the installation directory of the server.
- 5 If necessary, choose the JRE to be used by the server.
- 6 If you have an existing Web application to be added to the server, move it to the Configured projects box.
 - You can perform this action at a later time, as described in "Associating Web Applications with an Application Server" on page 34.
- 7 On the Select Tasks page, make sure the Deploy wm_cafshared.war option is selected and click Finish.
 - The wm_cafshared.war file contains JavaScripts, cascading stylesheets, images, and other resources needed for CAF controls to function. This option verifies that the WAR file is deployed to the server.

8 If you want the component libraries to be shared on the JBoss server, as opposed to deploying the libraries with each Web application, deploy the libraries to the server, as described in "Deploying Component Libraries on an Application Server" on page 62.

Related Topics

- ▶ "Security on JBoss" on page 60
- "Deploying Component Libraries on an Application Server" on page 62
- "Making the Tomcat Server Re-render the Page Automatically" on page 63
- "Running Servers in Debug Mode" on page 64

Web Application Security

As a developer of Web applications, you are responsible for authorization of users who attempt to view the pages of an application. Each application server has its own mechanisms for security and you should consult the documentation specific to the server you are using. The following links provide you with information on how to configure security for specific application servers while developing applications on webMethods CAF. One thing that is common among application servers is support for security roles, described in "Security Roles for Web Applications" on page 61.

Related Topics

- "Security on Tomcat" on page 58
- "Security on JBoss" on page 60
- "Security Roles for Web Applications" on page 61

Security on Tomcat

When you configure the Tomcat servlet container for use with webMethods CAF, an instance of the server is automatically placed in the Eclipse workspace for Web application. Changes you make to this instance of the server do not affect settings in the Tomcat installation. You can see files associated with the workspace instance of the server under the Servers node in the Project Explorer view. For development and testing, you can define users and the security roles of which they are members in the tomcat-users.xml file. The security roles you use should be the same as those you have defined for use in Web applications, as described in "Web Application Security" on page 58.



Note: The tomcat-users.xml file is only for use in developing Web applications on your machine. For configuring security in a Tomcat production environment, consult the Apache Tomcat documentation.

You can edit the tomcat-users.xml file to add users and security roles to be used in testing. Users and roles are elements (). These elements have one or more attributes (). The following procedures describe how to edit the tomcat-users.xml file in the Design tab. You can also use the Source tab to edit the XML file directly.

To display the tomcat-users.xml file in an editor

- 1 On the Project Explorer view, expand Servers > Tomcat and double-click the tomcatusers xml file.
- 2 On the tomcat-users.xml editor, click the Design tab at the bottom.

If you look in the tomcat-users.xml file, you will see that one of the default user names is tomcat and the password is tomcat.

To add a security role to the tomcat-users.xml file

- 1 Depending on where you want to place the role element, do one of the following:
 - To add a security role as the last element in the tree, right-click the tomcat-users element and Add Child > New Element.
 - To add a security role in a specific place in the tree, right-click an element and Add After > New Element or Add Before > New Element.
- 2 In the Element Name field, type role and click OK.

The role element requires one attribute, rolename.

- 3 Right-click the new role and click Add Attribute > New Attribute.
- 4 In the Name field, type rolename.
- 5 In the Value field, type the name of the role and click **OK**.

To add a user to the tomcat-users.xml file

- 1 Depending on where you want to place the user element, do one of the following:
 - To add a user as the last element in the tree, right-click the tomcat-users element and Add Child > New Element.
 - To add a user in a specific place in the tree, right-click an element and Add After > New Element or Add Before > New Element.
- 2 In the Element Name field, type user and click OK.

The user element requires three attributes: password, roles, and username. When you add an attribute to an element, it is appended at the end of attributes for the element.

- 3 Right-click the new user and click Add Attribute > New Attribute.
- 4 In the Name field, type password.
- 5 In the Value field, type user's password and click OK.

- 6 Right-click the new user and click Add Attribute > New Attribute.
- 7 In the Name field, type roles.
- 8 In the Value field, type the roles of which the user is a member, separated by commas, and click OK.
 - For example if there are two security roles, tomcat and role1, type tomcat, role1.
- 9 Right-click the new user and click Add Attribute > New Attribute.
- 10 In the Name field, type username.
- 11 In the Value field, type user name and click OK.

To edit an attribute

- 1 Right-click the attribute and click Edit.
- 2 Modify the entry in the Value field and click OK.

The values in the Name field should be changed only if they no not match the required entries described above.

To edit an element

You cannot edit an element. You need to remove the element and create a new one.

To remove an element or attribute

Right-click the element or attribute and click **Remove**. Removing an element also removes any attributes that belong to it.

Security on JBoss

For JBoss you have to provide the name of a security domain to determine what security handler to use for authentication. When you associate a Web application project with JBoss ("Associating Web Applications with an Application Server" on page 34), webMethods CAF creates a jboss-web.xml file in which to declare the security domain for use with the Web application.



To display the jboss-web.xml file

- 1 On the Project Explorer view, locate the web application in the tree view and expand it to expose jboss-web.xml:
 - Web_application/WebContent/WEB-INF/jboss-web.xml
- 2 Double-click jboss-web.xml.

If you open this file, you will see the following entry:

<security-domain>java:/jaas/caf</security-domain>

webMethods CAF provides the default caf security domain to be used for initial development, but you can replace it with your own.

Specifying a JBoss Security Domain

You need to specify the security domains to be used with a JBoss application server. To do so, you need to edit the login-config.xml file, located here:

```
|Boss_install_dir\default\conf\login-config.xml
```

To use the caf security domain for development, you need to add the following code to the login-config.xml file. Add this text following the policy> element.

The sample properties files identified in the caf security domain are located here:

```
|Boss_install_dir\ default\conf\props
```

If you look in the jboss-users.properties file, you will see that the default username for the caf security domain is kermit and the password is thefrog.

For more information about using security domains with JBoss, consult your JBoss documentation.

Security Roles for Web Applications

A security role is a standard J2EE concept for determining who can have access to a site, a page, or a control on a page. To assist you, webMethods CAF has mechanisms you can use to add security roles to a Web application.



To add a security role to a Web application

- 1 With a view from the Web application displayed on the design canvas, on the Bindings view, Implicit Variables > Application Security Roles.
- Right-click Application Security Roles and click Add > Application Security Role.
- In the Role Name field, type a valid security role name.
 - The security role appears in the Bindings view under the Application Security Roles node.
- 4 (Optional) In the **Description** field, type a description of the security role.



To apply a security role to a control

From the Bindings view, drag a security role onto a control in the design canvas.

On the General tab of the Properties view, you can see that the Rendered property has become bound to the security role. Only someone logged in as a user who is a member of the role can see the control.

You can also move the security role to a Disabled or Read-only property. Doing leaves the control visible, but if the user is not a member of the security role, the control is read only.

To determine how to define users and security roles on an application server, consult the documentation for the specific server.

Related Topics

- "Security on Tomcat" on page 58
- ▶ "Security on JBoss" on page 60

Deploying Component Libraries on an Application Server

webMethods CAF uses a set of component libraries to contain controls used by Web applications. There are two ways to make these component libraries available for use, include them in the WAR file for the Web application or deploy them to the server itself. By deploying component libraries to an application server, one set of libraries can be shared by all Web applications using the server. webMethods CAF has several component libraries, but only two are required:

- CAF JSF Base Runtime Control Library (Required)
- CAF Base Control Library (Required)
- CAF Glue WS Client (If you are using Web service connectors)
- CAF SQL Client (If you are using database connectors)



To deploy component libraries to an application server

- On the Servers view, right-click the application server and then click CAF Tools > Deploy Shared Libraries.
- 2 On the Deploy Shared Libraries dialog, select these two component libraries:
 - CAF JSF Base Runtime Control Library CAF Base Control Library

- 3 Click OK.
- 4 To start or restart the application server, select the server instance and, in the toolbar of the Servers view, click Start the server.

If you are using the Tomcat servlet container and want to deploy the component libraries to the installed Tomcat server, rather than the server instance in the Eclipse workspace, use the alternate method described in "Deploying Shared Libraries to the Tomcat Installation Directory" on page 55.

Making the Tomcat Server Re-render the Page Automatically

During Web application development, it is desirable to run the application on a server frequently to observe its operation. Doing so means you need to restart the server every time so it will clear out its cache and rebuild the view being displayed. On Tomcat, you can modify a J2EE application initialization parameter that causes the server to rebuild the view and re-render it without have to restart.



- 1 With a view from the Web application displayed on the design canvas, on the Bindings view expand Implicit Variables > Application Initialization Parameters.
- 2 Double-click com.webmethods.caf.faces.view.CACHE.
- On the Data Binding tab of the Properties view, in the Value field type false.

 Thereafter, when you make small changes to layout for a view and republish the view (click on the toolbar for the design canvas), the JSF engine rebuilds the component tree and re-renders the page.

Even with this parameter set to false, larger changes to a view will require that the server be restarted. You can determine what action is required by looking at the Servers view. If the Status column for the Tomcat server shows a status of Restart, click Start the server on the toolbar of the Servers view, and then click on the toolbar of the design canvas.

Running Servers in Debug Mode

If you want to debug Java code, you can run the application server in debug mode. To do so, you must stop the server and restart it in debug mode.



To start the application server in debug mode

- 2 On the toolbar of the Servers view, click 🏇 Start the server in debug mode.

Chapter 4. Getting Started with Portlet Application Development

Introduction to the My webMethods Perspective	66
Creating a Portlet Application	68
Adding a Portlet to a Portlet Application	70
Adding a View File to a Portlet	72
Adding a Control to a View File	73
Connecting to My webMethods Server	74
Associating Portlet Applications with My webMethods Server	76
Running Servers in Debug Mode	76
Publishing a Portlet Application to the Server	77
Adding a Portlet to a My webMethods Server Page	77
Uninstalling Portlet Applications from My webMethods Server	78
Troubleshooting Portlet Applications	79

Introduction to the My webMethods Perspective

A portlet application is composed of one or more portlets, which in turn are made up of one or more views (pages) intended to be displayed by My webMethods Server. Do not use this perspective to create Web applications for application servers (see "Getting Started with Web Application Development" on page 23). To create portlet applications, use the My webMethods perspective.



To open the My webMethods perspective

- 1 To start Designer, Start > Programs > webMethods7 > Tools > Designer.
- 2 In the Workspace Launcher window, accept the default workspace or click Browse and navigate to the folder where you want the Eclipse workspace to reside, and then click OK.

The first time you start Designer in a new workspace, it starts a local Metadata store and installs a light-weight version of the My webMethods Server runtime for use by the Preview Server. By default, Designer opens in the Business Analyst perspective. To create portlet application projects, change to the My webMethods perspective.

- 3 In Designer: Window > Open Perspective > Other > My webMethods.
- 4 Click OK.

The My webMethods perspective contains the following views. For more information on views, see "Views in webMethods CAF" on page 85:

View	Description
Bindings	A tree view that contains all data that can be used within this view or within the entire portlet application.
IS Browser	A tree view displaying the resources on instances of Integration Server to which Designer is connected. You can use this view to select services and create Web service connectors that can be used to interact with Integration Server.
MWS Admin	A tree view displaying the resources on instances of My webMethods Server to which Designer is connected. You can use this view to publish portlets to the servers and reorganize server resources.
Navigator (Eclipse)	A tree view of the resources in the workspace. From here, you can open files for editing or select resources for operations such as exporting.

View	Description
Outline (Eclipse)	An outline of a view that is currently open in the design canvas. This view is useful in locating specific controls within a complex portlet application and for moving controls from one location to another.
Palette	A tree view of controls you can use to build portlet applications. You can drag controls from this view directly to the design canvas.
Problems (Eclipse)	A display of system-generated errors, warnings, or information associated with a resource. For example, if you save a Java source file that contains syntax errors, the errors are automatically are logged in this view.
Properties (Eclipse)	A view of display names and values of properties for a resource selected elsewhere in Designer. This view is useful in changing the values of properties of controls in the design canvas and objects in the MWS Admin view.
Servers (Eclipse)	A table of runtime environments associated with Designer. Once configured, you can publish portlet applications to the server.
Solutions	A tree view that represents a logical representation of the resources in the workspace, that is simpler than the one provided in the Navigator or Project Explorer. You can perform selected operations on resources shown in this view.

In addition to the views, there are various editors:

Editor	Description
Design canvas	A JSF graphical editor in which you build portlet or Web applications by adding controls from the Palette view. This editor provides a logical view; to see how the portlet application appears on a page, you can preview it using Instant Preview or the Preview Server.
Portlet Application Configuration Editor	An editor that allows you to create and manage portlets and other resources belonging to a portlet application.
Java editor	A source code editor that provides specialized features for editing Java code.

Once in the webMethods CAF, you can perform the following activities:

- You can develop tasks or composite applications ("Creating a Portlet Application" on page 68) and test them using the Preview Server ("Live Preview" on page 104), but if you want to test them on a complete instance of My webMethods Server, you can do so as described in "Connecting to My webMethods Server" on page 74.
- Easily publish portlet applications to My webMethods Server using the MWS Admin view, which provides a tree view of resources on one or more instances of My webMethods Server. See "Connecting to My webMethods Server" on page 127.
- Connect to one or more instances of Integration Server, such as to create Web service connectors, using the IS Browser view. See "Connecting to Integration Server" on page 116.

Related Topics

- "Connecting to My webMethods Server" on page 74
- "Creating a Portlet Application" on page 68
- ▶ "Using the MWS Admin View" on page 125
- ▶ "Using the IS Browser View" on page 115
- ▶ "Introduction to the My webMethods Perspective" on page 66

Creating a Portlet Application

If you have the My webMethods displayed ("Introduction to the My webMethods Perspective" on page 66), you are ready to create a portlet application project.

1 In Designer:
File > New > Portlet Application Project.

If you do not see this menu item, try New > Other.

2 In the Project Name field of the New Dynamic Web Project window, type a name for the portlet application.

The project name cannot have spaces or any of the following characters:

- 3 From the Target Runtime list, choose My webMethods Server.
- 4 From the Configurations list, choose CAF Portlet Application.

- From the Project Template list, choose a page template to be used for the portlet application and do one of the following:
 - To accept all project defaults, click Finish.
 - webMethods CAF is designed to do all of the configuration necessary to create portlet applications for use with My webMethods Server.
 - Click Next and do the following (these are the choices webMethods CAF makes on your behalf):
 - On the Project Facets page, click Next.
 - 2 On the Web Module page, click Next.
 - For JSF Libraries on the JSF Capabilities page, choose the CAF Base JSF Runtime option and clear the Deploy check box.
 - This choice causes the CAF Base JSF Runtime component library to be associated with the Web application but not deployed with the application when you publish it to My webMethods Server.
 - Under Component Libraries, move the following libraries to the right box and clear the **Deploy** check box for each of them.

CAF Base Controls

CAF BPM Client (if you are developing Task applications)

CAF Portlet Controls

CAF MWS Controls

5 Click Finish.

The webMethods CAF displays a Portlet Application Configuration Editor. The result of this procedure is a standard portlet application project, which is a deployable unit packaged as a WAR (Web ARchive) file.

Related Topics

- ▶ "Associating Portlet Applications with My webMethods Server" on page 76
- "Adding a Portlet to a Portlet Application" on page 70
- "Creating a Portlet Application on the Solutions View" on page 150

Adding a Portlet to a Portlet Application

A portlet application is just a container to hold one or more portlets. To add a portlet, do the following.



To add a portlet to a portlet application

- 1 In the My webMethods, if the Portlet Application Configuration Editor is not displayed, do one of the following:
 - In the Solutions view, double-click any portlet belonging to the portlet application.
 - -OR-
 - In the Package Explorer view, Project Explorer view, or Navigator view, do the following:
 - a Locate the portlet application in the tree view and expand it to expose portlet.xml: portlet_application/WebContent/WEB-INF/portlet.xml
 - b Double-click portlet.xml.
- 2 If necessary, click the Configuration tab at the bottom of the view to make it the active tab.
- 3 Select Portlets and click Add.
- 4 From the Portlet Type list of the Project Selection window, select the type of portlet you want to create:

Portlet type	Description
Generic	A generic portlet. Suitable for creating any portlet
Search Bar	A prebuilt template for use in developing a search bar portlet.
Search Results	A prebuilt template for use in developing a search results portlet.
Task (various)	A prebuilt template for use in developing task portlets.

5 If you selected the Search Results portlet type, from the Display Style list, choose one of the following:

Display Style	Description
Table View	Search results are displayed in a table.
Tree View	Search results are displayed in a tree structure.
Table/Tree View Toggle	The user can change search results to either type of display.

- 6 Click Next.
- 7 In the Portlet Name field of the Create Portlet window, type a portlet name.

The portlet name cannot have spaces or any of the following characters:

8 In Portlet Modes, select the modes you want the portlet is to support. Select or clear the check box for these modes:

Portlet Mode	Description
View	One or more pages the user can view, navigate, or interact with. View mode is required for each portlet. The default name is default.view. This is a required view file.
Edit	One or more pages the user can use to customize the behavior of the portlet. The default name is edit.view.
Help	Help information about the portlet. The default name is help.view.

For each mode you select, Designer creates a view file that you can edit.

9 In the Title field, type a title that identifies the portlet when you publish it on My webMethods Server.

If you do not provide a title, My webMethods Server creates a title by appending the portlet name to the portlet application name.

10 To accept the defaults, click Finish.

We recommend that you accept the defaults for creating portlets. After you are skilled with portlet application development, you may find it appropriate to click Next and use the wizard to make modifications.

The design canvas, a JSF graphical editor, is displayed.

Related Topics

▶ "Creating a Portlet on the Solutions View" on page 150

Adding a View File to a Portlet

Although a portlet can contain one or more view files when it is created, you may want to add additional view files for use in a wizard or some other purpose. To add a view file, follow these steps.



To add a view file to a portlet

- 1 In Designer: File > New > Other.
- On the Select a Wizard page, expand webMethods > Composite Applications > Portlet View, and click Next.
- 3 On the JSF Application View File page, choose the portlet application project the view file is to be a part of.
- 4 In the File Name field, type the name of the view file, which must have an extension of .view, and click Next.
- From the For Portlet list on the Portlet View Options page, choose the portlet the view file is to be a part of.
- 6 In For Portlet Mode, choose the mode for this view:

Portlet mode	Description
View	A page the user can view, navigate, or interact with. View mode is required for each portlet.
Edit	A page the user can use to customize the behavior of the portlet.
Help	A page for help information about the portlet.

7 From the Initial View Content list, select the page template for the view file:

Initial view content	Description
Empty	A generic portlet. Suitable for creating any portlet
Search Bar	A prebuilt template for use in developing a search bar portlet.

Initial view content	Description
Search Results	A prebuilt template for use in developing a search results portlet.
Task (various)	A prebuilt template for use in developing task portlets.

- 8 Click Next.
- 9 From the Managed Bean Scope list on the Managed Bean panel, choose the scope for the managed bean. The default is session.
 - For descriptions of JSF scopes, see "JSF Scopes" on page 83.
- 10 Click Finish.

The design canvas, a JSF graphical editor, is displayed.

Adding a Control to a View File

In the webMethods CAF, you use the design canvas to place the elements that make up the page. The page elements are JSF controls that you drop from the Palette view.

When you drop a control onto the design canvas, the canvas ("Using the Design Canvas" on page 97) provides clues to valid drop points. Some controls have facets into which you can drop other controls and some controls require child controls to function properly.



To add controls to a portlet

- 1 If you do not have the view file open in the design canvas, do one of the following:
 - In the Solutions view, double-click the view file.
 - − OR −
 - In the Package Explorer view, Project Explorer view, or Navigator view, do the following:
 - c Expand the portlet application, expand the WebContent node, and expand the folder corresponding to the portlet you are developing.
 - **d** Within the portlet, double-click the view file to be developed.
- 2 Click the Palette view ("Using the Palette View" on page 138) to bring it to the front.
- 3 On the Palette view, expand the nodes to display the controls they contain. For example, click the Output node to display the Text control.

- 4 Drag a control until it highlights the Form command in the view file.
 - For example, drag the Text control and drop it into the target parent control in the view file.
- With the new control selected in the view file, in the Properties view ("Using the Properties View" on page 145) make changes to the properties of the control.

For example, with the Text control in the form selected, locate the Value property in the Properties view and type this text:

Hello World

When you click away from the Value property, the text appears inside the Text control in the form.

To test the result, publish the portlet to My webMethods Server and place the portlet on a page.

For information on editing pages in My webMethods Server, see the *My webMethods* Server Administrator's Guide.

Related Topics

- "Using the Design Canvas" on page 97
- "Using the Palette View" on page 138
- ▶ "Using the Properties View" on page 145

Connecting to My webMethods Server

In addition to using the MWS Admin view to publish portlet applications ("Using the MWS Admin View" on page 125) you can use the Servers view. In the Servers view, you can connect to one or more instances of My webMethods Server and publish portlet applications to them. Before you can connect to My webMethods Server, that instance of the server must be running.



To connect to My webMethods Server

- 1 Right-click inside the Servers view, and click New > Server.
- 2 In the New Server wizard, from the Server's host name field, type the name or IP number of the machine on which My webMethods Server exists.
- 3 Under Select the server type, expand webMethods and select My webMethods Server.
- 4 From the Server runtime list, select My webMethods Server and click Next.

5 Verify that the following fields are correct and make changes as needed:

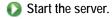
Field	Purpose	
Port	The port on which My webMethods Server is listening. The default value is 8585.	
Server	The name of the My webMethods Server instance. The default value is default.	
Debug Port	The port used by a Java debug listener. The default value is 5000.	
Publish Timeout (seconds)	The length of time My webMethods Server allows for a portlet application to be published to the server before timing out.	

6 Click Finish.



Tip! If you have existing portlet applications (projects) in your Eclipse workspace and want to configure them on the server, click Next and use the Add and Remove Projects page to add them to the Configured projects panel.

7 To begin communication with a running instance of My webMethods Server, in the Servers view, select the server instance and, in the toolbar of the Servers view, click





Note: My webMethods Server does not need to run while you create portlet applications, but does need to run when you publish them to the server.

To stop communication with My webMethods Server, in the Servers view, select My webMethods Server and, in the toolbar of that view, click Stop the server. This action does not stop the server, itself.

Related Topics

"Associating Portlet Applications with My webMethods Server" on page 76

Associating Portlet Applications with My webMethods Server

During portlet application development, you can test the portlet application using the Preview server ("Live Preview" on page 104) but it may also be helpful to publish the project to My webMethods Server. One way to do this is through the Servers view. Before you can publish a portlet application, you must associate it with (add it to) the server by doing the following:



To add a portlet application to My webMethods Server

- 1 In the Servers view, right-click an instance of My webMethods Server and then click Add and Remove Projects.
- 2 To add the portlet application to the server, move it to the Configured projects box and click Finish.

If My webMethods Server is running, Designer publishes the portlet application.

When you publish to the server, and have multiple portlet applications in the Configured projects box, all portlet applications are published. To improve performance, you can add and remove portlet applications as needed.

Related Topics

"Publishing a Portlet Application to the Server" on page 77

Running Servers in Debug Mode

If you want to debug Java code, you can run My webMethods Server in debug mode. To do so, you must stop the server and restart it in debug mode. For information on starting My webMethods Server in debug mode, see the *My webMethods Server Administrator's Guide*.



To start My webMethods Server in debug mode

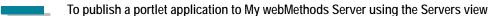
- 1 On the Servers view, select My webMethods Server and, in the toolbar of that view, click Stop the server.
- 2 On the toolbar of the Servers view, click part the server in debug mode.

Publishing a Portlet Application to the Server

When you publish a portlet application to an instance of My webMethods Server, you make all portlets in the portlet application available for use on the server. When you make changes to a portlet in the webMethods CAF, you need to republish the portlet application to the server.

One easy way to publish a portlet to My webMethods Server is to use the MWS Admin view, described in "Using the MWS Admin View" on page 125. Using this method, you can publish a portlet directly to a folder or a page.

If you want to do iterative development to a complex portlet, it may be easier to use the Server view to regularly publish new versions of the portlet to the server. Before you can use the Server view to publish a portlet application, you must associate it with the server, described in "Associating Portlet Applications with My webMethods Server" on page 76.



- 2 On the toolbar of the Servers view, click 📫 Publish to the server.

The portlet application is now available for use on the server. For information on adding portlets to server pages, see the *My webMethods Server Administrator's Guide*. If you already have the portlet displayed on a server page, you may need to refresh the page to see changes to the portlet.

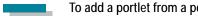
Related Topics

- "Associating Portlet Applications with My webMethods Server" on page 76
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Using the MWS Admin View" on page 125

Adding a Portlet to a My webMethods Server Page

An easy way to add a portlet to a folder or page on My webMethods Server is to use the MWS Admin view. Using that method, you can drag the portlet directly to the location where you want it to reside, as described in "Using the MWS Admin View" on page 125.

If you have more complex placement needs, after you have published a portlet application to My webMethods Server, you can log into My webMethods Server as the Designer user and add any portlets within it to a page. For more information about editing portal pages, refer to the *My webMethods Server Administrator's Guide*.



To add a portlet from a portlet application to a page on My webMethods Server

- In a Web browser and logged into My webMethods Server as Designer, move to the page.
- In the page title bar, click \(\mathbb{D} \) Tools > Edit Page.
- In the Root list of the Available Portlets panel, click Portlets.
- In the Portlets list of the Available Portlets panel, click the category that contains the portlet you want to add to the page.
- In the appropriate category list of the Page Editor, drag the portlet and drop it onto the page in the column where you want it to reside.
 - A red box appears beneath the cursor location whenever the cursor is over a page column, indicating where the portlet would be positioned if you released the mouse button.
- On the left side of the page control area, click Save.
 - The portlet appears on your portal page.

Uninstalling Portlet Applications from My webMethods Server

As you are developing portlet applications you may find that you have published portlets or portlet applications to My webMethods Server that you want to uninstall. If you have system administrator privileges on My webMethods Server, you can uninstall a portlet or portlet application by doing the following:



To uninstall a portlet or portlet application using the Install Administration page

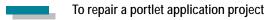
- Log on to My webMethods Server as system administrator.
- As system administrator: Administration > Configuration > Install Administration.
- Select the Uninstall option and click Next.
- Navigate to the portlet or portlet application you want to uninstall, click it, and click Uninstall.

Troubleshooting Portlet Applications

In the Package Explorer or Project Explorer view, if you find the \blacksquare overlay, here are two common causes:

- If you used a different import wizard to import the composite application projects into the workspace, the project may need to be repaired to match the environment of the workspace the projects were imported into. For example, you might import projects into the workspace using General > Existing Projects into Workspace or CVS > Existing Projects from CVS instead of webMethods > Existing CAF Projects into Workspace. The webMethods import wizard automatically repairs the project while the other import wizards do not.
- If the My webMethods Server runtime is manually removed and added back with a different name, the existing projects in the workspace may need to be repaired to use the new one.

In general if there are build errors for the project or validation errors in the .view files in the project, running the Repair CAF Project action, described here, may help



■ In the Solutions view, if you find the overlay on the node for a portlet application, right-click the portlet application and click CAF Tools > Repair CAF Project.

Chapter 5. webMethods CAF Concepts

Portlet Applications
Ajax
JSF Scopes

Portlet Applications

A *portlet application* is a container for one or more portlets. A portlet application is published to a server as a WAR (Web ARchive) file. A WAR file makes up a Java-based Web application, including Java servlets, JSPs, and other resources.

Related Topics

- "JSR 168" on page 82
- "JavaServer Faces" on page 82

JSR 168

A portlet is a server-side, mini-application that resides on My webMethods Server or a piece of functionality that runs on the back end of My webMethods Server. Portlets created in the webMethods CAF are compliant with Java Specification Request (JSR)168. This specification enables inter-operability among portlets and portals, defining a set of APIs for portlets and addressing standardization for preferences, user information, portlet requests and responses, deployment packaging, and security. Because JSR 168 has become a standard for portal development you can find numerous third-party publications that describe how such portlets function.

JSR 168 portlets are stateless. To bring state to portlets, JSR168 uses preferences, which are either strings or string arrays that are persisted per portlet type, for each user. Software AG adds extensions to preferences, adding simple data types such as booleans and integers, and adding additional scopes. The extended scopes are valid only on My webMethods Server. The behavior of portlets exported to third-party portals defaults to that of the JSR 168 specification.

JavaServer Faces

Portlets created in the webMethods CAF use JavaServer Faces (JSF) technology. JSF is an application framework for creating Web-based user interfaces. JSF provides Web application lifecycle management through a controller servlet and a rich component model with event handling and component rendering. JSF UI components are configurable, reusable elements that make up the user interfaces of JSF applications. A button is an example of a simple component; a table is a compound component, which can be made up of multiple components.

webMethods CAF portlets are made up of view files. A *view* is a JSF page made up of UI components.

For each portlet in a portlet application, there is a managed bean. A *managed bean* is a Java bean, which is a Java class with a set of properties. A managed bean includes metadata that defines its unique name within a portlet application, a bean type (the class type), and a scope. The scope determines how long the bean lives within the application.

Ajax

Ajax (Asynchronous JavaScript and XML), is a technique for creating interactive Web applications. In the old model for Web applications, the entire page has to be reloaded each time the user makes a change. In the Ajax model, an Ajax engine acts as an intermediary between the user and the server. The user interaction with the application is said to happen asynchronously, independent of communication with the server. In this way, the application appears to respond immediately to user input.

Many of the controls available on the Palette view of the webMethods CAF use Ajax principles, making it possible for you to incorporate Ajax into your portlet applications. Controls with Ajax capabilities are identified as such in the Controls Reference.

Related Topics

• "User Interface Controls Reference" on page 179

JSF Scopes

A *managed bean* is a Java bean, which is a Java class with a set of properties. A managed bean includes metadata that defines its unique name within a portlet application, a bean type (the class type), and a scope. The scope determines how long the bean lives within the application.

The following scopes are available to govern the lifecycles of JSF managed beans:

Scope	Description
Application	Shared by all objects in a portlet application. The managed bean is started once and does not expire until the server shuts down or the portlet application is republished to the server.
Session	Scoped per user session within a portlet. Data cannot be shared across portlets. The managed bean expires when the user session ends.
Request	Scoped per a single request. The managed bean expires before the next request.
None	The bean is created every time it is requested.

Chapter 6. Views in webMethods CAF

My webMethods Perspective Overview
Using the Portlet Application Configuration Editor
Using the Design Canvas
Using the Bindings View
Using the Data Source Explorer View
Using the IS Browser View
Using the MWS Admin View
Using the Outline View
Using the Palette View
Using the Properties View
Using the Solutions View
Specifying an External Web Browser

My webMethods Perspective Overview

A perspective defines the initial set and layout of views in the Eclipse window, and provides a set of features tailored to a specific type of task. The webMethods CAF allows you to develop portlet applications for use on My webMethods Server or on any JSR-168-compliant portal.

You compose portlets on a design canvas, which is a JSF graphical editor. In addition to the design canvas, there are several views that are used in the webMethods CAF. Some of these views are part of the Eclipse IDE while others are specific to the webMethods CAF. The following list provides brief descriptions of these views.

View	Description
Bindings	A tree view that contains all data that can be used within this view or within the entire portlet or Web application. See "Using the Bindings View" on page 106.
Data Source Explorer (Eclipse)	A tree view of data sources. Using this view you can connect to, navigate, and interact with resources associated with a selected connection profile, such as a database. Visible by default in the webMethods UI Developer. See "Using the Data Source Explorer View" on page 113
IS Browser	(My webMethods perspective only) A tree view displaying the resources on instances of Integration Server to which Designer is connected. You can use this view to select services and create Web service connectors that can be used to interact with Integration Server. See "Using the IS Browser View" on page 115.
MWS Admin	(My webMethods perspective only) A tree view displaying the resources on instances of My webMethods Server to which Designer is connected. You can use this view to publish portlets to the servers and reorganize server resources. See "Using the MWS Admin View" on page 125.
Navigator (Eclipse)	A tree view of the resources in the workspace. From here, you can open files for editing or select resources for operations such as exporting. Visible by default in the My webMethods perspective. The Navigator view is described as needed in various topics.
Outline (Eclipse)	An outline of a view that is currently open in the design canvas. This view is useful in locating specific controls within a complex portlet or Web application and for moving controls from one location to another. See "Using the Outline View" on page 137.
Palette	A tree view of controls you can use to build portlet or Web applications. You can drag controls from this view directly to the design canvas. See "Using the Palette View" on page 138.

View	Description
Problems (Eclipse)	A display of system-generated errors, warnings, or information associated with a resource. For example, if you save a Java source file that contains syntax errors, the errors are automatically are logged in this view.
Project Explorer (Eclipse)	A tree view of the resources in the workspace. From here, you can open files for editing or select resources for operations such as exporting. Visible by default in the webMethods UI Developer perspective. The Project Explorer view is described as needed in various topics.
Properties (Eclipse)	A view of display names and values of properties for a resource selected elsewhere in Designer. This view is useful in changing the values of properties of controls in the design canvas and objects in the MWS Admin view. See "Using the Properties View" on page 145.
Servers (Eclipse)	A table of runtime environments associated with Designer. Once configured, you can publish portlet or Web applications to the server.
Solutions	A tree view that represents a logical representation of the resources in the workspace which is simpler than the one provided in the Navigator or Project Explorer. You can perform selected operations on resources shown in this view. See "Using the Solutions View" on page 149.

In addition to the views, there are various editors:

Editor	Description
Design canvas	A JSF graphical editor in which you build portlet or Web applications by adding controls from the Palette view. This editor provides a logical view; to see how the portlet or Web application appears on a page, you can preview it using Instant Preview or the Preview Server. See "Using the Design Canvas" on page 97.
Portlet Application Configuration Editor	(My webMethods perspective only) An editor that allows you to create and manage portlets and other resources belonging to a portlet application. This editor is described in "Using the Portlet Application Configuration Editor" on page 88.
Java editor	A source code editor that provides specialized features for editing Java code.

Related Topics

- "Using the Portlet Application Configuration Editor" on page 88
- "Using the Design Canvas" on page 97
- ▶ "Using the Palette View" on page 138
- ▶ "Using the Properties View" on page 145
- ▶ "Using the Outline View" on page 137
- ▶ "Using the Solutions View" on page 149
- "Using the Bindings View" on page 106
- "Using the IS Browser View" on page 115
- "Using the MWS Admin View" on page 125

Using the Portlet Application Configuration Editor

The Portlet Application Configuration editor manages the portlet descriptors for all the portlets contained in a portlet application project in a <u>ISR-168</u> compliant manner. You can create portlets and various objects used by portlets and the portlet application. When you create a portlet application, the Portlet Application Configuration editor is automatically displayed. If you later closed that editor, you can open again by doing the following:



To open the Portlet Application Configuration editor for a portlet application

- 1 Do one of the following:
 - On the Solutions view, double-click any portlet belonging to the portlet application.
 - OR -
 - On the Package Explorer view, Project Explorer view, or Navigator view, do the following:
 - a Locate the portlet application in the tree view and expand it to expose portlet.xml: portlet_application/WebContent/WEB-INF/portlet.xml
 - b Double-click portlet.xml.

Related Topics

- ▶ "Adding a Portlet to a Portlet Application" on page 70
- "Creating Portlet Preferences" on page 89

- ▶ "Using Security Roles" on page 94
- "User Attributes" on page 95

Creating Portlet Preferences

Portlet *preferences* are a means of storing values that can be used within a portlet. Once a preference is created, you can provide access to it through the use of binding expressions. Preferences can be persisted through the use of scopes. In JSR 168, there is only one scope for preferences, value stored per user, but Software AG has added some extensions to provide greater flexibility.



To create a portlet preference

- 1 Do one of the following:
 - On the Bindings view. right-click a portlet and click Add > Portlet Preferences.
 - In the Portlet Application Configuration panel of the Portlet Application Configuration editor, expand the Portlets node, expand the portlet, click Preferences, and then click Add.
- 2 In the Name field of the Create Portlet Preference wizard, type a name for the preference.
- 3 (Optional) On the Value box, click **Add** and then type a default value for the preference.
- 4 If the preference is not to be modified, click the Is Read-Only option.
- 5 Click Next.
- 6 On the Type list, choose a data type for the preference.
 The default is String.
- 7 On the **Scope** list, choose the scope for the preference:

Scope	Discussion
Value stored per portlet instance	Different instances of the portlet on different portal pages have their own values. All users who access the portlet instance see the same value. Used in the majority of portlet cases, this scope is useful for wiring and linking of portlets. The default.
Value stored per user (JSR 168 default)	Each individual user has a separate preference value for each portlet instance.

Scope	Discussion
Value shared by all portlet instances	All instances of the portlet share the same value. Used this way, the preference becomes a static variable. This scope is useful when a preference is a configuration parameter that does not vary among portlet instances
Transient session value	The value is stored for an individual user until that user logs out of the portal session or until the portlet bean scope expires.

- 8 If you want the preference to be visible on a Properties page for a portlet on My webMethods Server, clear the Hidden option.
- Click Finish.

Related Topics

- "Using Preferences to Perform Wiring" on page 90
- ▶ "Storing a Preference" on page 92
- ▶ "Using the Bindings View" on page 106

Using Preferences to Perform Wiring

Wiring is the pulling of data from other portlets. The following example creates a portlet that contains a postal code stored in a preference and then pulls that preference into another portlet.



To use a preference to perform wiring from one portlet to another

- 1 Do one of the following:
 - On the Bindings view. right-click a portlet and click Add > Portlet Preference.
 - In the Portlet Application Configuration panel of the Portlet Application Configuration editor, expand the Portlets node, expand the portlet, click Preferences, and then click Add.
- 2 In the Name field of the Create Portlet Preference wizard, type a name for the preference, such as Postal Code.
- 3 On the Value box, click **Add** and then type a valid postal code.
- 4 Click Finish.

5 On the Bindings view, locate the Postal Code preference and drag it to the design canvas.

Two controls are automatically placed on the design canvas:

Control	Purpose
Control Label	Provides a display label for the portlet.
Text Input	Provides an input field to contain the postal code.

6 Click each control in the design canvas and check the Value property for that control in the Value tab:

Control	Value property
Control Label	The value you typed in the Name field when you created the preference, in this case, Postal Code.
Text Input	A binding expression to the preference.

- 7 Publish the portlet application to My webMethods Server.
- 8 On My webMethods Server, add the portlet to a server page.
 - When you view the page, the portlet should display the label and the postal code created in the preference.
- 9 Edit the portal page to add the sample Yahoo! Weather portlet. In the Available Portlets panel, click Portlets, Internet- News/Weather, and Yahoo! Weather.
- 10 Wire the Zip Codes property in the Yahoo! Weather portlet to the Postal Code property in new portlet by doing the following:
 - On the Yahoo! Weather portlet, click Tools > Wiring.
 - b In the Portlet list for Zip Codes, select the postal code portlet, and then click Browse.
 - c In the Choose the Wired Property window, click Postal Code (or whatever display name you chose in an earlier step) and click Select and Submit.

The Yahoo! Weather portlet should now display current weather conditions for the postal code in the initial preference.

If you type a different postal code in the editable field, the Yahoo! Weather portlet displays information for the new postal code.

Related Topics

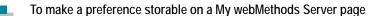
- "Creating Portlet Preferences" on page 89
- ▶ "Publishing a Portlet Application to the Server" on page 77
- "Adding a Portlet to a My webMethods Server Page" on page 77

Storing a Preference

Every portlet managed bean automatically has a Store Preferences method. On My webMethods Server, the method fetches the current state of the preference and then persists it. After the preference has been persisted, you can go to the Properties page for the portlet and find it there. In turn, you could also modify the preference on the Properties page and then see the change on the server page.



Note: For a preference to be visible on the Properties page, you must clear the Hidden option when you create the preference. See "Creating Portlet Preferences" on page 89.



- In the Portlet Application Configuration panel of the Portlet Application Configuration editor, expand the Portlets node, expand the portlet, click Preferences, and then click Add.
- 2 In the Name field of the Create Portlet Preference wizard, type a name for the preference and click Finish.
 - Unlike the example in "Using Preferences to Perform Wiring" on page 90, you do not add an initial value for the preference.
- 3 On the Bindings view, locate the new preference and drag it to the design canvas.

Two controls are automatically placed on the design canvas:

Control	Purpose
Control Label	Provides a display label for the portlet.
Text Input	Provides an input field to contain the postal code.

4 Click each control in the design canvas and check the Value property for that control in the Value tab:

Control	Value property	
Control Label	The value you typed in the Name field when you created the preference.	
Text Input	A binding expression to the preference.	

- On the Bindings view, locate the Store Preference method for the portlet managed bean and drag it to the design canvas beside the two controls previously added.
 - A Command Button control is automatically added to the design canvas. If you look at the Action property for the button on the Properties view, you can see a binding expression to the Store Preferences method.
- 6 (Optional) To change the button text, click the Command Button control on the design canvas and then modify the Value property on the Properties view. The default is Store Preferences.
- 7 Display the page using the Preview Server, as described in "Live Preview" on page 104.
 - When you view the page, the text input field should be empty.
- 8 Type text into the input field and then navigate away from the page without clicking the Store Preferences button.
 - The text is lost.
- 9 Type text into the input field and click the Store Preferences button.
 - The text is now persisted on the page until you, or another user, types new input and clicks Store Preferences again.

Related Topics

- "Creating Portlet Preferences" on page 89
- ▶ "Publishing a Portlet Application to the Server" on page 77
- ▶ "Adding a Portlet to a My webMethods Server Page" on page 77

Using Security Roles

The use of security roles is a J2EE concept that allows you to hide controls on a page unless the viewer belongs to a certain role. For information on assigning members to a role on My webMethods Server, see the My webMethods Server Administrator's Guide.

Related Topics

- ▶ "Creating a Security Role" on page 94
- ▶ "Binding a Security Role to a Control" on page 94
- "Using an Access Control Panel Control" on page 95

Creating a Security Role

You create security roles for a portlet. When the portlet is published to an instance of My webMethods Server, the security role becomes a static role on the server. You can assign users, groups, or other roles to be members of a static role on My webMethods Server, as described in the My webMethods Server Administrator's Guide.



To create a security role

- In the Portlet Application Configuration panel of the Portlet Application Configuration editor, expand the Portlets node, expand the portlet, click Security Roles, and then click Add.
- 2 In the Role Name field, type the name of the role.

The name you choose appears among roles after you publish the portlet to My webMethods Server. By default, the sole member of the role is the My webMethods Server system administrator (SysAdmin).

3 Click Finish.

The security role should appear on the Bindings view under the portlet in which you created it. You may need to click Refresh to update the view.

Binding a Security Role to a Control

To bind an existing security role to a control, do the following:



To bind a security role to a control

- 1 On the design canvas, place a control on the form and make sure it is selected.
- 2 On the Properties view, locate the Rendered property on the General tab. By default, the Rendered property for a control is set to true.

- 3 Create a binding expression to the security role by doing the following.
 - a Click #{...} at the right of the Rendered property.
 - b In the Expression Binding wizard, expand the portlet node, click the security role to be assigned, and click **OK**.

The security role is now a binding expression for the control.

The resulting binding expression will return true if the user is a member of the privileged role.

Using an Access Control Panel Control

In addition to binding a control to a security role, you can use an Access Control Panel control to manage security roles for one or more controls. Using this control provides more flexibility because you can both allow and deny security roles.



To manage security roles using an Access Control Panel control

- 1 Locate the Access Control Panel control in the Panel node of the Palette view, drag it to a location on the design canvas, and make sure it is selected.
- 2 In the Display tab of the Properties view, create binding expressions to security roles for either of both of the Allowed Security Role and Denied Security Role properties by doing the following:
 - a Click #{...} at the right of the property.
 - b In the Expression Binding wizard, expand the portlet node, click the security role to be assigned, and click **OK**.

User Attributes

Portlets commonly provide content personalized to the user making the request. To do so, they may require access to user attributes such as the name, E-mail, or telephone number of the user. Each portlet application has a default set of user attributes to which you can add. Once specified for a portlet application, user attributes are available for use in binding expressions.



Note: If you do not bind a user attribute related to localization to a portlet application, the behavior in My webMethods Server is that a **Default** value in a user's profile is a null value.

There are three aspects to using user attributes in portlet applications:

- ▶ "Adding a User to a Portlet Application" on page 96
- ▶ "Binding a User Attribute to Output Text" on page 96
- ▶ "Configuring User Attributes at Run Time" on page 97

Adding a User to a Portlet Application



To add a user attribute to a portlet application

- 1 In the Portlet Application Configuration panel of the Portlet Application Configuration editor, click User Attributes, and then click Add.
- 2 In the Attribute Name list of the Declare User Attribute wizard, choose the user attribute to be added and click Finish.

The user attribute is now available for use in binding expressions.

Binding a User Attribute to Output Text



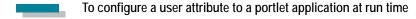
To use a user attribute in a binding expression

- 1 Locate the Text control in the Output node of the Palette view, drag it to a location on the design canvas, and make sure it is selected.
- 2 On the Value tab of the Properties editor, click #{...} at the right of the Value property.
- 3 In the Expression Binding wizard, expand the JSF Managed Beans node, click the user attribute to be assigned, and click **OK**.

For example, if you choose the user.name.given attribute, the published portlet would display the given name of the user who views the page.

You can also bind a user attribute to a control directly from the Bindings view. Expand nodes in that view until you expose the user attributes and drag a user attribute directly to a control container in the design canvas.

Configuring User Attributes at Run Time



- 1 Publish the portlet application to My webMethods Server.
 - Two different methods of publishing to My webMethods Server are:
 - "Publishing a Portlet Application to the Server" on page 77
 - "Deploying Portlets to My webMethods Server" on page 129
- 2 Log on to My webMethods Server as a system administrator (sysadmin).
- 3 On My webMethods Server: Administration Dashboard > Configuration > CAF Application Runtime Configuration.
- 4 On the CAF Application Runtime Configuration page, type the name of the portlet application in the **Keyword** field and click **Go**.
- 5 In the search results, click the name of the portlet application.
- 6 On the CONFIGURE APPLICATION: portlet_application panel, click User Attributes.
- 7 For each name in the Logical Attribute Name column, select the value from the Portal Attribute list.
- 8 Click Apply.

Using the Design Canvas

The design canvas (also called the view editor) is a graphical editor used to lay out all of the controls for a portlet view file. The design canvas does not provide an exact representation of the layout as seen on a My webMethods Server page. To see how the view file actually looks on the page, you can use either of these features:

- The Instant Preview, described in "The Instant Preview" on page 103
- The Preview Server, described in "Live Preview" on page 104

You create a page layout by dropping UI controls into containers in the design canvas. When you first create a portlet, these controls are automatically placed on the canvas:

Control	Purpose	Reference
Form	A JSF container for input controls.	"Form" on page 268
Formatted Messages	Displays messages (such as validation messages) to the user.	"Formatted Messages" on page 366

Some controls can contain other controls in a parent-child relationship. You can drag controls onto the canvas using the following guidelines:

- A vertical red bar appears at the place where the control will be dropped.
- If the **(**) overlay is visible on the cursor, you cannot drop the control.
- If the ♠ overlay is visible on the cursor, you can drop the control.
- You cannot drop an input control ("Input Controls" on page 289) outside of a Form control.
- As you drag the control, descriptive hints are displayed to help you determine where to drop it.

The design canvas is fully integrated with the Java building mechanism in Designer. As you make changes on the canvas, the underlying Java code in the managed beans is updated automatically.

Delete, Cut, Copy, Paste, and Undo commands from the File menu are valid on the design canvas. These commands are also available when you right-click individual controls on either the design canvas or the Outline view.

The following procedure shows a greatly simplified version of how you might use the design canvas along with other views in the webMethods CAF perspective to create a portlet page, also called a view file.



To use the design canvas and other views to create a portlet page

- 1 If you do not have the view file open in the design canvas, on the Solutions view ("Using the Solutions View" on page 149), expand the portlet application, expand the portlet, and double-click the view file.
- 2 On the Palette view ("Using the Palette View" on page 138), drag controls to the design canvas.
 - Some controls require the use of other controls.
- 3 On the Properties view ("Using the Properties View" on page 145), modify properties associated with the control.
- 4 As the hierarchy of controls on the design canvas becomes more complex, use the Outline view ("Using the Outline View" on page 137) to navigate among controls, drag controls from one location to another, and perform delete, cut, copy, paste, and undo operations.
- 5 Display the page using the Preview Server, as described in "Live Preview" on page 104.

Related Topics

- "The Design Canvas Toolbar" on page 99
- "Manipulating Controls on the Design Canvas" on page 100
- ▶ "The Instant Preview" on page 103
- ▶ "Live Preview" on page 104
- ▶ "Debugging from the Design Canvas" on page 106

Additional Topics

- ▶ "Using the Palette View" on page 138
- "Using the Properties View" on page 145
- "Using the Outline View" on page 137
- "Using the Solutions View" on page 149
- "User Interface Controls Reference" on page 179

The Design Canvas Toolbar

The toolbar at the top of the design canvas provides mechanisms for navigating in the design canvas and displaying both source and preview content.

Button	Name	Purpose
	Home	Sets the top node of the design canvas to be the View Root control. Enabled only after the Go Into action has set another another control as the top node.
\$	Back	Sets the top node of the editor to be the parent node of the current node. Enabled only after the Go Into action has set a control as the top node.
4	Go Into	Sets the top node of the editor to be the currently selected control. This action enables you to drill into a specific section of the canvas for reduced clutter.
Sy	Refresh	Redraws the visuals for the design canvas.
~	Validate	Executes the validation for the .view resource. Any validation errors discovered are available in the Problems view.
0	Run	Builds and publishes the application to the Preview Server and opens a browser to display the .view you are editing. See "Live Preview" on page 104.

Button	Name	Purpose
参	Debug	Builds and publishes the application to the Preview Server and opens a browser to display the .view you are editing.
	Match Width	Sets the width of the selected controls to the width of the last control selected. Enabled only when more than one control is selected.
	Match Height	Sets the height of the selected controls to the height of the last control selected. Enabled only when more than one control is selected.
	Source	Opens the page bean Java class associated with the .view. Visible only when the pageManagedBean property of the View Root control is a valid managed bean name.

Related Topics

- "Manipulating Controls on the Design Canvas" on page 100
- ▶ "The Instant Preview" on page 103
- ▶ "Live Preview" on page 104
- "Debugging from the Design Canvas" on page 106

Manipulating Controls on the Design Canvas

There are several ways to manipulate controls on the design canvas and modify their appearance and position.

- ▶ "Resizing Controls" on page 101
- "Moving Controls" on page 101
- "Minimizing Controls" on page 101
- "Matching Controls" on page 101
- ▶ "Editing CSS Values" on page 102
- ▶ "Changing Control Type" on page 102
- "Changing Labels Directly on the Design Canvas" on page 103
- "Selecting a Parent or Child Control" on page 103
- ▶ "Showing a View" on page 103

Resizing Controls

To resize a control or a column, click the control to display the resizing handles at the corners and the center of each edge. Drag a handle to resize as needed. If you have multiple controls selected (SHIFT+click), you can resize them at the same time.

Moving Controls

To move a control from one place to another, click the control to display the drag handle () in the upper left corner. Place the cursor over the drag handle and drag the control to a new location using these guidelines:

- A vertical red bar appears at the place where the control will be dropped.
- If the **(**O overlay is visible on the cursor, you cannot drop the control.
- If the \bigoplus overlay is visible on the cursor, you can drop the control.
- You cannot drop an input control ("Input Controls" on page 289) outside of a Form control.
- As you drag the control, descriptive hints are displayed to help you determine where to drop it.

Minimizing Controls

As the hierarchy of controls becomes more complex, you can minimize a control to hide all of its child controls. Click the control to display the collapse icon (\bigcirc) in the upper left corner, and then click the icon.

To restore the control, click the control to display the restore icon (\odot) in the upper right corner of the collapsed control and then click the icon. You can also restore the collapsed control by double-clicking the control.

Matching Controls

You can set a group of controls to have the same width or height:



To set a group of controls to have the same width or height

- 1 Resize one control to the desired width or height.
- 2 SHIFT+click each of the controls to have the matching width or height, clicking the correctly sized control last.

The last control selected becomes the primary selection.

- 3 Do one of the following:
 - To match control widths, on the design canvas toolbar click Match Width of Selected Objects to the Primary Selection.
 - To match control heights, on the design canvas toolbar click Match Height of Selected Objects to the Primary Selection.
- 4 Click the Preview tab at the bottom of the design canvas for a static view of the control layout.

Editing CSS Values

You can edit CSS (Cascading Style Sheet) values for controls without having to know CSS syntax. On the design canvas, right-click a control and then click **Style**. The resulting menu offers these choices:

Menu item	Description
Edit Style	Displays the CSS Style Definition editor. Use this editor to set values for a variety of CSS settings.
Reset Style	Click to reset the control to default CSS settings.
Style Classes	Choose a preset CSS style class. This feature requires that you have created an external CSS and used an Include Stylesheet control to associate it with the view. See "Include Stylesheet" on page 378.
Color	From the menu, choose a color to be used for foreground objects. The choices are a subset of colors available in the CSS Style Definition editor.
Background Color	From the menu, choose a color to be used for the background. The choices are a subset of colors available in the CSS Style Definition editor.

Changing Control Type

If you realize that you need a different control from the one you placed on the canvas, rather than delete the control and drag another one from the Palette view, you can change the control type directly from the design canvas.



To change the control type on the design canvas

- 1 Right-click the control to changed and click Change Control Type.
 - The Control Type Editor is displayed, with a tree view controls. The node representing the current control type is selected.
- 2 Click the new control type and then click **OK**.
 - If you click a control type that is not compatible with the current control, the **OK** button is disabled.

Changing Labels Directly on the Design Canvas

Rather than use the Properties view, you can change the label for a control directly on the design canvas.



To change a control label directly on the design canvas

- 1 Click the control to select it, and then pause for a second or two.
- 2 Click the control a second time to open an editing field.
- 3 Type the new label and press CTRL+ENTER or click away from the control.

Selecting a Parent or Child Control

With a particular control, you can quickly select its immediate parent control or any of its immediate child controls. Right-click the control, click Select, and then click the parent or child control. That control is now selected in the design canvas.

Showing a View

Occasionally a view is hidden because another view has become active. You can redisplay certain views directly from the design canvas. Right-click a control, click **Show**, and click the name of the view to be displayed. You can use this technique to display the Properties, Palette, or Outline view.

Related Topics

- ▶ "The Design Canvas Toolbar" on page 99
- ▶ "The Instant Preview" on page 103
- ▶ "Live Preview" on page 104
- ▶ "Debugging from the Design Canvas" on page 106

The Instant Preview

The Instant Preview is a feature of the design canvas that shows a static representation of the .view as it would be seen in a deployed portlet. The preview is static HTML and does not include data. To see the Instant Preview, click the Preview tab at the bottom of the editor. To return to the design canvas, click the Design tab at the bottom of the editor.

Related Topics

- ▶ "The Design Canvas Toolbar" on page 99
- "Manipulating Controls on the Design Canvas" on page 100
- ▶ "Live Preview" on page 104
- ▶ "Debugging from the Design Canvas" on page 106

Live Preview

Designer is installed with a light-weight version of the My webMethods Server runtime, called the Preview Server, that is dedicated to the live preview of portlet applications. Unlike the Instant Preview feature ("The Instant Preview" on page 103), the Preview Server uses an internal Web browser to display a fully functional instance of the portlet application. The internal browser is a standard Web browser with limited navigation capability.

- "Running the Preview Server" on page 104
- "Stopping the Preview Server" on page 104
- "Recreating the Preview Server" on page 105
- ▶ "Actions You Can Perform on the Live Preview" on page 105
- "Specifying an External Web Browser" on page 155

Running the Preview Server

When you start Designer the first time after installation, the Preview Server is initialized. The server is listed in the Servers view as My webMethods Preview Server.



To display or update a portlet application in the Preview Server

- On the toolbar of the design canvas (view editor), click Run.

 If the Preview Server is not currently running, the server is started. Designer builds the portlet application, deploys it to the server, and displays the portlet application in a browser.
- 2 To update the live preview after you have made changes in the design canvas, click Run.

Stopping the Preview Server

Once you have started the Preview Server, it continues to run until you exit from Designer or stop the server manually. To determine if the Preview Server is running, check the status in the Servers view. If the status for My webMethods Preview Server is Started, the Preview Server is running.



To stop the Preview Server manually

On the Servers view, select My webMethods Preview Server and on the toolbar, click in the toolbar of that view, click Stop the server.

Recreating the Preview Server

In some cases you may find it necessary to delete and recreate the Preview Server:

- The Preview Server becomes cluttered with too many portlet applications
- You are using a new version of Designer with an older workspace



To delete and recreate the Preview Server

- 1 On the Servers view, right-click My webMethods Preview Server and click Delete.
- 2 Right-click inside the Servers view, and click New > Server.
- 3 Select My webMethods Preview Server and click Finish.

Actions You Can Perform on the Live Preview

The live preview, displays a fully functional version of the portlet application running on the Preview Server. If the portlet application has multiple pages, you can navigate to each of the pages. The table below shows navigation buttons on the toolbar internal Web browser. External Web browsers have equivalent functions.

Button	Name	Purpose
\$	Back to the previous page	Displays the previous page of a portlet application.
\Rightarrow	Forward to the next page	Displays the next page of a portlet application.
	Stop loading the current page	Stops the loading of the current page.
8	Refresh the current page	Refreshes the display of the current page.

The toolbar also has a field in which to type a URL. This field exists because the internal browser is in fact a standard Web browser, but it has no practical use as part of the internal browser.

Related Topics

- "The Design Canvas Toolbar" on page 99
- "Manipulating Controls on the Design Canvas" on page 100
- ▶ "The Instant Preview" on page 103
- ▶ "Debugging from the Design Canvas" on page 106

Debugging from the Design Canvas

As you develop complex portlet applications, you may find it necessary to debug the underlying Java code. To do so, you can run the Preview Server ("Live Preview" on page 104) in debug mode.



To run the Preview Server in debug mode

- 1 On the toolbar of the design canvas, click **Source**.
- 2 In the Java editor, locate the point at which you want to set a breakpoint.
- Right-click the shaded bar at the left edge of the Java editor and click Toggle Breakpoint.

 A breakpoint is displayed in the shaded bar.
- 4 On the design canvas toolbar, click 🐞 Debug.
 - When the Preview server reaches the breakpoint, it stops and this prompt appears Confirm Perspective Switch.
- 5 If you click Yes, you are switched to the Debug perspective.

Related Topics

- "The Design Canvas Toolbar" on page 99
- "Manipulating Controls on the Design Canvas" on page 100
- ▶ "The Instant Preview" on page 103
- ▶ "Live Preview" on page 104

Using the Bindings View

The Bindings view provides a logical representation of all the data that is available to the portlet in the design canvas. The purpose of this view is to bind data to the UI Controls so the UI can display information dynamically, and to provide data bindings for data elements. The Bindings view is active whenever the design canvas or other editor can provide binding data. In turn, when you select a node on the Bindings view, properties for that node are displayed on the Properties view.

Binding expressions are the means by which you bind data to controls. Binding expressions take the following form:

#{name_of managed_bean.property_it_is_getting}

The Bindings view contains several types of managed beans and other assets used by the portlet:

- Portlet Application Bean () − Exposes resource bundles, resources, and eventually user attributes. There is one application bean for each portlet application.
- Portlet Bean() —Exposed preferences and security roles. There is one portlet bean for each portlet in a portlet application.
- Page Bean (□) —Exposes the UI Component Model that is generated from the Controls dropped on to the page. This is the primary Bean that developers interact with when creating User Interfaces. There is one Page Bean for each page in a portlet.

By default, the Bindings view is filtered to display data only for the portlet that contains the page visible in the design canvas. Normally, you should leave this filter in place to avoid using data from a different portlet; in most cases, you need to create binding expressions to data associated with the current portlet.

The Bindings view has automatic user interface generation. If you drag an object from the Bindings view to the design canvas, the canvas is populated with controls appropriate to the object. For some objects, such as Web service connectors, there are multiple sets of controls that might be appropriate; in that case, you can choose among available input controls. Also in the case of complex objects like Web connectors, you can choose which elements of the object to be included. For a simple example of this, see "Using Preferences to Perform Wiring" on page 90.

Related Topics

- "Creating a New Method" on page 107
- ▶ "Adding a New Property to a Page Bean" on page 109
- "Customizing Initial Values for a Page" on page 110
- "Showing All Managed Beans" on page 111
- ▶ "Adding a Reference to Another Page" on page 112

Creating a New Method

If you need to create a new method to be invoked by a command, you can do the following:



To create a new method in a page managed bean

1 On the Bindings view, right-click the Page Bean () and click Add > Action.

The Page Bean is the top bean on the Bindings view and takes this form:

/portlet_name/view_name

2 In the Action Name field, type a name for the method and click Finish.

For this simple example, leave the Implement this Action as Data Flow option cleared. You would choose this option if you want to use binding expressions to assign data from a source to a destination and call other actions without any coding. Each Page bean has an Initialize() method. You can use Data Flow Implementation context menu to define data flow for initialization the page. This method is called before the page gets rendered for the first time.

- 3 On the Bindings view, double-click the newly created method to open it in the Java editor.
- In the Java editor that opens, type the method, such as the following example, where the Page Bean is named testBean. The new method appears in **bold**:

```
return testBean;
}

public void doSomething() {
    System.out.println("something");
}
```

- 5 Click File > Save.
- 6 Click the tab for the design canvas.

The method appears as an action on the Bindings view beneath the portlet bean.

- 7 To use the method on the page, drag it from the Bindings view to the design canvas.
- By default, a Command button appears on the design canvas at the point to which you dragged the method and is bound to the method.

Related Topics

- "Adding a New Property to a Page Bean" on page 109
- "Customizing Initial Values for a Page" on page 110
- ▶ "Showing All Managed Beans" on page 111
- ▶ "Adding a Reference to Another Page" on page 112

Adding a New Property to a Page Bean

You can add a new property to a page managed bean by doing the following:



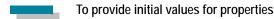
To add a property to a page managed bean

- 1 On the Bindings view, right-click the Page Bean () and click Add > Data.
- 2 In the **Property Name** field of the Add New Property wizard, type a name for the property.
- 3 In the Property Type field, type the property type or click Browse to find a type.
- 4 Select the option to make the property type a scalar (the default), a modifiable list, or a read-only array.
- 5 Select the **Property is writable** option if the property is to have a setter. If you leave the option cleared, the property will have only a getter.
- Click Finish.
 - The property appears as on the Bindings view beneath the Page Bean.
- 7 To use the property on the page, drag it from the Bindings view to the design canvas.
- 8 If prompted, select the UI controls for the data.
 UI controls are automatically placed on the design canvas at the point to which you dragged the property and are bound to the property.

- "Creating a New Method" on page 107
- "Customizing Initial Values for a Page" on page 110
- "Showing All Managed Beans" on page 111
- ▶ "Adding a Reference to Another Page" on page 112

Customizing Initial Values for a Page

You can provide initial values to be used for properties when a page is first displayed. For example, perhaps you want a "Text Input" on page 351 control to be filled with a value that may later be replaced by user input. To do so, use the Initialize method that is made visible on the Bindings view.



- 1 Assume that you have placed a control, such as a Text Input control, on the design canvas and have given it an appropriate label.
- 2 On the Bindings view, right-click the Initialize() method for the page (methods are identified by the icon) and click Data Flow Implementation.
- 3 On the Data Flow for Action wizard, click Add Assignment.
- 4 On the highlighted row, click the Target Property/Action column (left side), and then click #{...}.
- 5 On the Expression Binding wizard, select the target property, and click **OK**.
- 6 On the Data Flow for Action wizard, click the Source Property column (right side), and do one of the following:
 - Type the value that is to be displayed when the page is initialized.
 - Click #{...}, on the Expression Binding wizard select an attribute or property that will be applied to the source property, and click OK.

7 Click Finish.

You can use this procedure to add multiple target/source property pairs to the Initialize() method.

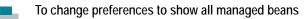
- "Creating a New Method" on page 107
- ▶ "Adding a New Property to a Page Bean" on page 109
- "Showing All Managed Beans" on page 111
- "Adding a Reference to Another Page" on page 112

Showing All Managed Beans

By default, the Bindings view displays only managed beans associated with the page bean currently displayed on the design canvas. In most cases this mode is desirable because you can drag to the canvas only those properties and managed beans that are part of the current page bean. There are, however, some reasons to display managed beans that are not part of the current page bean:

- To add references to other managed beans that are in the project but not in the current page bean
- To delete managed beans that are not referenced by the current page bean (or by any other managed bean in the project)

To enable the display of all managed beans in the project, you need to change a preference setting in Designer.



- 1 To change preferences in Designer: Window > Preferences > webMethods.
- 2 Expand the webMethods node and the Web Development node, and click the Bindings node.
- 3 On the Bindings view page, choose the Display the 'Show All Managed Beans' toolbar button checkbox and click OK.
 - You need to close and reopen the Bindings view to display a new toolbar button.
- 4 To close the Bindings view, click ⋈ Close.
- 5 To display the Bindings view: Window > Show View > Bindings.
- 6 To show all managed beans in the project, on the Bindings view toolbar, click Show All Managed Beans.
- 7 To show only the managed beans in the current page bean, click the button again.

- "Creating a New Method" on page 107
- "Adding a New Property to a Page Bean" on page 109
- "Customizing Initial Values for a Page" on page 110
- "Adding a Reference to Another Page" on page 112

Adding a Reference to Another Page

If you have the Show All Managed Beans preference set in Designer, you can add a reference from one page in a project to another page.



To add a reference from one page bean to another

- 1 Make sure Show All Managed Beans is displayed on the toolbar of the Bindings view.

 If this button is not displayed, use the procedure described in "Adding a Reference to Another Page" on page 112.
- 2 To show all managed beans in the project, on the Bindings view toolbar, click Show All Managed Beans.
- 3 Expand the nodes on the Bindings view and right-click the property or managed bean to be referenced to another page.
- 4 Click Add > Reference.
- 5 On the Add Managed Bean Reference Property to Other Managed Beans wizard, select the checkboxes of the managed beans in which you want to reference the property or managed bean selected in step 3.
- 6 Select the **Property** is writable option if the property is to have a setter. If you leave the option cleared, the property will have only a getter.
- 7 Select the Copy source property bindings option to copy the bindings for the existing bean to the new bean.
 - If you leave the option cleared, you will have to reconfigure the reference in the new location.
- 8 Click Finish.

- "Creating a New Method" on page 107
- ▶ "Adding a New Property to a Page Bean" on page 109
- "Customizing Initial Values for a Page" on page 110
- "Showing All Managed Beans" on page 111

Using the Data Source Explorer View

The Data Source Explorer is an Eclipse view that allows you to connect to, navigate, and interact with resources associated with a selected connection profile, such as a database. After you have connected to a database, you can drag resources onto the design canvas for use in Web applications. By default, this view is displayed in the webMethods UI Developer perspective but not displayed in the My webMethods perspective.

Related Topics

- ▶ "Connecting to a Database" on page 113
- "Creating a Database Connector" on page 114

Connecting to a Database

To connect to a database on the Data Source Explorer view, do the following:



To connect to a database in the Data Source Explorer

- 1 If the Data Source Explorer view is not visible: Window > Show View > Other > Connectivity > Data Source Explorer.
- 2 Do one of the following:
 - Click New Connection Profile.
 - -OR-
 - Right-click the Databases node and click New.
- 3 On the New Connection Profile page, choose SQL Model-JDBC Connection and click Next. This selection is appropriate for relational databases.
- In the Name field, type a name for the database connection, to be used in the Data Source Explorer view, and click Next.
- 5 In Select a driver from the dropdown, choose the appropriate database driver.
 - If you don't want to use a Software AG driver, you can locate your own.
- 6 In the URL field type the URL for the database, including the database name. For example:
 - jdbc:wm:sqlserver://localhost:1433;DatabaseName=Northwind
- 7 In the User Name field, type the valid user name for the database.
- 8 In the Password field, type the password.
- 9 Click Test Connection to test the connection to the database.

- 10 Click Finish.
- 11 Right-click the node you have created and click Connect.

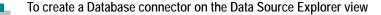
You can now expand the nodes of the database.

Related Topics

"Creating a Database Connector" on page 114

Creating a Database Connector

To create a Database connector on the Data Source Explorer view, do the following:



- 1 Expand the tree view of the Database Explorer view and locate the table or view for which you want to create a Database connector.
- 2 Do either of the following:
 - To accept all of the default values and populate the design canvas with all columns in the table, drag the table from the Data Source Explorer view to the design canvas.
 - -OR-
 - To choose values other than the defaults, do the following:
 - 1 Right-click the table and click Generate Database Connector.
 - 2 If the project in the Project field is correct, click Next.
 - 3 In Managed Bean Name field of the DB Client Properties panel, either accept the default name or type a new one.
 - 4 To allow the database connector to write to the database, clear the Read-only client option, and click Next.
 - Connectors are read-only by default.
 - 5 On the Query Columns Selection page, choose the table columns you want to query.
 - 6 (Optional) To add another table, click Add Table, choose a table, and click OK.
 For tables that have foreign key relations with the main table, the join condition is added automatically.

- 7 Click Finish.
 - The managed bean appears on the Bindings view ("Using the Bindings View" on page 106).
- 8 Drag the managed bean for the database connector from the Bindings view to the design canvas.

Related Topics

- ▶ "JSF Scopes" on page 83
- "Connecting to a Database" on page 113

Using the IS Browser View

The IS Browser view allows you to connect to multiple instances of Integration Server, browse to services, and then create Web service connectors. After you are connected to an instance of Integration Server, several commands are available when you right-click the server instance on the IS Browser view:

Command	Description	
Browse	Opens the Integration Server Administrator in a browser window.	
Refresh	Refreshes the contents of the view.	
Rename	Renames the server instance, as represented in the view. See "Renaming Integration Server on the IS Browser View" on page 123.	
Reconnect	Attempts to reconnect to the server instance. You can use this command if the Connect at Startup property is not set.	
Delete	Delete the server instance from this view. This command has no effect on the actual server instance.	

In connecting to Integration Server, these commands follow the rules described in "Authentication Credentials on the IS Browser View" on page 118.

- "Connecting to Integration Server" on page 116
- ▶ "The IS Browser View Toolbar" on page 117
- ▶ "Authentication Credentials on the IS Browser View" on page 118
- "Creating a Web Service Connector" on page 121
- "Deleting a Web Service Connector" on page 122
- "Renaming Integration Server on the IS Browser View" on page 123

- ▶ "Changing Properties on the IS Browser View" on page 124
- "Displaying Integration Server in a Browser" on page 125

Connecting to Integration Server

To connect to Integration Server on the IS Browser view, do the following:



To connect to Integration Server

- In Designer: Window > Show View > IS Browser.
- 2 Click OK.
- 3 On the IS Browser view, click <u>▼ Menu</u> and then click Create IS Data Provider.
- 4 In the Create IS Data Provider dialog, make the following entries:

Property	Description	
Data Provider Name	Type a descriptive name for Integration Server, to be used within the IS Browser view.	
Data Provider URL	Type the combination of host name (or IP address) and port number for Integration Server.	
Default Path	Type the highest level in Integration Server to be displayed. The default path begins at /.	
Connect at startup	Select this option to connect to the server when you start Designer.	
Remember Authentication	Select this option to remember authentication.	

- 5 Click Finish.
- 6 In the Authentication Request dialog, type the user name and password to be used with the server instance.

When Designer connects with Integration Server, the IS Browser view displays a tree view of the services.



Tip! When you select services on the IS Browser view, you can see their properties in the Properties view.

Related Topics

- ▶ "The IS Browser View Toolbar" on page 117
- ▶ "Authentication Credentials on the IS Browser View" on page 118
- "Creating a Web Service Connector" on page 121
- ▶ "Deleting a Web Service Connector" on page 122
- "Renaming Integration Server on the IS Browser View" on page 123
- "Changing Properties on the IS Browser View" on page 124
- "Displaying Integration Server in a Browser" on page 125

The IS Browser View Toolbar

The toolbar at the top if the IS Browser view has the following buttons:

Button	Name	Purpose
↓ <mark>a</mark> Z	Sort by Name	Sorts child items in a node alphabetically. When the button is not set, items appear in their original order.
→	Filters	Filters the display to contain only items you have selected. Click the button, select the items to be filtered from the view, and click OK .
+	Create IS Data Provider	Creates a connection to Integration Server. See "Connecting to Integration Server" on page 116.
∇	Menu	A pop-up menu for the view.

- "Connecting to Integration Server" on page 116
- "Authentication Credentials on the IS Browser View" on page 118
- "Creating a Web Service Connector" on page 121
- "Deleting a Web Service Connector" on page 122
- "Renaming Integration Server on the IS Browser View" on page 123
- "Changing Properties on the IS Browser View" on page 124
- ▶ "Displaying Integration Server in a Browser" on page 125

Authentication Credentials on the IS Browser View

When the IS Browser view connects to an instance of Integration Server, it uses the following method to assign authorization credentials. By default, any Web service connector created for this connection will use these credentials.

If the Remember authentication credentials property is	Designer does this	
True	Uses credentials of the current (or most recently connected) user.	
False	Uses credentials from these sources, in order:	
	1 The Integration Server Password Manager or	
	2 The Authentication Request dialog box.	
	Use this setting if you want users to authenticate every time they use a Web service or if the service is intended to be called anonymously.	



To change the Remember authentication credentials property for Integration Server

- 1 On the IS Browser view, select the Integration Server for which you want to change properties.
- 2 On the Connection Info tab of the Properties view, set Remember authentication credentials as needed:

This value	Does this		
true	Causes the Web service connector to use the stored credentials.		
false	Causes the credentials to be stored as null values.		

- "Modifying the Authentication Information for Integration Server" on page 119
- "Providing Integration Server Authentication at Run Time" on page 120
- "Connecting to Integration Server" on page 116

- ▶ "The IS Browser View Toolbar" on page 117
- "Creating a Web Service Connector" on page 121
- "Deleting a Web Service Connector" on page 122
- "Renaming Integration Server on the IS Browser View" on page 123
- "Changing Properties on the IS Browser View" on page 124
- "Displaying Integration Server in a Browser" on page 125

Modifying the Authentication Information for Integration Server



To modify the authentication information for Integration Server

- 1 On the Properties view, select the Web service connector () for which you want to change authentication information.
 - If you look at the Data Binding tab of the Properties view, you can see a Data Bindings summary that shows the endpoint address of the connection, the user name, and the encrypted password.
- 2 To change the endpoint address, do the following:
 - a Expand the Web service node, select the Endpoint Address node, and click the Data Binding tab of the Properties view.
 - b On the Value property, type a new endpoint address
 - -OR-

Use a different means of providing the endpoint address, as described in "Providing Integration Server Authentication at Run Time" on page 120.

- 3 To change the user name, do the following:
 - a Expand the Web service and Authentication Info nodes, select the User Name node, and click the Data Binding tab of the Properties view.
 - b On the Value property, type a new user name
 - -OR-

Use a different means of providing the endpoint address, as described in "Providing Integration Server Authentication at Run Time" on page 120.

- 4 To change the password, do the following:
 - a Expand the Web service and Authentication Info nodes, select the User Name node, and click the Data Binding tab of the Properties view.
 - b On the Value property, type a new user name
 - -OR-

Use a different means of providing the endpoint address, as described in "Providing Integration Server Authentication at Run Time" on page 120.



Note: By typing in a new password, you are storing the password in unencrypted form.

Related Topics

▶ "Providing Integration Server Authentication at Run Time" on page 120

Providing Integration Server Authentication at Run Time

In addition to storing the authentication credentials and endpoint address in place at the time the Web service connector was created, there are means of providing them dynamically at run time, as described in the following sections:

- "Using Preferences for Authentication" on page 120
- "Using Global Wiring for Authentication" on page 121
- "Using User Attributes for Authentication" on page 121

Using Preferences for Authentication

You can manage the authentication information for a Web service connector by making it configurable through portlet preferences.



To configure authentication information through portlet preferences

- 1 Create user name and password preferences using the procedure described in "Creating Portlet Preferences" on page 89.
- 2 On the Bindings view, expand the Web service and Authentication Info nodes
- 3 Select the User Name node, click the Data Binding tab of the Properties view, and click #{...} at the right of the Value property.
- 4 On the Expression Binding wizard, expand the nodes, select the user name preference you have created (preferences are identified by the [22] icon), and click **OK**.
- 5 Repeat steps 3 and 4 for the password preference.

Using Global Wiring for Authentication

Global wiring is the technique used by My webMethods Server for pulling portlet preferences from other portlets. Using global wiring, you can have a single Properties portlet drive the values (Endpoint and Authentication Information) for many different portlets. For more information about wiring preferences, see "Using Preferences to Perform Wiring" on page 90.

Using User Attributes for Authentication

The previous options have implied a per-system type of configuration where the same authentication credentials are used to contact the Web service provider (Integration Server). Your usage, however, may require a per-user connection. There are many ways of achieving a per-user connection. You could change the Scope of your preferences, wire your preferences to user attributes, or you could directly use user attributes in your portlet by declaring them in your portlet application and then configuring them at run time. For more information, see "User Attributes" on page 95.

Related Topics

"Modifying the Authentication Information for Integration Server" on page 119

Creating a Web Service Connector

To create a Web service connector on the IS Browser view, do the following:



To create a Web service connector on the IS Browser view

- 1 To turn off WSDL validation, in Designer: Window > Preferences > Validation.
- 2 For WSDL Validator, clear the Manual and Build options.
- 3 Expand the tree view of the IS Browser view and locate the service for which you want to create a Web service connector.
- 4 Drag the service from the IS Browser view to the design canvas.
 - You cannot drop a service outside of a Form control unless the service user interface will not have any input parameters.
- 5 On the Select WSDL Location panel, accept the defaults and click Next.
- 6 On the Select Operation panel, select the operation you want to use, and click Next.
 - A port type is an abstract collection of supported operations. For Web services with multiple port types, all port types and operations are displayed.
- 7 On the Java Type panel, click Next.

- 8 On the Add Managed Bean Reference Property to Other Beans panel, click Next.
- 9 On the Dynamic Data Binding for Bean Properties panel, verify that the user name and password for Integration Server are correct in the Binding Expression column and click Finish.

On this panel, there are other bean properties you might want to use. For example, if you want the Web service connector automatically invoked without having to submit a command, set the autoRefresh Bean Property to True.

Related Topics

- "Connecting to Integration Server" on page 116
- ▶ "The IS Browser View Toolbar" on page 117
- "Authentication Credentials on the IS Browser View" on page 118
- "Deleting a Web Service Connector" on page 122
- "Renaming Integration Server on the IS Browser View" on page 123
- "Changing Properties on the IS Browser View" on page 124
- "Displaying Integration Server in a Browser" on page 125

Deleting a Web Service Connector

To delete a Web service connector, you need to keep in mind whether you want only to delete a reference to the Web service or delete it completely from your portlet application.



To delete a Web service connector reference

- 1 On the Bindings view, locate the Web service connector reference () to be removed.
- 2 Right-click the Web service connector reference and then click Delete.
- 3 On the Delete dialog box, choose one of these options:
 - Delete both this reference and the referenced managed bean (the default)
 - Delete just this reference and leave the referenced managed bean
- 4 Click Yes.

- "Connecting to Integration Server" on page 116
- ▶ "The IS Browser View Toolbar" on page 117
- ▶ "Authentication Credentials on the IS Browser View" on page 118

- "Creating a Web Service Connector" on page 121
- ▶ "Renaming Integration Server on the IS Browser View" on page 123
- "Changing Properties on the IS Browser View" on page 124
- "Displaying Integration Server in a Browser" on page 125

Renaming Integration Server on the IS Browser View

You can rename the label given to an Integration Server on the IS Browser view. The renaming process has no effect on the actual instance of Integration Server.



To rename Integration Server on the IS Browser view

- 1 On the IS Browser view, right-click the node of the My webMethods Server instance and click Rename.
- 2 Type a new name to represent the Integration Server and press ENTER or click away from the field.

- "Connecting to Integration Server" on page 116
- ▶ "The IS Browser View Toolbar" on page 117
- ▶ "Authentication Credentials on the IS Browser View" on page 118
- "Creating a Web Service Connector" on page 121
- "Deleting a Web Service Connector" on page 122
- "Changing Properties on the IS Browser View" on page 124
- ▶ "Displaying Integration Server in a Browser" on page 125

Changing Properties on the IS Browser View

For any Integration Server instance on the IS Browser view, you can change properties, as described here:



To change properties on the IS Browser view

- 1 On the IS Browser view, select the node representing the Integration Server instance.
- 2 On the Properties view, change properties as follows:

Tab	Property	Description
General	Root Path	Type the highest level in Integration Server to be displayed. The default path begins at /.
	Connection URL	Type the combination of host name (or IP address) and port number for Integration Server.
Connection Info	Remember authentication credentials	Choose whether or not Designer stores login credentials:
		true-stores credentials
		false—does not store credentials
	Connect at startup	Choose behavior when Designer starts:
		true—connects to the server
		false—does not connect to the server

3 Click away from the property.

- ▶ "Connecting to Integration Server" on page 116
- ▶ "The IS Browser View Toolbar" on page 117
- ▶ "Authentication Credentials on the IS Browser View" on page 118
- "Creating a Web Service Connector" on page 121
- "Deleting a Web Service Connector" on page 122
- "Renaming Integration Server on the IS Browser View" on page 123
- "Displaying Integration Server in a Browser" on page 125

Displaying Integration Server in a Browser

From the IS Browser view, it is possible to display Integration Server in a browser.



To display Integration Server in a browser

- 1 On the IS Browser view, do either of the following to the node representing the Integration Server instance or any node nested under it:
 - Right-click the node and click Browse.
 - Double-click the node.
- 2 If required, provide a valid user name and password.

The specific node within Integration Server is displayed in an internal Web browser. To specify the use of an external Web browser, see "Specifying an External Web Browser" on page 155.

Related Topics

- "Connecting to Integration Server" on page 116
- ▶ "The IS Browser View Toolbar" on page 117
- ▶ "Authentication Credentials on the IS Browser View" on page 118
- "Creating a Web Service Connector" on page 121
- ▶ "Deleting a Web Service Connector" on page 122
- "Renaming Integration Server on the IS Browser View" on page 123
- "Changing Properties on the IS Browser View" on page 124
- "Specifying an External Web Browser" on page 155

Using the MWS Admin View

The MWS Admin view allows you to connect to multiple instances of My webMethods Server, browse to server resources, publish portlets to the servers, and perform some other activities that you might otherwise perform as a system administrator directly on the server. After you are connected to an instance of My webMethods Server, several

commands are available when you right-click the server instance on the MWS Admin view:

Command	Description	
Browse	Opens My webMethods Server in a browser window.	
Refresh	Refreshes the contents of the view.	
Rename	Renames the server instance, as represented in the view. See "Renaming My webMethods Server on the MWS Admin View" of page 133.	
Reconnect	Attempts to reconnect to the server instance. You can use this command if the Connect at Startup property is not set.	
Delete	Delete the server instance from this view. This command has no effect on the actual server instance.	

In connecting to My webMethods Server, these commands follow the rules described in "Authentication Credentials on the MWS Admin View" on page 129.

- "Connecting to My webMethods Server" on page 127
- "The MWS Admin View Toolbar" on page 128
- ▶ "Authentication Credentials on the MWS Admin View" on page 129
- "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Exporting from My webMethods Server" on page 130
- "Importing to My webMethods Server" on page 132
- "Renaming My webMethods Server on the MWS Admin View" on page 133
- "Changing Properties on the MWS Admin View" on page 134
- ▶ "Other Actions You Can Perform on the MWS Admin View" on page 135
- "Displaying My webMethods Server in a Browser" on page 136

Connecting to My webMethods Server

To connect to My webMethods Server on the MWS Admin view, do the following:



To connect to My webMethods Server

- 1 In Designer: Window > Show View > MWS Admin.
- 2 Click OK.
- 3 On the MWS Admin view, click w Menu and then click Create MWS Data Provider.
- 4 In the Create MWS Data Provider dialog, make the following entries:

Property	Description	
Data Provider Name	Type a descriptive name for My webMethods Server, to be used within the IS Browser view.	
Data Provider URL	Type the combination of host name (or IP address) and port number for My webMethods Server.	
Default Path	Type the highest level in My webMethods Server to be displayed. The default path begins at folder.root.	
Connect at startup	Select this option to connect to My webMethods Server when Designer starts. We recommend that you clear this option.	
Remember Authentication	Select this option to remember authentication.	

- 5 Click Finish.
- 6 In the Authentication Request dialog, type the user name and password to be used with the server instance.
 - When Designer connects with My webMethods Server, the MWS Admin view displays a tree view of server resources.



Tip! When you select server resources on the MWS Admin view, you can see their properties on the Properties view. If you modify properties of any object on the server on the Properties view, the changes are published to the server using the credentials of your current connection top My webMethods Server.

- "The MWS Admin View Toolbar" on page 128
- ▶ "Authentication Credentials on the MWS Admin View" on page 129

- ▶ "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Exporting from My webMethods Server" on page 130
- "Importing to My webMethods Server" on page 132
- "Renaming My webMethods Server on the MWS Admin View" on page 133
- ▶ "Changing Properties on the MWS Admin View" on page 134
- ▶ "Other Actions You Can Perform on the MWS Admin View" on page 135
- "Displaying My webMethods Server in a Browser" on page 136

The MWS Admin View Toolbar

The toolbar at the top if the MWS Admin view has the following buttons:

Button	Name	Purpose
↓ <mark>a</mark> ▼	Sort by Name	Sorts child items in a node alphabetically. When the button is not set, items appear in their original order.
→	Filters	Filters the display to contain only items you have selected. Click the button, select the items to be filtered from the view, and click OK .
+	Create IS Data Provider	Creates a connection to My webMethods Server. See "Connecting to My webMethods Server" on page 127.
∇	Menu	A pop-up menu for the view.

- "Connecting to My webMethods Server" on page 127
- ▶ "Authentication Credentials on the MWS Admin View" on page 129
- "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Exporting from My webMethods Server" on page 130
- "Importing to My webMethods Server" on page 132
- "Renaming My webMethods Server on the MWS Admin View" on page 133
- "Changing Properties on the MWS Admin View" on page 134
- "Other Actions You Can Perform on the MWS Admin View" on page 135
- "Displaying My webMethods Server in a Browser" on page 136

Authentication Credentials on the MWS Admin View

When the MWS Browser view connects to an instance of My webMethods Server, it uses this method to assign authentication credentials:

If the Remember Authentication property is	Designer does this
True	Uses credentials of the current (or most recently connected) user.
False	Uses credentials from the Authentication Request dialog box.

Related Topics

- "Connecting to My webMethods Server" on page 127
- ► "The MWS Admin View Toolbar" on page 128
- ▶ "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Exporting from My webMethods Server" on page 130
- "Importing to My webMethods Server" on page 132
- "Renaming My webMethods Server on the MWS Admin View" on page 133
- "Changing Properties on the MWS Admin View" on page 134
- ▶ "Other Actions You Can Perform on the MWS Admin View" on page 135
- "Displaying My webMethods Server in a Browser" on page 136

Deploying Portlets to My webMethods Server

After you have developed a portlet, you no longer need to publish it iteratively to My webMethods Server using the Servers view. Rather, you can deploy a portlet to any instance of My webMethods Server that is connected to Designer on the MWS Admin view.



To publish a portlet to My webMethods Server using the MWS Admin view

- 1 On the MWS Admin view, expand the server resources in My webMethods Server until you expose the page where the portlet is to reside.
- 2 Right-click the page, on the Create Data dialog, expand the Portlet node and locate the portlet.
- 3 In the Name field, type a display name for the portlet and click Finish.

Related Topics

- "Connecting to My webMethods Server" on page 127
- ▶ "The MWS Admin View Toolbar" on page 128
- "Authentication Credentials on the MWS Admin View" on page 129
- "Deploying Portlets to My webMethods Server" on page 129
- "Exporting from My webMethods Server" on page 130
- "Importing to My webMethods Server" on page 132
- "Renaming My webMethods Server on the MWS Admin View" on page 133
- "Changing Properties on the MWS Admin View" on page 134
- "Other Actions You Can Perform on the MWS Admin View" on page 135
- ▶ "Displaying My webMethods Server in a Browser" on page 136
- "Using the Solutions View" on page 149
- "Uninstalling Portlet Applications from My webMethods Server" on page 78

Exporting from My webMethods Server

You can export portal pages from My webMethods Server to the webMethods CAF in either of two forms:

- XML Import file—The raw XML that is a serializable representation of a portlet or portal page.
- Deployable component (CDP file) A binary representation of a portlet or portal page.

To export a portlet or portal page from the server to the webMethods CAF, do the following:



To export a portlet or portal page from the server to the webMethods CAF

- On the MWS Admin view, right-click the portlet or portal page and click Export > Export.
- 2 From the **Project** list in the Export Data wizard, do one of the following:
 - Choose the portlet application to which the object should be exported
 - -OR-
 - Click New and create a new portlet application, as described in "Associating Portlet Applications with My webMethods Server" on page 76.

3 To choose between an XML import file and a deployable component, do the following:

To create this	Do this
XML import file	Select the Generate an XML Import File option
Deployable Component	Select the Generate a Deployable Component option

If you have exported an XML import file, you can find it on the Navigator view, in this location:

portlet_application/WebContent/WEB-INF/config/xmlImport.xml

If you have exported deployable component, you can find it on the Navigator view, in this location:

portlet_application/WebContent/WEB-INF/config/component_name.cdp

You can now deploy this portlet application to another instance of My webMethods Server or another location on the same server.

- "Connecting to My webMethods Server" on page 127
- ▶ "The MWS Admin View Toolbar" on page 128
- ▶ "Authentication Credentials on the MWS Admin View" on page 129
- ▶ "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Importing to My webMethods Server" on page 132
- "Renaming My webMethods Server on the MWS Admin View" on page 133
- "Changing Properties on the MWS Admin View" on page 134
- "Other Actions You Can Perform on the MWS Admin View" on page 135
- "Displaying My webMethods Server in a Browser" on page 136

Importing to My webMethods Server

You can export portlet applications from the webMethods CAF to My webMethods Server in the form either of an XML import file or a deployable component by doing the following:



To import from the webMethods CAF to My webMethods Server

- 1 On the MWS Admin view, right-click the folder or page to receive the import and click Export > Import.
- 2 In the Import Data wizard, click Browse.
- 3 In the top field, type the name of a resource to import, either xmlImport.xml or *component_name*.cdp.
- 4 Use the Matching resources and In folders boxes to identify the import file and click OK.
- 5 Click Finish.

The portlet or page is deployed to the specified location on My webMethods Server.

- "Connecting to My webMethods Server" on page 127
- "The MWS Admin View Toolbar" on page 128
- ▶ "Authentication Credentials on the MWS Admin View" on page 129
- "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Exporting from My webMethods Server" on page 130
- "Renaming My webMethods Server on the MWS Admin View" on page 133
- "Changing Properties on the MWS Admin View" on page 134
- "Other Actions You Can Perform on the MWS Admin View" on page 135
- "Displaying My webMethods Server in a Browser" on page 136

Renaming My webMethods Server on the MWS Admin View

You can rename the label given to an instance of My webMethods Server on the MWS Admin view. The renaming process has no effect on the actual instance of My webMethods Server.



To rename My webMethods Server on the MWS Admin view

- 1 On the MWS Admin view, right-click the node of the My webMethods Server instance and click Rename.
- 2 Type a new name to represent My webMethods Server and press ENTER or click away from the field.

- "Connecting to My webMethods Server" on page 127
- ▶ "The MWS Admin View Toolbar" on page 128
- "Authentication Credentials on the MWS Admin View" on page 129
- ▶ "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Exporting from My webMethods Server" on page 130
- "Importing to My webMethods Server" on page 132
- "Changing Properties on the MWS Admin View" on page 134
- "Other Actions You Can Perform on the MWS Admin View" on page 135
- "Displaying My webMethods Server in a Browser" on page 136

Changing Properties on the MWS Admin View

For any Integration Server instance on the IS Browser view, you can change properties, as described here:



To change properties on the MWS Admin view

- 1 On the IS Browser view, select the node representing the Integration Server instance.
- 2 On the Properties view, change properties as follows:

Tab	Property	Description
General	Root Path	Type the highest level in Integration Server to be displayed. The default path begins at /.
	Connection URL	Type the combination of host name (or IP address) and port number for Integration Server.
Connection Info	Remember authentication credentials	Choose whether or not Designer stores login credentials:
		true-stores credentials
		false—does not store credentials
	Connect at startup	Choose behavior when Designer starts:
		true—connects to the server
		false—does not connect to the server

3 Click away from the property.

- "Connecting to My webMethods Server" on page 127
- ▶ "The MWS Admin View Toolbar" on page 128
- ▶ "Authentication Credentials on the MWS Admin View" on page 129
- "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Exporting from My webMethods Server" on page 130
- "Importing to My webMethods Server" on page 132

- "Renaming My webMethods Server on the MWS Admin View" on page 133
- ▶ "Other Actions You Can Perform on the MWS Admin View" on page 135
- "Displaying My webMethods Server in a Browser" on page 136

Other Actions You Can Perform on the MWS Admin View

In addition to publishing portlets to My webMethods Server, there are several other actions you can perform on the server, including the following:

For this action	Do this on the MWS Admin view
Delete a portlet or a page	Right-click the portlet and click Delete.
Move a portlet	Drag the portlet to another position within the same page, to a different page, or to a different server.
Copy a portlet	Right-click the portlet and click Edit > Copy. Then right-click another page and click Edit > Paste. You can copy resources from one server to another this way.
Create a variety of server resources	Right-click the node where you want to add a resource (such as folders, links, and portlets) and click New. On the tree view, select the resource you want to create, type a display name for the resource, and click Finish.
Change properties of a server resource	Select the server resource, such as a portlet or page. Then go to the Properties view and make changes as needed. For example, you can create or modify aliases to server resources such as pages or portlets.
Publish documents to the server	Drag the document from the Navigator view to the MWS Admin view. You can also drag documents from the MWS Admin view to the Navigator view.



Note: When you are making changes on the MWS Admin view, you have privileges on the server that are appropriate for the credentials with which you were authenticated. If you have system administrator privileges, making changes indiscriminately can corrupt other portions of the server.

- "Connecting to My webMethods Server" on page 127
- ▶ "The MWS Admin View Toolbar" on page 128
- "Authentication Credentials on the MWS Admin View" on page 129

- "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Exporting from My webMethods Server" on page 130
- "Importing to My webMethods Server" on page 132
- "Renaming My webMethods Server on the MWS Admin View" on page 133
- "Changing Properties on the MWS Admin View" on page 134
- "Displaying My webMethods Server in a Browser" on page 136

Displaying My webMethods Server in a Browser

From the MWS Admin view, it is possible to display My webMethods Server in a browser.



To display My webMethods Server in a browser

- 1 On the MWS Admin view, do either of the following to the node representing the My webMethods Server instance or any node nested under it:
 - Right-click the node and click Browse.
 - Double-click the node.
- 2 If required, provide a valid user name and password.

The specific node within My webMethods Server is displayed in an internal Web browser. To specify the use of an external Web browser, see "Specifying an External Web Browser" on page 155.

- "Connecting to My webMethods Server" on page 127
- "The MWS Admin View Toolbar" on page 128
- ▶ "Authentication Credentials on the MWS Admin View" on page 129
- "Deploying Portlets to My webMethods Server" on page 129
- "Uninstalling Portlet Applications from My webMethods Server" on page 78
- "Exporting from My webMethods Server" on page 130
- "Importing to My webMethods Server" on page 132
- "Renaming My webMethods Server on the MWS Admin View" on page 133
- "Changing Properties on the MWS Admin View" on page 134

- ▶ "Other Actions You Can Perform on the MWS Admin View" on page 135
- ▶ "Specifying an External Web Browser" on page 155

Using the Outline View

The Outline view is a standard Eclipse view that, in the webMethods CAF, provides a tree view of the portlet that is currently open in the design canvas. This view is useful in locating specific controls within a complex portlet and for moving controls from one location to another. You have the choice of two viewing mode, which you can select from the Outline view toolbar:

Button	Name	Purpose
造	Outline	Displays the controls in a tree view that you can expand and collapse as needed. When you select a control in this view, the design canvas displays the control. The default mode
	Overview	Displays a smaller version of the design canvas. The shaded area corresponds to the visible portion of the design canvas. As you drag the shaded area in the Outline view, the display in the design canvas changes to match it.



To use the Outline view

- 1 If the Outline view is not visible, in Designer: Window > Show View > Outline.
- 2 If you need to display a portlet in the design canvas, expand the portlet application and double-click this node:

portlet_application/WebContent/portlet_name/default.view

The tree view of the portlet appears on the Outline view. The top-level node is the Form control that contains all other controls in the portlet.

- 3 Expand the nodes as needed to display the hierarchical view of the portlet.
- 4 Perform the following actions on the Outline view as needed.

To do this	Do this on the Outline view
Locate a control in the design canvas	Expand nodes until you find the control. When you click the control, it is highlighted in the design canvas.
Move a control in the design canvas	Drag the control to a new location. You can only drop a control in a valid location. All children of the control are also moved.

To do this	Do this on the Outline view
Copy a control to another portlet or portlet application	Right-click the control and click Copy. (To remove the control from the first portlet, click Cut.) Open the new portlet in the design canvas. On the Outline view, right-click the location where the control is to reside and click Paste. All children of the control are included.
Delete a control	Right-click the control and click Delete. All children of the control are also deleted.
Display Overview mode	On the toolbar, click Overview to change the Outline view to Overview mode.
Display Outline mode	On the toolbar, click 🔓 Outline to change the Outline view to Outline mode.

Using the Palette View

The Palette view is a tree view that contains the various objects (controls, data objects, and converters) you can use to create the portlets in a portlet application. This view is populated only when a portlet is open in the design canvas.

The tree view contains the following nodes, each of which has objects you can move onto the design canvas. Actual controls are described in the Controls Reference. See "User Interface Controls Reference" on page 179.

Palette node	Contains
Control	JSF controls, organized by how they are to be used.
Behavior	Objects that allow developers to easily manipulate controls on the design canvas.
Command	Controls that provide various ways of submitting commands.
Dialog	Controls that alert users or retrieve complex user input.
Input	Controls for providing user input, such as check boxes and options.
List	Controls for managing lists and tabs.
Logic	Controls that add logic to the portlet, such as if/else and iteration.
Output	Controls that manage the display of output to the user.
Refresh	Controls that provide ways to refresh the page.
Panel	Controls that help in organizing controls on a page.

Palette node	Contains
Portlet	Controls used to provide linking and navigation among portlets and pages.
Scripts	Controls to add JavaScript events to portlets, such as to bind controls together.
Search	Controls to add search bars and results tables.
Table	Controls to add tables to portlets.
Toggle	Controls to manage toggling the visibility among controls in a portlet.
Webapp	Controls to provide navigation in a Web application.
Data	Objects that manage data.
Primitives	Objects that add simple properties to other Java beans.
Task	Objects used to construct tasks.

Each palette object has an icon in the tree view. When an object is on the design canvas, that same icon appears in a ToolTip when you pass the mouse over the object, and beside the object in the Outline view.

Converters and Validators, also available from the Palette view, are initially hidden because you can assign them to controls from the Properties view. To make them visible in the Palette view, see "Customizing the Palette View" on page 142.

Palette node	Contains
Converter	A variety of converters that convert the native value in a control to a Java object
Currency	Converters that take into account localized currency formatting.
Formatted	Converters that take into account localized number formatting.
Validator	Validators that provide client-side and server-side validation.
Formatted	Validators that take into account localized formatting.

- "Moving Objects from the Palette View" on page 140
- "Managing Favorites in the Palette View" on page 141
- "Customizing the Palette View" on page 142
- ▶ "The Palette View Toolbar" on page 143

- ▶ "Displaying Deprecated Items" on page 143
- ▶ "Filtering Controls" on page 144
- ▶ "Using the Outline View" on page 137
- "User Interface Controls Reference" on page 179
- "Converters Reference" on page 633
- "Validators Reference" on page 669

Moving Objects from the Palette View



Moving objects from the Palette view to the design canvas

- 1 Expand the nodes in the tree view until you find the palette object you need.
- 2 Drag the control or data type until you can drop it in the desired container in the design canvas.

You can only drop an object in a container whose boundary is highlighted when you drag the object over it. If the **O** overlay is visible on the cursor, you cannot drop the object. A drop hint appears to provide some descriptive text regarding where the new object will be inserted.



Moving data objects from the Palette view to the Bindings view

- 1 Expand the nodes in the tree view until you find the palette object you need.
- 2 Drag a data object to the Bindings view and:
 - Drop onto the Page Bean node () to add the selected data object to the current page bean
 - Drop onto the Managed Beans node (🗞) to create a new managed bean
 - Drop on any other complex type to add a property to that type

- "Managing Favorites in the Palette View" on page 141
- "Customizing the Palette View" on page 142
- ▶ "The Palette View Toolbar" on page 143
- ▶ "Displaying Deprecated Items" on page 143
- "Filtering Controls" on page 144

- "User Interface Controls Reference" on page 179
- "Converters Reference" on page 633
- "Validators Reference" on page 669

Managing Favorites in the Palette View

The Palette view contains a Favorites node in which you can place shortcuts to palette objects you use on a regular basis. You can drag these shortcuts to the design canvas exactly as you would the original object.



To add a palette object to the Favorites node

- 1 Expand the nodes in the tree view until you find the palette object for which a shortcut should be added to the Favorites node.
- 2 Right-click the palette object and click Add To Favorites.



To remove a palette object from the Favorites node

- 1 In the Palette view, right-click either of the following:
 - A shortcut in the Favorites node
 - -OR-
 - An object in the tree view (that is also represented in the Favorites node by a shortcut)
- Click Delete From Favorites.

- ▶ "Moving Objects from the Palette View" on page 140
- "Customizing the Palette View" on page 142
- ▶ "The Palette View Toolbar" on page 143
- "Displaying Deprecated Items" on page 143
- ▶ "Filtering Controls" on page 144
- "User Interface Controls Reference" on page 179
- "Converters Reference" on page 633
- "Validators Reference" on page 669

Customizing the Palette View

You can customize the Palette view so that it contains only the group of palette objects you are likely to need on a regular basis. You can hide individual objects or nodes, or cause nodes to display as expanded at start up. You can also provide your own names for objects and nodes.



To customize the Palette view

- 1 On the Palette view, click Menu and then click Customize.
- 2 On the Customize Palette page, navigate to a node or object and do any of the following:
 - In the Name field, type a Display name to be used for the node or object. This field is not valid for shortcuts in the Favorites node.
 - In the Description field, type a description that will be used for ToolTips in Designer. This field is not valid for shortcuts in the Favorites node.
 - Select the Hide checkbox to hide the node or object in the Palette tree view. If you hide a node, all objects nested in the node are also hidden.
 - Select the Expanded Initially at startup checkbox if you want the node to be expanded when the Palette view is opened. To be expanded, any parent nodes must also have this checkbox selected.
- 3 Click OK.

- "Moving Objects from the Palette View" on page 140
- "Managing Favorites in the Palette View" on page 141
- ▶ "The Palette View Toolbar" on page 143
- "Displaying Deprecated Items" on page 143
- "Filtering Controls" on page 144
- "User Interface Controls Reference" on page 179
- "Converters Reference" on page 633
- "Validators Reference" on page 669

The Palette View Toolbar

The toolbar at the top of the Palette view provides mechanisms for navigating in the view.

Button	Name	Purpose
	Collapse All	Collapse all nodes and show only the top set of nodes. A quick way to clean up the view.
	Home	Sets the top node of the Palette view to be the View Root control. Enabled only after the Go Into action has set another node as the top node.
\$	Back	Sets the top node of the Palette view to be the parent node of the current node. Enabled only after the Go Into action has set a node as the top node.
\$	Go Into	Sets the top node of the editor to be the currently selected control. This action enables you to drill into a specific section of the palette for reduced clutter.
∇	Menu	A pop-up menu for the view.

Related Topics

- ▶ "Moving Objects from the Palette View" on page 140
- "Managing Favorites in the Palette View" on page 141
- ▶ "Customizing the Palette View" on page 142
- ▶ "Displaying Deprecated Items" on page 143
- "Filtering Controls" on page 144

Displaying Deprecated Items

By default, deprecated items in the Palette view are hidden. You can display those items by doing the following:



To display deprecated items in the Palette view

- On the Palette view, click

 Menu.

 If Hide Deprecated Items has a check mark beside it, all deprecated items are hidden.
- 2 To clear the check mark, click Hide Deprecated Items.

In the same way, you can later hide deprecated items.

Related Topics

- ▶ "Moving Objects from the Palette View" on page 140
- "Managing Favorites in the Palette View" on page 141
- ▶ "Customizing the Palette View" on page 142
- ▶ "Filtering Controls" on page 144
- "User Interface Controls Reference" on page 179
- "Converters Reference" on page 633
- "Validators Reference" on page 669

Filtering Controls

The Palette view contains a large number of objects. If you the name of an object, you can use the filter field at the top of the Palette view to find the object quickly.



To filter controls in the Palette view

In the filter field at the top of the Palette view, type a word that is part of the name of a palette object, using the following rules:

- The word does not have to represent the first word in the name. Typing group finds any object with group in the title, such as Radio Group and Page Group.
- Typing a space at the end of a word causes the filter to assume that word is the first word in the name. Typing "group " causes the filter to hide everything because there is no object whose name begins with group.
- Always start typing at the beginning of a word. Typing rou does not find the word group.
- Use wildcards as needed. Typing *oup or ?roup finds the word group.
- You can type partial words, but the more letters you type will filter out more objects. If you type pa, names containing Parameter, Panel, and Page are displayed. If you type pan, only names containing Panel are displayed.

- "Moving Objects from the Palette View" on page 140
- "Managing Favorites in the Palette View" on page 141
- "Customizing the Palette View" on page 142
- "Displaying Deprecated Items" on page 143

Using the Properties View

The Properties view is an Eclipse view that displays the names and values of properties belonging to resources displayed elsewhere in Designer. When the focus is in a particular view file on the design canvas, properties for resources in that view file appear on the Properties view:

- IS Browser view—See, but not modify, resources for services.
- MWS Admin view See and modify properties that appear on the Properties page for a resource. For example, you can create or modify aliases for a page.
- Bindings view See and modify properties associated with managed beans and other resources in the view.
- Design canvas and Outline view—See and modify properties associates with controls.
- Navigator view and Package view See properties associated with various resources in a portlet.

Related Topics

- ▶ "The Properties View Toolbar" on page 145
- ▶ "Actions in the Properties View" on page 146
- "Adding Converters or Validators to a Control" on page 147
- ▶ "Binding Data to a Control" on page 148

The Properties View Toolbar

The toolbar at the top if the Properties view has the following buttons:

Button	Name	Purpose
<u> </u>	Hide Expert Items	Hides or displays expert properties, which can appear in an additional tab, such as the Client-Side Events tab for the Text Input control, or as additional properties in an existing tab.
Z	Show Deprecated Items	Displays or hides deprecated properties, which appear as additional properties in a tab.

Button	Name	Purpose
Q	Go Up	For any child control, selects the parent control in the design canvas and displays its properties in the Properties view. You can also click the Down arrow immediately to the right of this button and click any control in the menu. The top item of the menu is the parent of the current control, followed by each succeeding parent control in turn.
∇	Menu	A pop-up menu for the view.

Related Topics

- ▶ "Actions in the Properties View" on page 146
- ▶ "Adding Converters or Validators to a Control" on page 147
- ▶ "Binding Data to a Control" on page 148

Actions in the Properties View

To organize data, the Properties view uses tabs that extend from the left side of the table. The tabs that are displayed depend on the nature of the resource for which properties are being exposed. There are multiple ways you can modify properties on the Properties view, as shown in this example:



To modify properties on the Properties view

- Locate the Text Input control in the Input node of the Palette view, drag it to a location on the design canvas, and make sure it is selected.
- 2 On the General tab of the Properties editor, type a control ID in the ID field and then click away from the field.
 - The Value field of the Value tab is another place where you might type text. In an output control such as a Command Button, text you type in the Value field becomes a label for the button.
- On the General tab, set the Rendered property to true or false to cause the control to be displayed or hidden on the page in My webMethods Server.
 - There are various boolean properties associated with controls. For example, the **Disabled** property on the Display tab causes the control to be disabled if it is set to true.
- On the Width list of the Display tab, choose the width of the input field, in characters.

 Alternatively type an integer to represent the number of characters for the input field.

5 Click #{...} at the right of a property (where the button exists) to create a binding expression.

See "User Interface Controls Reference" on page 179.

- 6 Click ____ at the right of a property (where the button exists) to create a JavaScript event associated with the property.
 - The ability to create JavaScript is typically provided for client-side action related to using a mouse or other input device.
- 7 Click the control on the design canvas to see any changes you have made. For example, if you have typed text in the Value property of the Text Input control, that text now appears in the control itself. When you publish the portlet to the server, that text appears on the page.

Related Topics

- ▶ "The Properties View Toolbar" on page 145
- "Adding Converters or Validators to a Control" on page 147
- ▶ "Binding Data to a Control" on page 148
- ▶ "User Interface Controls Reference" on page 179

Adding Converters or Validators to a Control

From the Properties view, you can add Converters to input and output controls, and you can add Validators to input controls.



To add converters or validators to a control

- 1 On the design canvas select an input or output control to which you want to add a converter or a validator.
- 2 On the Properties view click either of these tabs:
 - Conversion to add a Converter
 - Validation to add a Validator (input controls only)
- 3 To the right of the Converters or Validators table, click ▼ Add.
- 4 On the menu, click the Converter or Validator to be added.

A control can have only one converter but multiple validators. When a control that has at least one Converter or Validator is selected on the design canvas, a push pin icon () appears at the upper right corner. Icons representing the converter or validators are also displayed.

Some Converters and Validators have properties that you can modify. To display the properties, do the following:



To display the properties of a Converter or Validator

- 1 On toolbar of the Outline view, click **a Outline** to change the Outline view to Outline mode.
- 2 Expand the nodes until you can select the Converter or Validator.
 - If the Converter or Validator has properties you can modify, they appear in the Properties view.
- 3 After you have changed properties, on the Properties view toolbar, click **QGO Up** to select the parent control in the design canvas.

Related Topics

- ▶ "The Properties View Toolbar" on page 145
- ▶ "Actions in the Properties View" on page 146
- "Adding Converters or Validators to a Control" on page 147
- ▶ "Binding Data to a Control" on page 148

Binding Data to a Control

Binding expressions are the means by which you bind data to controls. JSF binding expressions take the following form:

#{name_of managed_bean.property}

Properties can have sub-properties, in which case a binding expression might have a more complex form, such as this:

#{name_of managed_bean.property.subproperty1.subproperty2}

On the Properties view, you can bind data to most of the properties available for a control. Specific cases of binding data are described elsewhere; the basic procedure for data binding is as follows:



To bind data to a control

- 1 On the design canvas, select the control for which you want to create a binding expression.
- 2 On the Properties view, locate a property and click #{...} at the right of that property.

In the Expression Binding wizard, expand the Page Bean node (), click the data to be bound to the control, and click **OK**.

The resulting binding expression appears in the field associated with the property.

Related Topics

- ▶ "The Properties View Toolbar" on page 145
- ▶ "Actions in the Properties View" on page 146
- "Adding Converters or Validators to a Control" on page 147

Using the Solutions View

The Solutions view provides a quick way to see important assets in a Web or portlet application project and perform some actions against them.



To use the Solutions view

- 1 On the Solutions view, expand the User Interfaces node to expose the assets in the We or portlet application.
- 2 Expand the nodes as needed to display the hierarchical views of the Web or portlet application projects in your workspace.
- To display the Portlet Application Configuration editor (portlet.xml) for a portlet application, double-click any portlet belonging to the portlet application.
- 4 To display the design canvas for a particular view file, double-click the view file.
- 5 To display the Faces Configuration editor (faces-config.xml) for a portlet application, double-click Faces Configuration.

- ▶ "Portlet Applications in the Solutions View" on page 150
- "Web Applications in the Solutions View" on page 153
- ▶ "Database Connectors in the Solutions View" on page 154

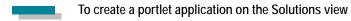
Portlet Applications in the Solutions View

You can perform the following actions on portlet applications in the Solutions view:

- "Creating a Portlet Application on the Solutions View" on page 150
- "Creating a Portlet on the Solutions View" on page 150
- "Creating a Portlet View File on the Solutions View" on page 152

Creating a Portlet Application on the Solutions View

Using the Solutions view, you can create a new portlet application.



- 1 On the Solutions view, right-click the User Interfaces node and click New Portlet Application.
- 2 In the Project Name field of the New Dynamic Web Project window, type a name for the portlet application.

The project name cannot have spaces or any of the following characters:

3 To accept all project defaults, click Finish.

We recommend that you accept the defaults for creating portlet applications. After you are skilled with portlet application development, you may find it appropriate to click Next and use the wizard to make modifications.

The webMethods CAF displays a Portlet Application Configuration Editor. The result of this procedure is a standard portlet application project, which is a deployable unit formatted as a WAR (Web ARchive) file.

Creating a Portlet on the Solutions View

Using the Solutions view, you can create a new portlet.



- On the Solutions view, right-click a portlet application and click New Portlet.
- 2 From the Project list, choose the portlet application to which the portlet should belong.

By default, the current portlet application is selected.

3 From the Portlet Type list of the Project Selection window, select the type of portlet you want to create:

Portlet type	Description
Generic	A generic portlet. Suitable for creating any portlet
Search Bar	A prebuilt template for use in developing a search bar portlet.
Search Results	A prebuilt template for use in developing a search results table portlet.
Task View (various)	A prebuilt template for use in developing task portlets.

If you selected the Search Results portlet type, from the Display Style list, choose one of the following:

Display Style	Description
Table View	Search results are displayed in a table.
Tree View	Search results are displayed in a tree structure.
Table/Tree View Toggle	The user can change search results to either type of display.

- 5 Click Next.
- 6 In the Portlet Name field of the Create Portlet window, type a portlet name.

The portlet name cannot have spaces or any of the following characters:

7 In Portlet Modes, select the modes you want to create for the portlet. Select or clear the check box for these modes:

Portlet Mode	Description
View	One or more pages the user can view, navigate, or interact with. View mode is required for each portlet. The default name is default.view. This is a required view file.
Edit	One or more pages the user can use to customize the behavior of the portlet. The default name is edit.view.
Help	Help information about the portlet. The default name is help.view.

For each mode you select, Designer creates a view file that you can develop.

8 In the Title field, type a title that identifies the portlet when you publish it on My webMethods Server.

If you do not provide a title, My webMethods Server creates a title by appending the portlet name to the portlet application name.

9 To accept the defaults, click Finish.

We recommend that you accept the defaults for creating portlets. After you are skilled with portlet application development, you may find it appropriate to click Next and use the wizard to make modifications.

The design canvas, a JSF graphical editor, is displayed.

Creating a Portlet View File on the Solutions View

Using the Solutions view, you can create a new view file.



- 1 On the Solutions view, right-click a portlet and click New View.
- 2 In File name field of the Portlet View File wizard, type the name of the new view file. You need to add a .view extension to the file name.
- 3 For the Portlet Mode option, choose the portlet mode:

Portlet Mode	Description
View	One or more pages the user can view, navigate, or interact with.
Edit	One or more pages the user can use to customize the behavior of the portlet.
Help	Help information about the portlet.

4 From the Template list, choose the initial content of the view file and click Next.

You have the following choices in the Template list:

Template	Purpose
Empty	The design canvas has no controls. To begin adding controls, you must first add a Form control to the canvas.
Form	The design canvas has a Form control, into which you can place other user interface controls.

Template	Purpose
Search Bar	The design canvas contains a default set of controls for creating a search bar.
Task Notification	The design canvas contains a default set of controls for creating a task notification view file.

5 On the Managed Bean Scope list, leave the default scope selected and click Finish.

Web Applications in the Solutions View

You can perform the following actions on portlet applications in the Solutions view:

- "Creating a Web Application on the Solutions View" on page 153
- "Creating a Web Application View File on the Solutions View" on page 154

Creating a Web Application on the Solutions View

Using the Solutions view, you can create a new Web application. Specific procedures may vary, depending on the type of application server you are creating the Web application for.



To create a Web application on the Solutions view

- On the Solutions view, right-click the User Interfaces node and click New Web Application.
- 2 Depending on the application server you are developing the Web application for, follow the steps in one of these procedures:
 - "Creating Web Applications for Tomcat" on page 26
 - "Creating Web Applications for JBoss" on page 27
 - "Creating Web Applications for Multiple Server Types" on page 29
 - "Creating Web Applications for My webMethods Server" on page 30

Creating a Web Application View File on the Solutions View

Using the Solutions view, you can create a new view file.



To create a new Web application view file on the Solutions view

- 1 On the Solutions view, expand the Web application to expose the WebContent node, right-click WebContent, and click New Web Application View.
 - You can create subfolders within WebContent to hold the view file (right-click WebContent and the click New > Folder) but do not place any files within the META-INF or WEB-INF nodes.
- 2 In the File Name field, type the name of the view file, which must have an extension of .view, and click Next.
- 3 On the Portlet View Options page, choose the Form template from the Template list.
- 4 Leave the Support Application Navigation option selected.
 - This option is required to make the view file able to participate in the Web application navigation model.
- 5 Click Next.
- 6 From the Managed Bean Scope list on the Managed Bean panel, choose the scope for the managed bean. The default is session.
 - For descriptions of JSF scopes, see "JSF Scopes" on page 83.
- 7 Click Finish.

The design canvas, a JSF graphical editor, is displayed.

Database Connectors in the Solutions View

After you have connected to a database using the Data Source Explorer view (see "Connecting to a Database" on page 113), you can create a new database connector or modify an existing one. From the Solutions view, you can also delete an unneeded Database connector (right-click the table and click Delete).

Before you can create a new Database Connector in the Solutions view, two conditions must be met:

- webMethods CAF needs to be connected to a database, as described in "Connecting to a Database" on page 113.
- You need to have an already created a Database Connector for the Web or portlet application.

To create a new database connector in the Solutions View, do the following.



To create a Database Connector in the Solutions view

- On the Solutions view, expand the Web application to expose the Database Connectors node, right-click Database Connectors, and click New Database Connector.
 - The Database Connection Configuration wizard appears, preconfigured with information about the database to which webMethods CAF is connected.
- 2 If needed, modify the values for the JDBC Driver, URL, Username, and Password fields, and click Next.
- 3 In Managed Bean Name field of the DB Client Properties panel, either accept the default name or type a new one.
- 4 To allow the database connector to write to the database, clear the Read-only client option, and click Next.
 - Connectors are read-only by default.
- 5 In the Table list, choose a database table and click **OK**.
- 6 On the Query Columns Selection page, choose the table columns you want to query.
- 7 (Optional) To add another table, click Add Table, choose a table, and click OK.

 For tables that have foreign key relations with the main table, the join condition is added automatically.
- 8 Click Finish.

Specifying an External Web Browser

By default, the live preview, My webMethods Server, and Integration Server display in an internal Web browser within Designer. You can, however, specify that an external Web browser be used.



To specify the use of an external Web browser

1 On the Designer main menu, Window > Web Browser.

The menu shows the Web browsers that are available on your machine. A check mark shows which browser will be opened when you display the Web browser.

2 To change the Web browser selection, click the appropriate menu item.

This menu item	Selects this
0 Internal Web Browser	The internal Web browser that will display within Designer. This is the default browser.
1 Default system Web browser	The external Web browser that is set as the default browser on your machine.
2 and above	Each type of Web browser that is installed on your machine.

3 Close the display and start it again to use the new display mode.

Chapter 7. Samples for the My webMethods Perspective

Importing Samples into the My webMethods Perspective	158
Common Sample Issues	159
Access Control Sample	160
Core Controls Sample	161
Core Provider Sample	161
File Export Sample	162
Locale Order Sample	162
Localization Sample	163
Northwind Sample	164
Portlet Links Sample	164
Search Sample	165
Wired Config Sample	165
Wizard Sample	166

Importing Samples into the My webMethods Perspective

There are several sample application projects you can import into the webMethods CAF to help you learn how to create various types of portlet applications. The samples are included in the SDK option of the My webMethods Server installation, which is not selected by default. If you are not sure whether the samples have been installed, look for this directory in the My webMethods Server installation directory:

webMethods install dir/MWS/samples

To add the samples, return to the webMethods Installer. In the component selection list, move to the appropriate location and choose to install this element:

XXXX 7.1.1 > CAF SDK 7.1.1

Assuming you have the samples installed, to import them, in the form of portlet applications, into the webMethods CAF, do the following:



- 1 In Designer: File > Import.
- In the Import wizard, expand webMethods, select Existing CAF Projects into Workspace, and click Next.
- 3 For the Select root directory field, click Browse and navigate to this location: webMethods_install_dir/MWS/samples
- 4 In the Projects list, select or clear portlet applications as appropriate and click Finish.
- In the Package Explorer view, Project Explorer view, or Navigator view you can do either or both of the following:
 - To display the Portlet Application Configuration Editor, expand the portlet application and double-click this node:
 - portlet_application/WebContent/WEB-INF/portlet.xml
 - To display the initial view for a portlet in the design canvas, expand the portlet application and double-click this node:

portlet_application/WebContent/portlet_name/default.view

Some portlet applications may have additional levels under WebContent, and some view files may not be named default.view.

Common Sample Issues

Read-Only Files

If you cannot modify or open a view or portal.xml file for edit, make sure none of the files in your samples are read-only. Some of these files must be opened in a edit or writable state. If the design canvas cannot open the file as writable, it displays a message to that effect.

Build errors

If the sample has build errors, make sure that you have set a JVM-JRE in your Designer preferences.

The JVM is unbound.

If the JVM is unbound, do the following:

- 1 On the Package Explorer view, right-click the sample and click **Properties**.
- 2 On the properties tree, click Java Build Path and click the Libraries tab to bring it to the front.
- 3 Select the JRE System Library and click Edit.
- 4 Select a JRE, click Finish, and click OK to apply the change to this project.

Missing libraries or Duplicate

If you have missing or duplicate libraries, do the following.

- 1 On the Package Explorer view, right-click the sample and click **Properties**.
- 2 On the properties tree, click Java Build Path and click the Libraries tab to bring it to the front.
- 3 Click Add Library and select Server Runtime as the type of library to add.
- 4 Select My webMethods Server and click Finish.

Now you may need to remove any duplicate jar files listed in the Libraries tab.

- 5 If you know which jar files are duplicates, select them and click Remove.
- 6 If you do not know which jar files are duplicates then expand the My webMethods Server node.
 - This will list all the jars the server runtime it has in its internal Libraries.
- 7 Select any duplicate jar files and click Remove.

Access Control Sample

Purpose

Demonstrates how to work with security roles to control access and visibility of the user interface and the content on a portlet. The sample contains a set of portlet views that change their text content, access to controls, and navigations dependent on the security role.

The default security role defaults to include the My webMethods Server system administrator. When logged in as the system administrator, you will see text for the administrator, and can control check boxes that configure what additional items a generic user can select. Submitting the form will navigate you to a view that displays all the items available to the generic user. As the administrator you do not get to select an item. As a generic user you will see text for the user and be able to select an item. Submitting the form will navigate you to a view that confirms your selection.

Location

webMethods_install_dir\MWS\samples\caf_controls\wm_access_control

Description

Adding an If and Else control block: Drop the logic If control on the canvas. Insert something into the control body. In this case it is a text output for the people in this role. For the If control's value property, bind the security role to it. If the user is in this role, the text will be displayed for them. Drop the Else control right after the If control. The Else control will key off the If control and require no binding. Just drop your content into the body of the control.

Adding an Access Control Panel: Drop the Access Control Panel control on the canvas. Insert something into the control body. In this case it is a text output for the people in this role. Set the Access Control Panel's property "Allowed Security Role". This hides it from those not in the security role and displays it to those in the role. For the next panel, the "Disallowed Security Role" property is used to hide the panel from those in the role and show it to those not in the role.

Using the security role to determine the navigation or other outcome from the command button: Add a Command Button control to the canvas. Create a method to bind to, and bind it to the control. Using the BasePortletBean method isUserInRole, add this to the command method to check if the current user is in the specified role. Return a string based on the method results. The navigation is controlled in the faces-config.xml file. For navigation examples, see "Wizard Sample" on page 166.

- "Importing Samples into the My webMethods Perspective" on page 158
- "Creating a Security Role" on page 94

- ▶ "Wizard Sample" on page 166
- ▶ "Common Sample Issues" on page 159

Core Controls Sample

Purpose

Demonstrates the core controls used to build portlet application. The controls are displayed in all of their common configurations and uses. All controls are interactive and include output results were applicable. Controls demonstrated are: Panels, table, input, output, list, logic, picker, script, search, table, toggle, util, and validator.

Location

webMethods_install_dir\MWS\samples\caf\wm_corecontrolstest

Related Topics

- "Importing Samples into the My webMethods Perspective" on page 158
- ▶ "Common Sample Issues" on page 159

Core Provider Sample

Purpose

Demonstrates the advanced use of the table controls along with the use of the table content provider. Includes table row deleting, updating, and sorting, and code samples for Pageable Table and Refreshable Content Providers.

Location

webMethods_install_dir\MWS\samples\caf\wm_coreproviderstest

- "Importing Samples into the My webMethods Perspective" on page 158
- ▶ "Common Sample Issues" on page 159

File Export Sample

Purpose

Demonstrates the exporting of the file formats: image, csv, xml, and text. The sample uses export methods by posting the action to the containing portlet and by posting the action directly to the .view servlet, bypassing the portlet.

Location

webMethods_install_dir\MWS\samples\caf\wm_caf_misc

Related Topics

- "Importing Samples into the My webMethods Perspective" on page 158
- "Common Sample Issues" on page 159

Locale Order Sample

Purpose

Demonstrates the use of the webMethods G11N API name and address object for displaying their content in the correct order based on the locale of the user. In Japanese, the last name is always displayed before or above the first name. In US English, the first name is always displayed before or above the last name. The same issues exist for addresses.

Location

webMethods_install_dir\MWS\samples\caf_localization\wm_localeOrder

Description

The Name view puts the input first and last name in the proper order for US English and Japanese. They are displayed as inline text and as output for controls ordered via IF ELSE control logic.

For the Address view, a preconfigured input is displayed on a single line of text using HTML breaks to make the address display as a column of address items. The address is also iterated into a ListTableContentProvider for display in a table.

- "Importing Samples into the My webMethods Perspective" on page 158
- "Common Sample Issues" on page 159

Localization Sample

Purpose

Demonstrates how to work with localizable resource bundles for your portlets and portlet applications. This sample shows how to access application resources and portlet resources, and how to load a resource bundle for JavaScript. The sample is a simple portlet that displays these resourced strings and uses a JavaScript alert to display a string from a loaded bundle.

Location

webMethods_install_dir\MWS\samples\caf_localization\wm_L10N

Description

Adding an Application resource: Click and drop a text output control on the canvas. From the Wm_L10NBean (application) node on the Bindings view, select applicationResources, title.1. Bind this resource to the control. The binding expression will look like, #{Wm_L10NBean.applicationResources["title.1"]}

Adding a Portlet resource: Drop a text output control on the canvas. From the Bindings view, select LocaleSampleViewPage (LocaleSampleViewPageBean) > localeSampleBean > portletResources > com.webmethods.localesample.title. Bind this resource to the control. The binding expression will look like,

 $\#\{Locale Sample View Page.locale Sample Bean.portlet Resouces ["com.webmethods.locales ample.title"] \}$

Adding a java script resource: Select a ".properties" file, copy and rename it to be the JavaScript resource bundle. Drop an Include Resource Bundle control on the canvas. For the Include Resource Bundle's value property enter the class path for the bundle. In this sample it is "com.webmethods.portal.localesample.JScriptResource". Drop a Script Block control on the canvas. For the Script Block's value property, add a script that will reference the include resource bundle. CAF.getBundle('from name')['message key']);

- "Importing Samples into the My webMethods Perspective" on page 158
- "Common Sample Issues" on page 159

Northwind Sample

Purpose

Demonstrates a custom database application of customer information. The intent of this sample is to show how to link a composite application to a database. The composite application uses a search bar and results page. You select a customer attribute to search on. The wild card is % and not *. The results are displayed in a table with paging and links to a separate page display of specific information about that customer. These customers can be edited along with the specific information on them. This sample requires an Integration Server and the ability to install a package on that server.

When running this sample, for best search results use all capital letters for your search terms. For example, use ANTO instead of anto.

Location

webMethods_install_dir\MWS\samples\caf\northwind

Related Topics

- "Importing Samples into the My webMethods Perspective" on page 158
- "Common Sample Issues" on page 159

Portlet Links Sample

Purpose

Demonstrates links in portlets to other views, data, or actions.

Location

webMethods_install_dir\MWS\samples\caf\wm_portlet_links

Description

Control Events view: Links to dynamically submit a value when you click a control.

JSR 168 view: Links to control JSR 168 functionality.

Multiple Portlets view: Click and update properties of the first portlet to see it automatically changed in the second portlet.

Multiple Views: Links to show other views from within the composite application

Order Search view: Search bar and Search Results example.

Portlet Include view: Folder view that includes the portlet wm_folderview, for My webMethods Server in this view.

Table view: A table links to display results in an associated properties group.

Related Topics

- "Importing Samples into the My webMethods Perspective" on page 158
- ▶ "Common Sample Issues" on page 159

Search Sample

Purpose

The sample shows how to use a search bar and search results to search a database. Selecting a name in the search results table will take you to the linked view of customer information data. The WSDL endpoint is stored in the wm_search_ws application environment setting. Edit the endpoint through the link to the composite application runtime configuration. Search for wm_search_ws and select Environment Entries. This sample requires an Integration Server and the ability to install a package on that server.

When running this sample, for best search results use all capital letters for your search terms and the wildcard value. For example, use ANTO* instead of anto.

Location

webMethods_install_dir\MWS\samples\caf\wm_search_ws

Related Topics

- "Importing Samples into the My webMethods Perspective" on page 158
- ▶ "Common Sample Issues" on page 159

Wired Config Sample

Purpose

This sample demonstrates how to work with portlet preferences and access these settings from other portlets using wiring. The portlets are distributed and wired using an xmlImport file. A third portlet is included to simulate Integration Server being installed and uninstalled. Submitted changes in the system settings are reflected in the results portlet controls wired to them. As you change the simulated installed servers, their system settings become enabled and disabled.

Location

webMethods_install_dir\MWS\samples\caf\wm_functional_privileges_test

Description

Adding preferences to the portlet: Using the Portlet Application Configuration editor you can add preferences to a portlet. Set the scope to be "session," which preserves the setting

for published instances of that portlet. Any portlet wired to this portlet will have access to any of these preferences. Once all the preferences are added, in the design canvas fill in the form with controls and a Command button to submit change to the preferences.

Adding a portlet to emulate an installed product: After adding a portlet and preferences for the install state of the servers, complete the form with controls and a Command button. The portlet's BaseFacesPreferencesBean is edited to add methods to get the server install state using the FabricSystem class. This class is for working with the Fabric products. Use the method productExists and the product constant to determine if that product is running on My webMethods Server. The PortalClient class is used to set the product constant.

Adding wiring to display your configuration settings: Using copy and paste, duplicate the configuration panel in a new portlet to display the results of the wiring. On the design canvas, add all portlets to the page. Set all the wiring. This layout will be used in the xmlImport file so you can deploy this page. Now export the page and use the xmlImport file inside of the exported file.

Related Topics

- "Importing Samples into the My webMethods Perspective" on page 158
- ▶ "Common Sample Issues" on page 159

Wizard Sample

Purpose

Demonstrates how to use navigation through views coupled with using a Submit Group control to create a wizard. The sample will show you a table of inventory items and an Add button. By pressing the Add button you can go through the wizard to input all the information for a new item in the table. The last wizard page shows all the content added and allows you to finish adding the item to the table or cancel the process.

Location

webMethods_install_dir\MWS\samples\caf_wizard\wm_wizard_sample

Description

Adding a table view: Add a table to a portlet. Add a managed bean to hold the data to be added to the table. Add a Command Button control and bind it to a method in the page view bean. From this method you will return a string that starts your navigation to another view. Values are cleared as the scope of the bean holding the data is session scope. The string returned by the method bound to the Command Button is checked against the navigations defined in the faces-config.xml file. If the string matches the rules, you are sent to WizPageOne.view.

Adding page one of the wizard: There are two actions available on this view, Cancel and Next. Cancel is a generic action that needs to be bound to a string. In this sample it is bound to the string "cancel". The navigation rule for cancel looks for a response of "cancel" from any view under the directory /cafWizardDemo.

The remaining navigation for an action of Next, Back, and Finish are only dependent on the page they come from, the method binding the response is sent from (optional), and the response.

The Next button: The navigation rules for page ID /cafWizardDemo/WizPageOne.view process any response from the method WizPageOneBean.updateSettings. If this method responds with "success" then the rule will navigate to the next view /cafWizardDemo/WizPageTwo.view.

Adding page two of the wizard. See "Adding page one of the wizard". The rule for this view, /cafWizardDemo/WizPageTwo.view, has the same layout as page one except for the page ID. If successful this view takes you to the view /cafWizardDemo/WizPageThree.view.

The Back button: The navigation rules for page ID /cafWizardDemo/WizPageOne.view process any response from the method WizPageTwoBean.updateSettings and take the response of "previous" from any bound method. A response of "previous" will cause you to navigate to the pervious view /cafWizardDemo/WizPageOne.view.

Adding page three of the wizard: The same as page Two. Notice the rule is the same except that for the word Two it has Three and you go to page four and back to page two.

Adding page four of the wizard: This view contains navigation that will Finish the wizard. This is an addition to the rule that looks for the response of "success" from page four. If successful you are taken to the view /cafWizardDemo/view.view, the view you started from. The Finished button is bound to a method on view.view's page bean. That method updates the table's data model and responds with "success".

Related Topics

"Importing Samples into the My webMethods Perspective" on page 158

Chapter 8. A Tutorial for Portlet Application Samples

Using the Portlet Tabs Tutorial	170
Portlet Tabs Tutorial	170

Using the Portlet Tabs Tutorial

The portlet application tutorials provide step-by-step descriptions of how certain sample portlet applications were developed. You can use the samples three ways:

How to use the samples

Import the sample into Designer to examine it.

Use Designer to build your own version.

Publish the sample (or your own version) to My webMethods Server to see h.ow it runs.

Where to learn more

"Importing Samples into the My webMethods Perspective" on page 158

"Portlet Tabs Tutorial" on page 170

"Publish and View Portlets on My webMethods Server" on page 175

Before you follow the tutorial, you should familiarize yourself with the design canvas, as described in "Using the Design Canvas" on page 97 and especially "Manipulating Controls on the Design Canvas" on page 100.

There are some basic tasks that are applicable to creating any portlet application:

- "Create a Portlet Application Project" on page 171
- "Create a Portlet" on page 171
- "Publish and View Portlets on My webMethods Server" on page 175

Portlet Tabs Tutorial

Sample Location

webMethods_install_dir/MWS/samples/caf_controls/wm_portlet_tabs

Description

This sample demonstrates how to create a portlet with multiple views, with tabs to navigate among those views.

Before you follow this tutorial, you should familiarize yourself with the design canvas, as described in "Using the Design Canvas" on page 97 and especially "Manipulating Controls on the Design Canvas" on page 100.

Follow the Tutorial

- 1 "Create a Portlet Application Project" on page 171
- 2 "Create a Portlet" on page 171
- 3 "Create a List of Views" on page 172

- 4 "Create the Tabs Control" on page 173
- 5 "Create Some Content" on page 174
- 6 "Create the Other Pages" on page 174
- 7 "Publish and View Portlets on My webMethods Server" on page 175

Create a Portlet Application Project

Procedure

1 In the My webMethods perspective: File > New > Portlet Application Project.

If you do not see this menu item, try New > Other.

2 In the Project Name field of the New Dynamic Web Project window, type a name for the portlet application.

The project name cannot have spaces or any of the following characters:

- 3 From the Target Runtime list, choose My webMethods Server.
- 4 From the Configurations list, choose CAF Portlet Application and click Next.
- 5 On the Web Module page, click Next.
- 6 On the JSF Capabilities page, choose the Server Supplied JSF Implementation option.
- 7 To accept all project defaults, click Finish.

Next Tutorial Step

"Create a Portlet" on page 171

Create a Portlet

Before you can begin this step, you must:

"Create a Portlet Application Project" on page 171

Procedure

1 In the Project Explorer view, open the portlet.xml file from this directory for the project. /

/project_name/WebContent/WEB-INF directory.

- 2 At the bottom of the portlet.xml editor, click the Configuration tab.
- 3 On the Configuration page, Select Portlets and click Add.

- 4 On the Project Selection page, click Next.
- 5 On the Create Portlet / Specify Portlet Information page, type a portlet name and click Next.
- 6 On the Create Portlet / Specify Deployment-Time Text page, click Next.
- 7 On the Create Portlet / Specify Portlet Class page, click Next.
- 8 On the Create Portlet / Specify Resource Bundle page, click Finish.

Next Tutorial Step

▶ "Create a List of Views" on page 172

Create a List of Views

Before you can begin this step, you must:

"Create a Portlet" on page 171

Procedure

- 1 In the Project Explorer view, open the default.view (project/WebContent/portlet/default.view).
- 2 In the Bindings view, locate the portlet node (the node with the name you created for *portlet*), right-click the node, and click Add > Data.
- 3 On the Add New Property page:
 - a In the Property Name field type Views.
 - b In the Property Type field, set java.util.ArrayList.
- 4 Double-click the new Views node you have just created.

The Java editor opens here:

```
public java.util.ArrayList getViews() {
   if (views == null) {
      views = new java.util.ArrayList();
   }
```

In the view array initialization block, add an entry in the array for each page that will be part of the tabs. Each entry itself consists of a string-array: the first index is the path to the page (starting at the WebContent directory); the second index is the page label. In this example, we have the default view, plus two pages that we will create later:

```
views.add(new String[] {"/Tabs/default.view", "Default"});
views.add(new String[] {"/Tabs/two.view", "Page Two"});
views.add(new String[] {"/Tabs/three.view", "Page Three"});
```

The completed entry looks like this:

```
public java.util.ArrayList getViews() {
    if (views == null) {
        views = new java.util.ArrayList();
        views.add(new String[] {"/Tabs/default.view", "Default"});
        views.add(new String[] {"/Tabs/two.view", "Page Two"});
        views.add(new String[] {"/Tabs/three.view", "Page Three"});
    }
    return views;
}
```

Next Tutorial Step

"Create the Tabs Control" on page 173

Create the Tabs Control

Previous Tutorial Step

"Create a List of Views" on page 172

Procedure

- 1 In the Package Explorer view, open the default.view (project/WebContent/portlet/default.view).
- 2 From the Palette view, drag a Tabs control (Control > List > Tabs) into the existing Form control in default view.
- 3 Select the new Tabs control, and in the Properties view, select the General tab; set the Rows property value to 0 (unlimited).
- 4 Select the Value tab, and bind the Value property value to the views list from the portlet bean (for example, "#{Tabs.views}").
 - (Click #{...} , locate the Views node you created in "Create a List of Views" on page 172, click the node, and click **OK**.)
- 5 Select the Display tab, and bind the Selected property value to:

```
#{row[0]==facesContext.viewRoot.viewId}
```

(Click $\#\{...\}$, type the binding expression in the Expression Binding field, and click OK.)

This expression will evaluate to true if the current row represents the current page (and so the tab will be displayed as "selected").

From the Palette view, drag a Portlet Simple Link control (Control > Portlet > Portlet Simple Link) into the Tabs control.

- From the Palette view, drag an Extended Portlet Url control (Control > Portlet > Extended Portlet Url) into the top yellow box of the Portlet Simple Link control (the Value container).
- 8 Select the new Extended Portlet Url control, and in the Properties view, select the Portlet URL tab; bind the Target View property value to #{row[0]} (the first index of the row entry the view page path).
- 9 From the Palette view, drag a Text control (Control > Output > Text) into the main pink box of the Portlet Simple Link control.
- 10 Select the new Text control, and in the Properties view, select the Value tab; bind the Value property value to #{row[1]} (the second index of the row entry the view page label).

Next Tutorial Step

"Create Some Content" on page 174

Create Some Content

Previous Tutorial Step

▶ "Create the Tabs Control" on page 173

Procedure

- From the Palette view, drag a Text control (Control > Output > Text) into the bottom of the form, below the Tabs control.
- 2 Select the new Text control, and in the Properties view, select the Value tab; bind the Value property value to This is Page One (#{facesContext.viewRoot.viewId})!.

Next Tutorial Step

▶ "Create the Other Pages" on page 174

Create the Other Pages

Previous Tutorial Step

"Create Some Content" on page 174

Procedure

- In the Project Explorer view, locate the default.view (project/WebContent/portlet/default.view), right-click the parent directory, and click New > Portlet View.
- 2 On the Portlet View File page, type the file name of the new view page (for example, "two.view"), and click Next.
- 3 On the Portlet View Options page, select the portlet for which to create the view (*portlet*), select the Form Template, and click Next.
- 4 On the Managed Bean page, click Next.
- 5 On the Java Type page, click Finish.
- 6 Repeat the steps in "Create the Tabs Control" on page 173 and "Create Some Content" on page 174 for this page (two.view).
- Repeat the steps in this topic, and the steps in "Create the Tabs Control" on page 173 and "Create Some Content" on page 174 for any additional pages (for the wm_portlet_tabs sample, we created one more page: three.view).

Next Tutorial Step

▶ "Publish and View Portlets on My webMethods Server" on page 175

Publish and View Portlets on My webMethods Server

To publish projects to My webMethods Server and view portlets, you need to do the following:

- "Create a My webMethods Server Reference" on page 175
- "Publish to My webMethods Server" on page 176
- "Create a Portlet Instance on My webMethods Server" on page 176

Create a My webMethods Server Reference

Create a reference to an instance of My webMethods Server to which you can publish portlets.

- In Designer:
 File > New > Other.
- 2 On the Select a Wizard page, choose Server > Server and click Next.
- 3 On the Define a New Server page, type the Server's host name.
 - a Select the server type of webMethods > My webMethods Server (Remote).
 - b Click Next.
- 4 On the New My webMethods Server page, type the server's Http Port and click Finish.

Publish to My webMethods Server

Perform steps 1 and 2 the first time you configure a project for publishing to My webMethods Server. Thereafter, you can skip to step 3 every time you want to republish an updated version of this project.

- 1 From the Servers view, right-click My webMethods Server (Remote) and click Add and Remove Projects.
- 2 In the Add and Remove Projects dialog, move the new project to the Configured projects list and click Finish.
- In the Servers view, if My webMethods Server shows a status of Stopped, select the server instance and, in the toolbar of the Servers view, click Start the server to begin communication with the server.



Note: Clicking **Start** the server cannot actually start an instance of My webMethods Server that is not already running.

- 4 Select My webMethods Server (Remote) and click Publish to the server.
- 5 If prompted for a user name and password, enter the credentials for a user who is a member of the Admin Role, such as sysadmin or designer.



Note: If the same instance of My webMethods Server is connected through the MWS Admin view ("Using the MWS Admin View" on page 125) and is logged in under different credentials, a publish action will try to log in using those credentials. If the owner of the connection in the MWS Admin view does have right to publish a portlet to the server the publish action will fail. To resolve this issue, right-click the server in the MWS Admin view, click Reconnect, and type credentials that have the right to publish. You can also delete that server from the MWS Admin view publish the portlet as described in this procedure.

Create a Portlet Instance on My webMethods Server

- 1 From the MWS Admin view, locate the My webMethods Server instance browse to the Public Folders folder
- 2 Right-click Public Folders and click New.
- 3 On the Create Data page, browse to the new portlet (portlet > project > portlet) and select it.
- 4 In the Name field, type a display name for the portlet.
- 5 Click Finish.

View the Portlet Instance

- 1 Open a Web browser, such as Firefox or Internet Explorer.
- In the browser's address bar, type the URL to the My webMethods Server instance, appending folder.public to that address (for example, http://localhost:8585/folder.public).
- 3 Enter the credentials of a MWS user at the login screen, such as sysadmin or designer.

Part II. User Interface Controls Reference

Controls Listed by Component Library	181
User Interface Controls Concepts	191
Behavior Controls	235
Command Controls	249
Dialog Controls	275
Input Controls	289
List Controls	397
Logic Controls	113
Map Controls	125
Output Controls	355
Panel Controls	149
Portlet Controls	179
Script Controls	189
Search Controls	503
Table Controls	525
Table Row Controls	585
Toggle Controls	511
Webapp Controls	525

webMethods Designer - Composite Application Framework Help Version 7.1.1

Chapter 9. Controls Listed by Component Library

The controls that are displayed on the Palette view ("Using the Palette View" on page 138) are made up of component libraries. Which component libraries you use may be dependent on whether you use the webMethods UI Developer or My webMethods perspective.

The webMethods UI Developer perspective uses the controls in these component libraries:

- ▶ "CAF JSF Base Runtime Control Library" on page 181
- ▶ "CAF Base Control Library" on page 182

The webMethods perspective uses the controls in these component libraries:

- "CAF JSF Base Runtime Control Library" on page 181
- ▶ "CAF Base Control Library" on page 182
- ▶ "CAF Portlet Control Library" on page 188
- ▶ "MWS Control Library" on page 189

CAF JSF Base Runtime Control Library

Use the controls in this library for:

- The webMethods UI Developer perspective
- The My webMethods perspective.

Palette	Control
Command	"Command Button" on page 261
Command	"Command Link" on page 265
Command	"Form" on page 268
Input	"Checkbox" on page 292
Input	"Checkbox Group" on page 294
Input	"Dropdown" on page 305
Input	"HTML Input" on page 318
Input	"Multi-Line Text Input" on page 322

Palette	Control
Input	"Option" on page 324
Input	"Option Group" on page 326
Input	"Radio Button" on page 328
Input	"Radio Button Group" on page 330
Input	"Secret Input" on page 333
Input	"Select-Many Listbox" on page 335
Input	"Select-One Listbox" on page 342
Input	"Text Input" on page 351
Logic	"Content Parameter" on page 416
Logic	"Control Parameter" on page 416
Logic	"Naming Container" on page 424
Output	"Control Label" on page 360
Output	"Image" on page 374
Output	"Link" on page 379
Output	"Message" on page 381
Output	"Messages" on page 383
Output	"Text" on page 392
Panel	"Grid Panel" on page 455
Panel	"Inline Panel" on page 462

CAF Base Control Library

Use the controls in this library for:

- The webMethods UI Developer perspective
- The My webMethods perspective.

Palette	Control
Behavior	"Custom Drag" on page 236
Behavior	"Custom Drop" on page 237
Behavior	"Drag To Move Rows" on page 239

Palette	Control
Behavior	"Drag To Reorder Columns" on page 243
Behavior	"Drag To Resize" on page 244
Behavior	"Drag To Resize Columns" on page 245
Behavior	"Raise On Change" on page 246
Behavior	"Synchronize Values" on page 247
Command	"Async Command Button" on page 250
Command	"Async Command Icon" on page 252
Command	"Async Command Interval" on page 255
Command	"Async Command Link" on page 256
Command	"Async Hidden Command" on page 259
Command	"Command Icon" on page 263
Command	"Hidden Command" on page 271
Command	"Servlet Command Form" on page 272
Dialog	"Modal Dialog" on page 276
Dialog	"Modeless Dialog" on page 278
Dialog	"Progress Dialog" on page 286
Input	"Autocomplete Text Input" on page 290
Input	"Combobox" on page 297
Input	"Date Input" on page 299
Input	"Date Range Input" on page 301
Input	"Extended Select-Many Listbox" on page 308
Input	"Extended Select-One Listbox" on page 310
Input	"File Input" on page 312
Input	"Filter Input" on page 314
Input	"HTML Input" on page 318
Input	"In Place Text Input" on page 320
Input	"Select-One Button" on page 338
Input	"Select-One Link" on page 340
Input	"Select-One Tabs" on page 345
Input	"Simple Schedule Input" on page 347

Palette	Control
Input	"Swapbox" on page 349
List	"Async List" on page 398
List	"Async Listbox" on page 401
List	"Async Tabs" on page 403
List	"Listbox" on page 406
List	"Simple List" on page 408
List	"Tabs" on page 411
Logic	"Async Iterator" on page 414
Logic	"Else" on page 417
Logic	"If" on page 418
Logic	"Import View" on page 418
Logic	"Iterator" on page 421
Logic	"Load Resource Bundle" on page 423
Мар	"Dynamic Map Marker List" on page 426
Мар	"Google Map" on page 431
Мар	"Google Map Key" on page 435
Мар	"Invoke Map Script" on page 435
Мар	"Map Marker" on page 438
Map	"Map Marker Group" on page 443
Map	"Map Navigation Control" on page 444
Мар	"Return Map Coords Script" on page 445
Мар	"Return Map Value Script" on page 446
Output	"Applet" on page 356
Output	"Button" on page 357
Output	"Custom Element" on page 362
Output	"Flash" on page 363
Output	"Formatted Message" on page 365
Output	"Formatted Messages" on page 366
Output	"Formatted Text" on page 366
Output	"Frame" on page 368

Palette	Control
Output	"Header" on page 369
Output	"Horizontal Rule" on page 370
Output	"Icon" on page 371
Output	"Include HTML" on page 375
Output	"Include Resource Bundle" on page 376
Output	"Include Script" on page 377
Output	"Include Stylesheet" on page 378
Output	"Refresh Button" on page 385
Output	"Refresh Icon" on page 386
Output	"Refresh Interval" on page 387
Output	"Refresh Link" on page 388
Output	"Script Block" on page 389
Output	"Select-One Output Text" on page 389
Output	"Specific Formatted Messages" on page 390
Output	"Specific Messages" on page 391
Output	"Truncated Text" on page 394
Panel	"Access Control Panel" on page 450
Panel	"Attachments Panel" on page 451
Panel	"Block Edge Panel" on page 452
Panel	"Block Panel" on page 453
Panel	"Disableable Panel" on page 454
Panel	"Hideable Panel" on page 457
Panel	"Inline Hideable Panel" on page 460
Panel	"Overlay Panel" on page 463
Panel	"Page Group" on page 465
Panel	"Popup Panel" on page 466
Panel	"Property Group" on page 467
Panel	"Property Line" on page 469
Panel	"Property Sub-Group" on page 469
Panel	"Scrolling Panel" on page 471

Palette	Control
Panel	"Stack Panel" on page 472
Panel	"Static Cell" on page 473
Panel	"Static Row" on page 474
Panel	"Submit Group" on page 475
Panel	"Titlebar Tabs Wrapper" on page 476
Panel	"Tooltip" on page 476
Script	"Control Reference" on page 490
Script	"Custom Script" on page 490
Script	"Invoke Script" on page 491
Script	"Parameter" on page 494
Script	"Return Value Script" on page 500
Table	"Async Categorized Table" on page 526
Table	"Async Table" on page 531
Table	"Async Tree" on page 536
Table	"Atom Feed Icon" on page 541
Table	"Calendar" on page 549
Table	"Categorized Table" on page 553
Table	"Column Sort Link" on page 557
Table	"Data Pages" on page 558
Table	"Data Prev/Next" on page 559
Table	"Data Total" on page 560
Table	"Data Total Selected" on page 562
Table	"Export Table Button" on page 562
Table	"Hidden Column" on page 565
Table	"Newsfeed" on page 565
Table	"Row Headers Column" on page 566
Table	"Scrollbar Column" on page 568
Table	"Standard Column" on page 568
Table	"Tree" on page 575
Table	"Tree Toggle" on page 580

Palette	Control
Table	"Truncating Column" on page 582
Table Row	"Add Row Button" on page 586
Table Row	"Add Row Icon" on page 587
Table Row	"Add Row Link" on page 589
Table Row	"Move Row Down Button" on page 590
Table Row	"Move Row Down Icon" on page 591
Table Row	"Move Row Down Link" on page 593
Table Row	"Move Row Up Button" on page 594
Table Row	"Move Row Up Icon" on page 595
Table Row	"Move Row Up Link" on page 596
Table Row	"Remove Row Button" on page 597
Table Row	"Remove Row Icon" on page 599
Table Row	"Remove Row Link" on page 600
Table Row	"Select All Rows Checkbox" on page 602
Table Row	"Select All Visible Rows Checkbox" on page 603
Table Row	"Select Row Checkbox" on page 604
Table Row	"Select Row Column" on page 605
Table Row	"Select Row Link" on page 606
Table Row	"Select Row On Click" on page 607
Table Row	"Select Row Quad-State Checkbox" on page 608
Table Row	"Select Row Tri-State Checkbox" on page 609
Toggle	"Initiate Toggle" on page 612
Toggle	"One Way Toggle Button" on page 612
Toggle	"One Way Toggle Checkbox" on page 614
Toggle	"One Way Toggle Link" on page 615
Toggle	"Toggle Button" on page 616
Toggle	"Toggle Dropdown" on page 618
Toggle	"Toggle Link" on page 620
Toggle	"Toggle Radio Button Group" on page 621
Toggle	"Toggle Tabs" on page 623

Palette	Control
Webapp	"Breadcrumbs" on page 626
Webapp	"Popup Menus" on page 627
Webapp	"Static Menus" on page 629
Webapp	"Toggle Menus" on page 631

CAF Portlet Control Library

Use the controls in this library for:

■ The My webMethods perspective.

Palette	Control
Portlet	"Portlet Parameter" on page 483
Portlet	"Portlet Simple Link" on page 484
Portlet	"Portlet URL" on page 487
Script	"Portlet Url Script" on page 496
Script	"Portlet Url Script Parameter" on page 498
Search	"Async Search Result Table" on page 504
Search	"Async Search Result Tree" on page 510
Search	"Search Bar" on page 514
Search	"Search Result Table" on page 517
Search	"Search Result Tree" on page 521

MWS Control Library

Use the controls in this library for:

■ The My webMethods perspective.

Palette	Control
Dialog	"Portal Resource Picker Dialog" on page 280
Dialog	"Principal Picker Dialog" on page 283
Portlet	"Extended Portlet Parameter" on page 480
Portlet	"Extended Portlet URL" on page 481
Portlet	"Portlet Include" on page 482

webMethods Designer - Composite Application Framework Help Version 7.1.1

Chapter 10. User Interface Controls Concepts

Control ID Reference
Hideable Controls
Toggle Controls
Scriptaculous Effects
Client-Side Model
Image URLs
Table Row Tools
Client-Side Libraries
Filterable Controls

Control ID Reference

Every JSF control can be assigned an ID. This is useful because many controls need to reference other controls in the same view. Common examples of this are the "for" attribute of a "One Way Toggle Button" on page 612, and the select item (option) values of a "Toggle Button" on page 616, both of which take a control ID reference to specify the control or controls to toggle.

A control's ID must be unique within the control's naming container. Naming containers implement the JSF javax.faces.component.NamingContainer interface. The following controls are naming containers:

- "Form" on page 268
- "Table" on page 570
- "Iterator" on page 421
- "Simple List" on page 408
- "Tabs" on page 411
- "Import View" on page 418
- "Tree" on page 575

Absolute References

Because control IDs do not need to be unique outside of naming containers, a single view can contain several different controls with the same ID. To disambiguate between these controls, control ID references may include the control's naming container ancestor IDs in the reference. For example, given a form control with an ID of "myForm" and a hideable panel control within that form that has an ID of "myPanel," the "absolute" control ID reference to the panel is ":myForm:myPanel." An absolute control ID reference starts with a leading colon (:), and unambigously identifies a control from any other control on the page.

Relative References

Control ID references also can be "relative." A reference is relative if only part of the path is specified; a relative reference can simply be the control's ID. For example, to reference the above panel control from a specific control—say, a button below the form—we can specify the ID of the panel ("myPanel"). When a relative reference consists of only a control ID, the reference is resolved by finding the "nearest" control to the referencing control with the specified ID.

If, however, some other control with the same ID is nearer to the referencing control, we need to qualify the reference with part of the absolute reference. Extending the above example, say there is a panel sibling to the button, also with an ID of "myPanel". To reference the "myPanel" control that is in the form from the button, we have to qualify the reference as "myForm:myPanel". Relative control ID references never start with a leading

colon (:), and depend on the location of the referencing control to correctly identify the referenced control.

Absolute Versus Relative

Note that in the last example above the relative control ID reference ("myForm:myPanel") is similar to the absolute control ID reference (":myForm:myPanel"); however, using the relative version has an advantage. If we add some other naming container around both the form and the button (say by importing this view into some other view by means of an ImportView control), we can still use the same relative control ID reference to reference the panel from the button; if we used an absolute control ID reference, we would now have to prepend to it the naming container's absolute control ID reference. For example, if the ImportView's control ID was "myImportView" (with an absolute control ID reference ":myImportView"), the panel's absolute control ID reference would now become ":myImportView:myForm:myPanel."

External Portlet References

Some controls can accept a reference to another control outside of the current view, in an external portlet on the same portal page. An external portlet reference begins with the URI of, or an alias to, the portlet, followed by a pound sign (#), followed by the absolute control ID reference (sans leading colon). For example, say the "myPanel" control from the first example is part of a portlet with an alias of "myPortlet." The "myPanel" control can be referenced by other portlets on the same page as "myPortlet#myForm:myPanel."

Literal HTML References

Some controls can accept a reference to a raw HTML element directly, in place of a conventional control ID reference. To reference a raw HTML element, specify the element's ID prefixed with a dollar sign (\$). For example, "\$myDiv" references a div with an ID of "myDiv". You should not use a literal HTML reference when you can use a regular control ID reference instead.

Nearest Control Algorithm

The "nearest" control in a relative control ID reference actually means the nearest control in the nearest subtree to the referencing (or "source") control. The exact algorithm for finding the "nearest" control is the following:

- 1 Test the source control.
- 2 Search the descendants of the source control, depth-first.
- 3 Search the siblings of the source control.
- 4 Search the descendants of the siblings of the source control, depth-first.
- 5 Test the parent of the source control.
- 6 Search the siblings of the parent of the source control.
- 7 Search the descendants of the siblings of the parent of the source control, depth-first.

- 8 Test the grandparent of the source control.
- 9 Search the siblings of the grandparent of the source control.
- 10 Search the descendants of the siblings of the grandparent of the source control, depthfirst.

Hideable Controls

Hideable controls can be toggled between visible and hidden through client-side JavaScript code. Special disable When Hidden controls encapsulate this code within controls that can be created and configured visually, using the webMethods CAF. See "Toggle Controls" on page 197 documentation for more information about toggle controls.

Server-Side Properties

Hideable controls have a server-side "visible" (boolean) property, which determines whether or not the control is rendered as initially visible. They also have a "disableWhenHidden" (boolean) property, which disables all contained controls when the hideable control is not visible. Disabled controls' values are not submitted to the server. Setting the "defaultFocus" ("Control ID Reference" on page 192) property of most hideable controls will focus (highlight) the specified control when the hideable control is made visible.

Many hideable controls also have "hideEffect," "hideEffectOptions," "showEffect," and "showEffectOptions" expert properties, which allow a developer to fine-tune the control's hide and show behavior. The "hideEffect" and "showEffect" properties name the Scriptaculous Effect ("Scriptaculous Effects" on page 198) to use when hiding or showing the control, and the "hideEffectOptions" and "showEffectOptions" enumerate any custom options to use with the hide or show effects. By default, no effect is used when hiding or showing a control.

Client-Side Functionality

On the client side, the control's visibility can be toggled using the control's CAF client-side model. Changing the client-side visibility of a control modifies its server-side "visible" property.

The following methods on the client-side model affect the control's client-side visibility:

- isVisible(): returns true if the control is visible
- setVisible(visible): sets the control as visible/hidden
- show(): makes the control visible if hidden
- hide(): makes the control hidden if visible
- toggle(): makes the control visible if hidden, or hidden if visible

For example, setting the value of a checkbox control's "onclick" property to the following would result in toggling a separate "myHideableControlId" hideable control between visible and hidden whenever the checkbox is clicked:

```
CAF.model('#{myPageBean.clientIds'myHideableControlId'}').toggle();
```

See "Client-Side Model" on page 201 for more information about the webMethods client-side model.

Lazy Loading

Many hideable controls have lazy-loading capabilities, meaning that the control's content can be loaded asynchronously, using a secondary (hidden, or "Ajax" on page 83) request. The panel's "lazy" property controls this capability; setting "lazy" to true enables it, while false (the default) disables lazy-loading.

refreshOnShow Property

Several common properties modify a hideable control's lazy-loading capabilities. They have no effect unless the panel's "lazy" property is true. The "refreshOnShow" property controls whether or not the control's content is refreshed every time the control is toggled visible; setting "refreshOnShow" to true will make the control re-retrieve its content via a secondary request every time the control is toggled visible, while false (the default) makes it so that once the control's content has been loaded, it will not be refreshed if the control is toggled hidden and then visible again.

twoPass Property

The "twoPass" property forces a hideable control to retrieve its content via a secondary request, even when initially visible. Setting the "twoPass" property to true prevents the control's content from rendering with the initial page; it is instead retrieved by a secondary request once the page is loaded by the browser. Setting "twoPass" to false makes the control's content render synchronously with the rest of the page, if the control is initially visible. If the control is not initially visible, the "twoPass" property has no effect.

supressInputs Property

When a secondary request is made to the server to retrieve the content of a lazy-loaded control — and the control is part of a form — the state of all the form's controls are posted up to the server, the controls are validated, and the control values are used to update the data models to which they are bound. Of course, this behavior is not desirable in all cases. You can use the "supressInputs" property to specify a comma-separated list of IDs of the controls you do not want validated or updated. The descendants of any control in the list of suppressed inputs is also suppressed; for example, if you want to suppress the validation and update processing of all the controls in a property-group control, you can simply specify the ID of the property-group instead of the ID of each individual control in the group. However, any direct ancestor of the hideable control in the list of suppressed inputs is ignored so you cannot add the ID of the form to the suppressed list.

Progress Bar Customization

A progress bar automatically appears when a hideable control's content is loaded asynchronously. The progress bar's display can be configured with the following hideable control properties:

- progressDelay: Milliseconds to delay before showing progress bar (defaults to 0). A -1 (negative one) suppresses the progress bar completely.
- progressFlashOnComplete: True to flash control when control's content finishes loading (defaults to true).
- progressMsg: Progress bar message (defaults to localized value; in English, "Loading...").
- progressUseHideShowEffect: Use effects specified by "hideEffect", "hideEffectOptions", "showEffect", and "showEffectOptions" properties to hide and show progress bar (defaults to false).

Hideable Control Instances

The server-side component object model of controls that are "hideable" implement the com.webmethods.caf.faces.component.IHideablePanel interface. These controls include the following:

- "Hideable Panel" on page 457
- "Inline Hideable Panel" on page 460
- "Modal Dialog" on page 276
- "Modeless Dialog" on page 278
- "Overlay Panel" on page 463
- "Progress Dialog" on page 286

Related Topics

- "Control ID Reference" on page 192
- ▶ "Toggle Controls" on page 197
- "Scriptaculous Effects" on page 198
- "Client-Side Model" on page 201

Toggle Controls

Toggle controls adjust the client-side visibility of a group of "hideable" controls, or the server-side "rendered" property of a group of "non-hideable" controls. Only the selected control from this group (specified by the "value" property of the toggle control) is visible. Selection options are specified via the toggle control's javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children, with each value specifying a toggleable control ID. Selecting an option whose value does not specify a control ID hides all controls in the group.

For example, if the toggle control has three options, "control-one," "control-two," and "control-three," and the toggle control's "value" property is empty, then "control-one," "control-two," and "control-three" are all hidden. If the toggle's control "value" property is "control-one," "control-one" will be visible, and "control-two" and "control-three" will be hidden.

A toggle control can also be used to toggle a single control between visible and hidden. For example, if the toggle control has two options, "control-one" and "", and the toggle control's "value" property is empty, then "control-one" will be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible.

Behavior With Hideable Controls

For hideable controls, the toggle control modifies the hideable control's client-side visibility without requiring a page refresh or contacting the server at all. See "Hideable Controls" on page 194 for more information about hideable controls.

Toggle control options may reference external hideable controls via external portlet references, or raw HTML elements via literal HTML references. See "Control ID Reference" on page 192 to construct such references.

Behavior With Non-Hideable Controls

For non-hideable controls, instead of modifying the control's client-side visibility, the toggle control toggles the non-hideable control's server-side "rendered" property, refreshing the page to display the updated state (visible or hidden) of the non-hideable control. To do this properly, the toggle control must be contained within a "Form" on page 268 control. The current state of all the controls within the form will be transferred to the refreshed page.

Control Source Order

When a toggle control is positioned after (top-to-bottom, left-to-right) any of the controls it toggles, you must add an "Initiate Toggle" on page 612 control to the view before the first of those controls, at the same level as the toggle control. For example, if the toggle control is in a table column, the Initiate Toggle control should be the first control in the

first column of the table. The Initiate Toggle control's "for" property must be set to reference the toggle control.



Note: An Initiate Toggle control is needed only when a toggle control comes after one of the controls it toggles, which, in practice, happens rarely.

Toggle Control Instances

The server-side component object model of "toggle" controls generally extend the com.webmethods.caf.faces.component.toggle.Toggle class. These controls include the following:

- "Toggle Button" on page 616
- "Toggle Link" on page 620
- "Toggle Dropdown" on page 618
- "Toggle Radio Button Group" on page 621
- "Toggle Tabs" on page 623

Related Topics

- ▶ "Control ID Reference" on page 192
- ▶ "Hideable Controls" on page 194

Scriptaculous Effects

The webMethods CAF uses the <u>Scriptaculous</u> library for client-side effects (for example, the Tooltip control's fade/appear effect or the ModalDialog control's grow/shrink effect). Some controls (in particular, HideableControls) allow the developer to configure the control's hide and show effects via the "hideEffect" and "showEffect" expert properties. You can specify any standard Scriptaculous effect name (for example, Effect.Fade), or any custom composite application effect name (for example, Effect.CAF.SlideOpen), for these values.

You can specify additional options for these effects via the "hideEffectOptions" and "showEffectOptions" expert properties. Options are specified in JavaScript Object Notation (JSON); for example:

direction: "bottom", duration: 0.5, transition: Effect.Transitions.wobble

Many effects work fine with their defaults, and do not require you to specify any options.

Standard Scriptaculous Effects

Show

- Effect.Appear: Fade in.
- Effect.BlindDown: Reveals from top to bottom.
- Effect.SlideDown: Slide open, downward.
- Effect.Grow: Grow from center.

Hide

- Effect.Fade: Fade out.
- Effect.BlindUp: Cover from bottom to top.
- Effect.SlideUp: Slide close, upward.
- Effect.Shrink: Shrink to center.
- Effect.Puff: Expand and fade out.
- Effect.SwitchOff: Flicker and drop out.
- Effect.DropOut: Move down and fade out.
- Effect.Squish: Shrink to corner.
- Effect.Fold: Blind up, then shrink to corner.

Highlight

- Effect.Shake: Shake back and forth.
- Effect.Pulsate: Fade in, fade out, fade in, fade out.
- Effect.Highlight: Fade in yellow background, fade back to white background.

Custom CAF Effects

Show

- Effect.CAF.Appear: Extends Effect.Appear with a default duration of 0.3.
- Effect.CAF.SlideOpen: Slide open in any direction; options:
 - *direction*: Direction to slide: "left", "top", "right", or "bottom" (default).
 - *duration*: Duration (in seconds) of slide (defaults to 0.3).
- Effect.CAF.SlideOpen.Left: Slide open leftward.
- Effect.CAF.SlideOpen.Top: Slide open upward.
- Effect.CAF.SlideOpen.Right: Slide open rightward.
- Effect.CAF.SlideOpen.Bottom: Slide open downward.

Hide

- Effect.CAF.Fade: Extends Effect.Fade with a default duration of 0.3.
- Effect.CAF.SlideClose: Slide close in any direction; options:
 - direction: Direction from which to slide: "left", "top", "right", or "bottom" (default).
 - *duration*: Duration (in seconds) of slide (defaults to 0.3).
- Effect.CAF.SlideClose.Left: Slide close from the left.
- Effect.CAF.SlideClose.Top: Slide close from the top.
- Effect.CAF.SlideClose.Right: Slide close from the right.
- Effect.CAF.SlideClose.Bottom: Slide close from the bottom.

Highlight

- Effect.CAF.Shake: Extends Effect.Shake with additional options:
 - *x*: Amount (in pixels) to shake horizontally (defaults to 20).
 - y: Amount in (pixels) to shake vertically (default to 0).
 - *duration*: Duration (in seconds) of a single shake (defaults to 0.05).
 - count: Number of shakes (defaults to 5).
- Effect.CAF.Highlight: Fade in a halo, fade back to normal; options:
 - duration: Duration (in seconds) of effect (defaults to 1.0).
- Effect.CAF.Flicker: Fade to white, fade back; options:
 - duration: Duration (in seconds) of flicker (defaults to 1.0).
 - *to*: Degree to which to fade, from 1.0 to 0.0, where 1.0 is a complete fade and 0.0 is no fade (defaults to 0.7).
- Effect.CAF.Flash: Combination flicker and highlight; options:
 - *duration*: Duration (in seconds) of effect (defaults to 1.0).

Common Effect Options

- duration: Duration (in seconds) of effect (defaults to 1.0).
- transition: Transition function to use for effect. This function is passed a number between 0.0 and 1.0, and returns a number between 0.0 and 1.0 (defaults to Effect.Transitions.sinoidal).
- from: Transition start (defaults to 0.0).
- *to*: Transition end (defaults to 1.0).
- **q** *queue*: Advanced queue options.
- direction: Transition direction (a string like "top"). Used only by directional effects.

Effect Transitions

A transition function can be specified for most effects via the effect's transition option. This function can modify the behavior of the effect by making the effect appear to speed up or slow down as it progresses, or even oscillate between the start state and the end state. The transition function is passed a number between 0.0 and 1.0, and it returns a number between 0.0 and 1.0. The default function is Effect.Transitions.sinoidal, which causes the effect to accelerate in the middle and slow as it ends.

Standard Scriptaculous Transitions

- Effect.Transitions.linear: Linear transition from 0 to 1.
- Effect.Transitions.sinoidal: Sine wave transition (accelerates in the middle, slows at the end) from 0 to 1.
- Effect.Transitions.reverse: Linear transition from 1 to 0.
- Effect.Transitions.flicker: Unstable transition between 0.5 and 1.
- Effect.Transitions.wobble: Oscillates between 0 and 1, converging on 0.5.
- Effect.Transitions.pulse: Five linear oscillations between 0 and 1.
- Effect.Transitions.none: Always returns 0.
- Effect.Transitions.full: Always returns 1.

Custom CAF Transitions

- Effect.Transitions.CAF.pulse1: Smooth single pulse from 0 to 1 and back to 0.
- Effect.Transitions.CAF.flash1: Pulses from 0 to 1 quickly, then from 1 to 0 more slowly.

Client-Side Model

JavaServer Faces provides a convenient abstract model by which to interact programmatically with a UI on the server. No similar standard abstract model exists on the client. Software AG, however, does provide a proprietary client-side model that mirrors, where possible, the JSF server-side model.

Related Topics

- ▶ "CAF.Model" on page 202
- ▶ "CAF.Output.Model" on page 202
- "CAF.Link.Model" on page 203
- ▶ "CAF.Command.Model" on page 203
- "CAF.Input.Model" on page 204

- ▶ "CAF.Checkbox.Model" on page 205
- ▶ "CAF.Select.Model" on page 205
- ▶ "CAF.Table.Model" on page 207
- "CAF.Tree.Model" on page 209
- ▶ "Advanced Considerations" on page 211

CAF.Model

The client-side CAF.model(id) function can create a model object for each control rendered on a page. All control IDs used by the CAF client-side model are client-side IDs, which are different from, but related to, the server-side IDs that a developer specifies at design-time. Client-side IDs are generated at run-time, but can be mapped from server-side IDs in one of two ways. In Java code, calling the getClientId() method of a javax.faces.component.UIComponent object gets the client-side ID of the control represented by the UIComponent object. In a binding expression, referencing the clientIds content-provider on any page bean (#{activePageBean.clientIds['myControlId']}) produces a client-side ID for any given server-side ID. See CommandIDReferences for more information about server-side IDs.

The client-side model objects (models) produced by the CAF.model() function allow a developer to manipulate controls in an abstract way, without knowledge of the controls' concrete DHTML implementation. The model itself holds no state; every invocation of CAF.model() creates a new model instance. The model simply exposes methods by which the state of the modeled control can be examined and modified.

All models generally have three standard, read-only properties: ID, element, and form. ID is the control's client-side ID (ID), element is the control's primary HTML element (if it exists), and form is the control's containing HTML form element (if the control is contained by a form).

CAF.Output.Model

The base model class is CAF.Output.Model. Almost all control models are or extend this class, including all "UIOutput" controls (like Text) and all "UIPanel" controls (like BlockPanel).

The value of a CAF.Output.Model can be examined and modified via the getValue() and setValue(value) methods. This is usually just the control element's innerHTML property. If the control is a javax.faces.component.ValueHolder, this value will likely correspond to the control's "value" property. For example, the value of an Image control is the Image's URL; given an Image control with an ID of ID, the following line of JavaScript would dynamically set the Image to display a Google logo:

CAF.model(id).setValue("http://www.google.com/logos/Logo_50wht.gif");

The disabled, focused, and visible states of a CAF.Output.Model can be examined and modified via the isDisabled()/setDisabled(disabled), isFocused()/setFocused(focused), and isVisible()/setVisible(visible) methods. Often setting the disabled and focused states of an UIOutput or UIPanel control will not affect the control itself, but may affect descendant controls. The show() and hide() methods also modify the control's visibility, as does the toggle() method, which shows the control if it is hidden, or hides the control if it is visible.

The contents of any control can be refreshed from the server via an asynchronous request by calling the model's refresh() method. For example, let's refresh a panel with a server-side ID of "myPanelId" when the user double-clicks on it. We'd add a Script Block control to the page, and in its "value" property we'd use the Prototype library's Event.observe function to register a double-click event handler, and call the refresh() method in that handler's body:

```
Event.observe('#{activePageBean.clientIds['myPanelId']}', 'dblclick', function(event) {
    CAF.model('#{activePageBean.clientIds['myPanelId']}').refresh();
});
```

The toggle() and refresh() behaviors of "hideable" controls can be fine-tuned by configuring the control's server-side properties. See HideableControls for details. See the CAF.Output.Model Javadocs for more information about the CAF.Output.Model.

CAF.Link.Model

Link controls extend the CAF.Output.Model with the CAF.Link.Model. The value of the CAF.Link.Model is the link's URL. The CAF.Link.Model has one additional method: raise(). This method actuates the link (simulates a click on the link).

CAF.Command.Model

"UICommand" controls extend the CAF.Link.Model with the CAF.Command.Model. A CAF.Command.Model's value is the control's ID. The raise() method is also available on the CAF.Command.Model; it actuates the command (simulates a click on the command button or link). The CAF.Command.Model also has a go(params) method, which allows a developer to invoke the command directly (without simulating a click on the command button or link) with the specified request parameters. For example, if we created a HiddenCommand with an server-side ID of 'myCommandId', and the created a simple output Button and set its "onclick" property to the following, when the user clicks the Button it will submit the containing form using the HiddenCommand, and include "myParam1" and "myParam2" as request parameters:

```
CAF.model('#{activePageBean.clientIds['myCommandId']}').go({
   myParam1: "My Value One",
   myParam2: "My Second Value"
});
```

The CAF.Command.Model also allows a developer to register client-side action-listeners, which are invoked when the command in initiated, after the form has been validated but before it is submitted. A client-side action-listener is a function that accepts one argument, the client-side ID of the command control. If the action-listener returns false, the command aborts and the form is not submitted. For example, the following JavaScript code adds an action-listener to the command with a server-side ID of "myCommandId", and aborts the processing of the command until the command has been invoked five times:

```
var g_myAnnoyingCount = 0;
CAF.model('#{activePageBean.clientIds['myCommandId']}').addActionListener(function(id
) {
   return (g_myAnnoyingCount++ > 5);
});
```

CAF.Input.Model

"UIInput" controls extend the CAF.Output.Model with the CAF.Input.Model. In addition to the functionality of the CAF.Output.Model, the CAF.Input.Model allows client-side value-change listeners and validators to be registered for a given control. A client-side value-change listener is a function that takes three arguments: the control's client-side ID, the control's old value, and the control's new value. Value-change listeners are called whenever the control's value changes, after the value has already changed.

For example, say we created a Dropdown control, where each option value was the client-side ID of a HideablePanel control. Only the HideablePanel that is currently selected by the Dropdown should be shown. We could add the following value-change listener to the Dropdown that, when the user changes the Dropdown selection state, hides the previously selected HideablePanel, and shows the newly selected HideablePanel (note this is what the ToggleDropdown control does automatically):

A client-side validator is a function that takes two arguments: the control's client-side ID, and the control's value. It returns an empty string if the control is valid, and a non-empty, error-message string if the control's value is invalid. For example, the following JavaScript registers a validator which validates that the input's value does not contain white space:

```
CAF.model('#{activePageBean.clientIds['myInputId']}').addValidator(function(id, value) {
    return (/\s/.test(value) ? "Cannot contain whitespace." : "");
});
```

CAF.Checkbox.Model

Often checkboxes are used as part of a larger checkbox group, which is modeled by the CAF.Select.Model (see the next section). However, the individual checkbox control ("UISelectBoolean") is modeled by the CAF.Checkbox.Model (which extends the CAF.Input.Model). With the checkbox model, getValue() returns the string "true" if the checkbox is checked, and an empty string ("") if the checkbox is not checked. Developers can take advantage of JavaScript's notion of truthy and falsy, where "true" equals boolean true and "" equals boolean false.

For example, say we have a Checkbox control with an ID of 'myCheckboxId' and next to a TextInput control with an ID of 'myTextInputId'. We want to disable and clear the text input if the checkbox is toggled unchecked, and enable the text input if the checkbox is toggled checked. We could apply the following value-change listener to accomplish this:

Developers can also use the getLabel() and setLabel(label) methods to dynamically get and set the checkbox's label.

CAF.Select.Model

"UISelectOne" and "UISelectMany" controls extend the CAF.Input.Model with the CAF.Select.Model. The value of a select-one control (such as a dropdown and radio-button group) is the (string) value of the selected item, or an empty string ("") if no item is selected. The value of a select-many control (such as a listbox or a checkbox group) is the array of selected item values, or an empty array if no items are selected.

Individual select items are modeled by CAF.Select.Item.Model, which has value, label, description, disabled, style (inline CSS style), styleClass (CSS class name), and icon (icon URL) properties. Setting these properties doesn't affect the control directly; the set(x, item) method on the select control model must be called with the updated select item in order to update the control. The first argument (x) to this method can either be the (zero-based) index of the item, or the string value of the item.

Symmetric to the set(x, item) method, the get(x) method gets the CAF.Select.Item.Model at either the specified (zero-based) index or with the specified string value. The full list of CAF.Select.Item.Model items in the control can be acquired via the list() method; the selected() method returns an array of just the selected CAF.Select.Item.Model items.

Items can be added to the control via the add(x, item) method; the first argument to this method (x) can be either the index, string value, or CAF.Select.Item.Model of the existing item before which to insert the new item (or can be omitted entirely to append the new item as the last item in the control). Items can be removed via the remove(x) method;

again the first argument to this method (x) can either be the index, string value or CAF.Select.Item.Model of the item to remove.

The following example appends "Item Ten" to a dropdown (and then selects it). The first argument specified to the CAF.Select.Item.Model's constructor is the value of the new item; the second argument is its label. When reading the Javadocs for CAF JavaScript objects, note that the constructor for the object is the method named initialize() (which is the convention defined by the Prototype library).

```
dropdown.add(new CAF.Select.Item.Model("value-ten", "Item Ten"));
dropdown.setSelected("value-ten", true);
```

Items can be selected via the setSelected(x, selected) method; the first argument to this method (x) can be either the index, string value, or CAF.Select.Item.Model of the item to select. Selection state of an item can be tested via the isSelected(x) method; the first argument to this method (x) can be either the index, string value, or CAF.Select.Item.Model of the item to test.

Like all CAF.Input.Model models, the CAF.Select.Model model supports value-change listeners (which are notified every time the control's selection state changes); in addition, it allows for selection-change listeners. Selection-change listeners are notified only when a change occurs to selection state of the particular item for which they were registered. The listener receives three arguments: The control ID, the item value for which the listener was registered, and the new selection state (true for selected, false for unselected).

The addSelectionChangeListener(fn, value) allows for selection-change listeners to be registered. The first argument to this method (fn) specifies the listener function, and the second argument (value) specifies the value of the item for which to listen. The following example adds a selection-change listener to the "myDropdownId" control. If the "value-one" item is selected, the listener shows "myPanelOneId"; if its not selected, it hides "myPanelOneId":

CAF.Table.Model

"UIData" controls extend the CAF.Select.Model with the CAF.Table.Model. Rows in a table model can be manipulated (added, removed, updated, selected, etc.) just like items in a select model.

Related Topics

- ▶ "CAF.Table.Row.Model" on page 207
- ▶ "Template Row" on page 208
- ▶ "Paging" on page 208
- ▶ "Listeners" on page 209

CAF.Table.Row.Model

The CAF.Table.Row.Model, which models a row in a table, extends the CAF.Select.Item.Model (which models a particular item in a select control). The value property of a table row is its server-side row ID (the ID specified for each row by an IAddressableTableContentProvider). The label property of a table row is a CSV string of the row's HTML column content. The description, disabled, style, and styleClass properties all apply the same to a table row as they do to a select item. The icon property does not apply to a table row.

The CAF.Table.Row.Model also has a values property, which replaces the label property as the second argument to its constructor. The values property is a map; it maps the server-side IDs of the controls contained within the row to the values of those controls. So if a table contained a dropdown with an ID of "myDropdownId" and a output text control with an ID of "myOutputId", the following example would append a new row to the table with a row-ID of "row-ten", and set the row's dropdown to the "value-ten" item and the row's output text to display "Row Ten" (assuming the table had a template row, described by the next section):

table.add(new CAF.Table.Row.Model("row-ten", {myDropdownId: "value-ten", myOutputId: "Row Ten"}));

A CAF.Table.Row.Model also has a few extra methods designed to help manipulate the row's contents. The getControlId(localId) method gets the ID of any control contained by the row, given its server-side ID. For example, say we have a table with an ID of "myTableId", and one of the columns within the table has a dropdown with an ID of "myDropdownId". We can get a reference to the control in a specific table row (in this case, with a row-ID of "row-one"), by creating a model for the row ("row-one"), and calling its getControlId() method with the dropdown's server-side ID:

```
var table = CAF.model("#{activePageBean.clientIds['myTableId']}");
var row = table.get("row-one");
var dropdownId = row.getControlId("myDropdownId");
var dropdown = CAF.model(dropdownId);
```

Template Row

If the table is bound to a content-provider that implements the IUpdateableTableContentProvider interface, and the content-provider returns true for the interface's getCanTemplateRow() method, then the table will automatically render a hidden template row. If a row is added to the table using the CAF model, and the table doesn't have a template row, the new row will be completely empty. The only way to add content to the new table row in this case would be to either use the raw HTML DOM, or alternatively to set the label property on a CAF.Table.Row.Model, and use the CAF.Table.Model's set(x, item) method to update the row with the HTML column content specified by the label.

If, however, a row is added to the table using the CAF model, and the table does have a template row, the contents of the template row will be copied to the new row (hence the name: the template row is a template for new rows). The template row will contain the same controls as any other controls in the table, but initialized with the values from the content-provider in its "template" state (the state after it is setTemplateRow() method is called). With the template content in place, the row's values property can be used to set the values of the all the controls in the new row to something other than their template state.

The CAF.Table.Model's template() method returns the CAF.Table.Row.Model for the template row, if it exists. See "Table Row Tools" on page 214 for more information about working with template rows.

Paging

The CAF.Table.Model has several methods for accessing the table's paging state. The getFirst() method returns the (zero-based) index of the first row on the current page. The getRows() method returns the current page size (not necessarily the actual number of rows on the page; the size() method — inherited from the CAF.Select.Model — does that). The getRowCount() method returns the total number of rows in the table, or if the total is unknown, returns -1. The getRowSelectedCount() method returns the total number of selected rows in the table.

All of the CAF.Table.Model row manipulation methods inherited from CAF.Select.Model, such as add(), remove(), setSelected(), etc. operate only on the current page; for example, calling table.setSelected(0) sets the first item of the table on the current page selected, not the first item in the entire table. Two special methods, however, go(first, rows, sort) and move(from, to), operate across page boundaries, and use row indices relative to the entire table.

The go(first, rows, sort) method pages and/or re-sorts the entire table. It allows three arguments; if the first argument (first) is greater or equal to zero, it pages the table to the index specified by the argument; if the second argument (rows) is greater than zero, it sets the new page size to the size specified by the argument; and if the third argument is non-null and non-empty, it re-sorts the table by the columns, and in the order, specified by the sort string.

The sort string is a CSV list of column IDs to sort, in order from the primary sort, to the secondary sort, to the tertiary sort, etc. Each column ID should be followed by a + or - to indicate ascending order or descending order. For example, the sort string "user+,permissions+,modified-" would specify the table be sorted first by the user column in ascending order (A's at the top), then by the permissions column in ascending order, and finally by the modified column in descending order (most recent at the top).

The move(from, to) method moves a row from the table index specified by the first argument (from) to the table index specified by the second argument (to). It does not page the table, even if one or both of the indices are outside of the current page. Note that the table content-provider must implement the IReorderableTableContentProvider interface (and return true for the interface's isReorderable() method) in order to for row movements to be persisted. See "Table Row Tools" on page 214 for more information.

Listeners

Custom client-side code can listen for changes to the table by registering table-change and row-change listeners. Table-change events are raised when the table is paged, or when its select-all/none state is toggled. Row-change events are raised when a row is added, removed, updated, or selected via the CAF.Table.Model.

A table-change listener can be added via the addTableChangeListener(fn) method. Its one argument (fn) is the listener function; when a table-change event is raised the listener function is passed two arguments: the table ID and the event type ("page", "select-all", or "select-none").

A row-change listener can be added via the addRowChangeListener(fn) method. Its one argument (fn) is also the listener function; when a row-change event is raised the listener function is passed three arguments: the table ID, the row ID, and the event type ("add", "remove", or "update", or "select"). The following example alerts the user when a row in the table is selected:

```
var table = CAF.model('#{activePageBean.clientIds['myTableId']}');
table.addRowChangeListener(function(tableId, rowId, eventType) {
   if (eventType "select") {
     var row = CAF.model(rowId);
     if (table.isSelected(row))
        alert(row.getValue() + " selected");
   }
});
```

CAF.Tree.Model

The CAF.Tree.Model extends the CAF.Table.Model, adding methods for expanding and collapsing tree rows. Individual rows can be collapsed or expanded via collapse(value), expand(value), or toggle(value), given their server-side row-ID (the value arg). Also, the CAF.Tree.Row.Model roots can be determined via the getRoots() method.

The state of the tree rows themselves can be determined via methods on the CAF.Tree.Row.Model, which of course extends the CAF.Table.Row.Model. The getParent() method of the CAF.Tree.Row.Model returns the CAF.Tree.Row.Model for the parent of the row, while the getChildren() method returns the array of CAF.Tree.Row.Model children (in order). The isOpen() method returns true if the row is expanded and false if collapsed; the setOpen(open) method expands or collapses the row (unlike other CAF.Select.Item.Model and CAF.Table.Row.Model methods, this one takes effect immediately, instead of only when the container model's set() method is invoked with the changed model).

Roots can be added to or removed from the tree using the CAF.Select.Model's add() and remove() methods. Other rows can be added or removed via the addChild() and removeChild() methods of a particular row's CAF.Table.Row.Model. Again, unlike other CAF.Select.Item.Model and CAF.Table.Row.Model methods, addChild() and removeChild() take effect immediately, instead of only when the container model's set() method is invoked with the changed model.

Row templating on a tree works the same as with the table (see above). In the following example, if the tree can template rows, and it contains a dropdown with an ID of "myDropdownId" and a output text control with an ID of "myOutputId", the following script would append a new row with a row-ID of "my-new-row" to the very last node in the tree, and set the row's dropdown to the "value-new" item and the row's output text to display "My New Row":

```
var tree = CAF.model(treeId);
var node = tree.getRoots().last();
while (node && node.getHasChildren()) node = node.getChildren().last();
node.addChild(new CAF.Tree.Row.Model("my-new-row", { myDropdownId: "value-new", myOutputId: "My New Row" });
```

Note that with an async tree, not all rows may be present on the client. If a row has children, but its children are not yet on the client, the CAF.Tree.Row.Model will return true for getHasChildren(), but an empty array for getChildren(). The CAF.Tree.Row.Model's isLoaded() method can also be used as a shortcut to test for this state (it will return false if the row has children but the children are not on the client). To force the tree to request the children from the server, invoke setOpen(true) on the CAF.Tree.Row.Model, or invoke expand(value) on the CAF.Tree.Model.

In addition to the table-change and row-change events raised by tables, trees raise a few extra events. They raise "expand-all" and "collapse-all" table-change events when the tree's expand-all/collapse-all state is modified. They raise "toggle" row-change events when individual rows are toggled expanded or collapsed. And the async tree raises "loaded" row-change events when the children of a row have been loaded.

Advanced Considerations

Script Placement

Because a model merely wraps a DHTML control, when a page is rendered on the client, the control's HTML element must exist on page (the browser must have parsed the HTML and included it in page's DOM) before a model for that control can be created. For example, if a script block on a page has some code which models a control as the page is being loaded, the script block must be placed after the modeled control. If, however, the script block that models a control does so in a function that is executed later (after the page is loaded — say, as the result of a mouse click on the control), the script block can be placed anywhere on the page.

Model References and Global Variables

A common scenario for advanced scripters is to extract complicated JavaScript functions into an external script file (using the IncludeScript control). Because client-side control IDs are generated by a complicated algorithm on the server, and aren't predictable ahead of time on the client, it is often necessary to calculate client-side control IDs (via the clientIds property of the active page bean: #{activePageBean.clientIds['myControlId']}) while rendering the page, and then store the IDs in global variables. Note that it is generally a better practice to store the control ID than the model instance; the model instance may cache certain information about the control and over time become out-of-sync with the actual control state.

A good way to prevent global variables defined by one portlet instance's page from colliding with variables from another's is to define a master global object that can be shared among all instances of the portlet on a page (or all instances of a set of portlets on a page). For example, all webMethods Task Engine portlets might share a global object called WMTE, conditionally defined in a script block at the top of each portlet page:

```
// if the WMTE object does not exist, create a new empty WMTE object
if (WMTE)
  var WMTE = {};
```

In JavaScript, all objects can be treated as a generic map, which maps arbitrary string keys (the object's property names) to arbitrary object values (the object's property values). So this master global object can be used to map arbitrary global variable names to arbitrary values; for example:

```
WMTE['taskCount'] = 32;
```

More often, a developer will use a key unique to the portlet instance, like its view root ID, to map a global variable to a dynamic value:

```
WMTE['#{activePageBean.clientIds['myRootId']}' + '.taskCount'] =
#{activePageBean.taskCount};
```

Then any function defined in an external script file, as long as the function was passed the view root ID as a parameter, can lookup the taskCount global variable specific to the current portlet instance:

```
function wmte_doSomething(rootId) {
    alert(WMTE[rootId + '.taskCount']);
}
```

This technique can be used with function names as well; for example, an external script file might contain the following code:

```
if (!WMTE)
  var WMTE = {};

WMTE.doSomething = function(rootId) { alert(WMTE[rootId + '.taskCount']); }
```

This function can be invoked as WMTE.doSomething(rootId) (or as WMTE['doSomething'](rootId)).

Image URLs

Controls that render icons or other images often allow the icon image to be configurable. The image may be specified by either an absolute URL such as this:

http://www.google.com/images/logo_sm.gif

or a URL relative to the root of the web application, such as this:

```
/myimages/myicon.gif
```

Relative URLs should not contain the web application's context path and should not be encoded for WSRP because the control does both automatically.

If you need to produce the image URL independently, for example because you have some client-side code that dynamically modifies the image source, you can produce the URL to an image (or any other WAR resource) with the following java code:

```
String url = "/myimages/myicon.gif";
FacesContext context = getFacesContext();

url = context.getApplication().getViewHandler().getResourceURL(context, url);
url = context.getExternalContext().encodeResourceURL(url);
```

Some images with well-known URLs may be replaced automatically with skinned images.

In a standard Web application project, the source directory for the root of the web application is called WebContent (visible in the Navigator view, for example):

```
wm_myapp
build
src
WebContent
myimages
myicon.gif
```

Related Topics

"Skinning" on page 213

Skinning

When you use image or icon controls, the My webMethods Server runtime automatically replaces references to images from your web application's /skin/images directory with skinned images, if the images are in the current user's skin. If the images are not in the current user's skin, the default images from your web application are used.

For example, say you have two images in your web application's /skin/images directory, Icon_Error.gif and Icon_No-Error.gif:

```
wm_myapp
build
src
WebContent
META-INF
skin
images
Icon_Error.gif
Icon_No-Error.gif
WEB-INF
```

You have two image controls on your page, one with /skin/images/Icon_Error.gif for its URL property value, and one with /skin/images/Icon_No-Error.gif for its URL property value. If the current user's skin has an Icon_Error image, the first control displays the skin's Icon_Error image. If the current user's skin does not have an Icon_No-Error image, the second control displays the Icon_No-Error image from your web application's /skin/images directory.

See "Image URLs" on page 212 for more information about referencing images.

Related Topics

▶ "Image URLs" on page 212

Table Row Tools

Related Topics

- ▶ "Adding and Removing Rows" on page 214
- ▶ "Moving Rows" on page 215
- ▶ "Selecting Rows" on page 216

Adding and Removing Rows

Several controls add or remove rows to or from a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IUpdateableTableContentProvider. A row is first added or removed dynamically on the client, and then later, when the enclosing form is posted to the server (and passes validation), the row is added to or removed from the table content provider.

Generating New Rows on the Server

The following default content providers implement IUpdateableTableContentProvider:

- com.webmethods.caf.faces.data.object.ListTableContentProvider
- com.webmethods.caf.faces.data.object.SelectableListTableContentProvider
- com.webmethods.caf.faces.data.tree.object.NodeTreeContentProvider
- com.webmethods.caf.faces.data.tree.object.ListTreeContentProvider
- com.webmethods.caf.faces.data.sql.RowSetTableContentProvider
- com.webmethods.caf.faces.data.object.ListTableContentProvider

Before you add rows to a content provider implementation that extends the ListTableContentProvider (such as SelectableListTableContentProvider, NodeTreeContentProvider, ListTreeContentProvider, and others), you must first

- Assign a unique ID to each row by setting the provider's rowIdBinding property to a binding expression that will produce a unique ID for each row.
- 2 Enable "template" rows to be created; do either of the following:
 - Specify a the row class via the Rowtype property (the class must have a public, zero-argument constructor).
 - Override the newRow() method to create a new, blank row object, and override the getCanTemplateRow() method to return true.

Generating New Rows on the Client

When a new row is added to a table, the content for each column in the row is generated by applying the column's content to the "template" row from the table's content provider.

Add Row Control Instances

- ▶ "Add Row Button" on page 586
- ▶ "Add Row Icon" on page 587
- ▶ "Add Row Link" on page 589

Remove Row Control Instances

- ▶ "Remove Row Button" on page 597
- ▶ "Remove Row Icon" on page 599
- ▶ "Remove Row Link" on page 600

Moving Rows

Several controls move the selected rows up or down in a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IReorderableTableContentProvider. A row is first moved dynamically on the client, and then later, when the enclosing form is posted to the server (and passes validation), the row is moved in the table content provider.

The following default content providers implement IReorderableTableContentProvider:

- com.webmethods.caf.faces.data.object.ListTableContentProvider
- com.webmethods.caf.faces.data.object.SelectableListTableContentProvider

Move Row Control Instances

- ▶ "Move Row Down Button" on page 590
- "Move Row Down Icon" on page 591
- "Move Row Down Link" on page 593
- "Move Row Up Button" on page 594
- ▶ "Move Row Up Icon" on page 595
- ▶ "Move Row Up Link" on page 596

Selecting Rows

Several controls select and deselect rows in a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.ISelectableTableContentProvider. This interface provides two fundamental selection modes: 'selected' and 'unselected'. In 'selected' mode, all rows are implicitly unselected; rows can be explicitly selected by adding them to the list of selected rows. In 'unselected' mode, all rows are implicitly selected; rows can be explicitly deselected by adding them to the list of unselected rows. This makes it practical for a user to select all rows (or all rows except one or two) in a large table.

A regular "Select Row" control allows a user to select and deselect an individual row (the row containing the control). For example, the "Select Row Checkbox" control allows a user to toggle a row between selected and unselected by toggling the state of a checkbox in the row. A "Select All Rows" control allows a user to select and deselect all rows in the entire table (by toggling between the 'selected' and 'unselected' modes). A "Select All Visible Rows" control allows a user to select and deselect all rows in just the current, visible page of the table.

Selecting Tree Rows

In addition to the conventional "Select Row" controls, trees can use "Select Row Mixed-State" controls, where the selection state of descendant rows are reflected by the mixed state of ancestor rows. For instance, with the "Select Row Tri-State Checkbox" control, the selection state of container rows is controlled entirely by the selection state of descendants: if all descendants of a row are selected, the row is selected; if all descendants are unselected, the row is unselected; and if some descendants are selected and some are unselected, the row is mixed. With the "Select Row Quad-State Checkbox" control, the selected/unselected state of container rows is independent of the rows' descendants; however, a row's checkbox displays as mixed if any descendants have a different selection state than the row.

Optionally, to facilitate the calculation of mixed-state, a tree content-provider can implement the com.webmethods.caf.faces.data.tree.ISelectableTreeContentProvider interface.

Select Row Instances

- "Select Row Checkbox" on page 604
- ▶ "Select Row Link" on page 606
- ▶ "Select Row On Click" on page 607

Select Row Mixed-State Instances

- "Select Row Tri-State Checkbox" on page 609
- ▶ "Select Row Quad-State Checkbox" on page 608

Select All Rows Instances

- ▶ "Select All Rows Checkbox" on page 602
- ▶ "Select All Visible Rows Checkbox" on page 603

Client-Side Libraries

The core composite application client-side JavaScript library, automatically included into every composite application HTML page, is built on the open-source Prototype library. Prototype provides many convenience functions for manipulating strings, arrays, and DOM objects, as well as for making asynchronous requests. See Using Prototype.js and the Prototype.js Documentation List for documentation on the Prototype library.

The core composite application library provides a few extra Prototype-style convenience functions, a facility for logging debug messages, a facility for dynamically loading other script files, a facility for displaying alert and progress dialogs, access to the "Client-Side Model" on page 201, and access to the "Scriptaculous Effects" on page 198 library.

Related Topics

- ▶ "CSV" on page 217
- ▶ "String" on page 218
- ▶ "Logger" on page 218
- ▶ "CAF.Dialog Class" on page 220
- "CAF.Progress Class" on page 221
- ▶ "CAF.Tooltip Class" on page 222
- "CAF.Request Class" on page 222
- "CAF.Updater Class" on page 223
- "CAF.Draggable Class" on page 223
- ▶ "CAF.Droppable Class" on page 225

CSV

Composite applications often use CSV (comma-separated value) strings to serialize lists for transfer between server and client.

String

The composite application library extends the core JavaScript String object (and beyond what Prototype adds) with extra methods:

- formatMessage(params) (formats a parameterized message much like the java.text.MessageFormat class)
- escapeJS() (escapes for use in JavaScript blocks within programmatically generated HTML fragments)
- escapeRegExp() (escapes for use by new RegExp())
- trim() (trims leading and trailing whitespace).

Logger

The CAF Logger sends client-side log messages to registered client-side log appenders, much like Java's log4j logging system does on the server side. Each message has a category (an arbitrary string) and a level (either Logger.ERROR, Logger.WARN, Logger.INFO, or Logger.DEBUG). ERROR messages are displayed to the user via a modal dialog, and WARN messages are displayed via a modeless dialog.

Messages can be logged using the following static functions:

- Logger.error()
- Logger.warn()
- Logger.info()
- Logger.debug()
- Logger.log()

The properties of an object can be logged at the DEBUG level using the Logger.dump() function.

Related Topics

- ▶ "Logger Bookmarklet" on page 218
- ▶ "Logger and Firebug" on page 219

Logger Bookmarklet

INFO and DEBUG log messages are not displayed in the UI by default, but you can view them by installing the CAF Logger bookmarklet from an instance of My webMethods Server:

http://your.server:port)/wm_cafshared/ui/js/bookmarklet/

From that page, drag the CAF Logger link to your browser's Bookmark Toolbar. To use the bookmarklet, first navigate to a composite application page, and then click it. A CAF Logger window pops up, containing all the CAF messages logged to the main window (or tab, if you use Firefox or Internet Explorer 7).

You can clear the old log messages using the Clear button on the CAF Logger window. You can limit the log messages displayed by entering a regular-expression into the Regexp match field. For example, if you type Info into the field and press ENTER, only messages with the string "Info" in them (like messages logged at the INFO level) are displayed. Delete the contents of the field and press ENTER to re-display all messages.

You can evaluate a line of JavaScript in the context of the main window (or tab, if you use Firefox or Internet Explorer 7) by entering some JavaScript into the Quick eval field at the bottom of the CAF Logger window. For example, if you type document.location.href into the field and press ENTER, a message is logged with the main window's (or tab, if you use Firefox or Internet Explorer 7) page URL. When the Quick eval field is focused, you can use the up and down arrow keys to scroll through the list of previously entered commands (similar to the history feature of a UNIX shell).

Logger and Firebug

You can direct CAF Logger messages to the Firebug console with the following javascript:

```
Logger.appendToFirebug();
```

Alternately, you could start a Logger. Firebug Appender instance that listens to a custom log level and/or category (so as to avoid all the CAF debug messages):

```
Logger.history.replay(new Logger.FirebugAppender(Logger.INFO, "my-
category").start().append);
```

Library

The composite application library dynamically loads script files on demand. With the Library.register(id, url) function one can register a named group of script files to be loaded later. The Library.load(id, onload) function loads the named group (or directly-specified group) of script files, and then executes the specified callback function when loading is completed. For example, the scripts needed to execute "Scriptaculous Effects" on page 198 are pre-registered as the "scriptaculous/effects" group. To execute a scriptaculous effect, you must call Library.load() to load the scriptaculous effect library, pass a function to execute once the library has finished loading:

```
Library.load("scriptaculous/effects", function() {
    Effect.Fade("myElement");
});
```

Instead of using a registered group name, you also can pass Library.load() the URL of a script to load directly. Multiple urls (or named groups) can be specified as a space-separated string (for example "http://example.com/foo.js http://example.com/bar.js").

CAF. Dialog Class

A composite application can simulate the core JavaScript alert(), confirm(), and prompt() popup dialogs with a DHTML "popup window". The main functional difference (other than the ability to display HTML-formatted messages) is that the core JavaScript versions block program flow until the user closes the dialog, whereas the CAF. Dialog versions do not. If you want to perform some action as a result of the user input, you have to pass a callback function. For example, the following code prompts a user for a zodiac sign, and then responds to a click on the prompt's OK button by displaying a second (alert) dialog:

```
CAF.Dialog.prompt("What's your sign?", "Capricorn", function(value) {
    CAF.Dialog.alert("I'm a " + value + " too!");
});
```

You can also apply validation to a prompt() dialog, like with the following dialog, which will not let the user click OK without first entering a number between one and ten:

```
CAF.Dialog.prompt(
    'On a scale of 1-10, rate this prompt:',
    '10',
    function(value) {
        CAF.Dialog.alert('You entered ' + value + '.');
    function(value) {
        CAF.Dialog.alert('You canceled!');
        buttons: {
            ok: {
                validate: function(form) {
                    var n = parseInt(form.prompt);
                    if (n < 1 | | n > 10)
                       return 'Please enter a number between 1 and 10.';
                    return '';
            }
        }
);
```

The CAF.Dialog.show() function can create more sophisticated dialogs with custom content. The following example displays an account creation dialog, with Username, Password, and Confirm Password fields, as well as a custom "?" button.

```
CAF.Dialog.show({
   title: "Create an Account",
   content: "Username: <input name='username' class='input10'>"
          + "<br>"
           + "Password: <input name='password' type='password' class='input10'>"
           + "Confirm Password: <input name='password2' type='password' class='input10'>",
   buttons: {
       ok: {
            label: "Create",
            "class": "button6",
            defaultCommand: true,
            fn: function(form) {
               CAF.Dialog.alert("Created new user " + form.username);
            validate: function(form) {
                var errors;
                if (form.username.length < 8) {</pre>
                    if (!errors) errors = {};
                    errors.username = "Username must be at least 8 characters long.";
                if (form.password.length < 8) {</pre>
                    if (!errors) errors = {};
                    errors.password = "Password must be at least 8 characters long.";
                if (form.password != form.password2) {
                    if (!errors) errors = {};
                    errors.password2 = "Confirm Password must match original Password.";
                return errors;
        },
        cancel: {
            label: "Cancel",
            "class": "button6"
        },
        popup: {
            label: " ? ",
            validate: function(form) {
               CAF.Dialog.alert("I dunno either");
                return {}; // prevent dialog from closing
        }
});
```

CAF.Progress Class

CAF can display a progress bar over any HTML element. The CAF.Progress.overlay(element, msg) function overlays a progress bar over an HTML element, whereas the CAF.Progress.insert(element, msg) function replaces the element's content with a progress bar. The CAF.Progress.show(element, msg) function decides for you whether to overlay or insert the progress bar; it inserts the progress bar if the element

isn't large enough to contain the progress bar (thus enlarging the element to the necessary size), otherwise it overlays the bar. The CAF.Progress.hide(element) function hides the progress bar (regardless of whether it was overlaid or inserted).

For example, the following line of JavaScript overlays a progress bar over the entire page:

```
CAF.Progress.overlay(document.body, "Please wait...");
```

The next line hides that progress bar:

```
CAF.Progress.hide(document.body);
```

CAF. Tooltip Class

You can programmatically attach a ToolTip to any HTML element. The following code attaches a ToolTip with the text "This control does something totally sweet!" to the control with the ID myControlId:

CAF.Request Class

The CAF.Request class extends Prototype's asynchronous request class, Ajax.Request, allowing asynchronous JSF requests. Its API is identical to Ajax.Request, and you can use it in the same way as Ajax.Request. See the <u>Using the Ajax.Request class</u> Prototype documentation.

You should use CAF.Request instead of Ajax.Request when making requests to the JSF servlet, adding a scope property to the request options, the value of which should be the JSF view root's client ID. For example, given a URL and the viewRootClientId, the following code alerts the response:

```
new CAF.Request(url, {
    method: 'get',
    onComplete: function(transport) {
        alert("response: " +
    CAF.Request.extractResponseText(transport.responseText));
    },
    scope: viewRootClientId
});
```

To simulate a CAF command request, you can call the _createFragmentURL() method on the model object of a CAF control (see "Client-Side Model" on page 201 for more information), which will create a URL that will refresh that control. You can convert the current state of the control's form to a URL parameter string with Prototype's Form.serialize() function. With those two strings (and having set the CAF.Command.field().value to the id of the command control to invoke) you can create a

new CAF.Request which will post the current form state to the server and render the updated control fragment as a result:

```
var m = CAF.model('#{activePageBean.clientIds['myCommand']}');

// set active command on form CAF.Command.field(m.form).value = m.id;

// calculate url and form post parameters
var url = m._createFragmentURL();
var formParams = Form.serialize(m.form);

// send request
var request = new CAF.Request(url, {
    method: "post",
    parameters: formParams,
    scope: CAF.viewRootId(m.id),
    onComplete: function(transport) {
    CAF.Dialog.alert(request.extractResponseText());
    }
});
```

CAF. Updater Class

The CAF.Updater class extends Prototype's asynchronous request class, Ajax.Updater, allowing the content of an element to be updated via an asynchronous JSF request. Its API is identical to Ajax.Updater, and you can use it in the same way as Ajax.Updater. See the <u>Using the Ajax.Updater class</u> Prototype documentation.

You should use CAF.Updater instead of Ajax.Updater when making requests to the JSF servlet, adding a scope property to the request options, the value of which should be the JSF view root's client ID. For example, given a URL and the ID of the element to update, the following code updates the element with the response content:

```
new CAF.Updater(id, url, {
   method: 'get',
   scope: CAF.viewRootId(id)
});
```

CAF.Draggable Class

The CAF.Draggable class extends the "Scriptaculous Effects" on page 198 Draggable class, providing uniform drag behavior across different types of draggable elements. Unlike the default Draggable behavior, the CAF.Draggable object creates a copy of the element to drag and inserts it into a special drag container, ensuring the user drags around the copy of the element and not the original (it uses the CAF.Draggable.duplicateAsHTML() method to generate unique temporary IDs for the dragged element and its children). If you wanted to override this behavior, say, to replace the dragged element content with an

image, you could override the startDrag() method to replace the default drag container's content with your custom content:

```
Object.extend(new CAF.Draggable(id), {
    startDrag: function(event) {
        CAF.Draggable.prototype.startDrag.apply(this, arguments);
        var container = this.getDragContainer();
        container.innerHTML = "<img
src='http://www.google.com/intl/en_ALL/images/logo.gif'>";
     }
});
```

The CAF.Draggable startDrag() and endDrag() methods also maintain a CAF.Draggable.current object while the drag is in process. The CAF.Draggable.current object has three useful fields:

- draggable—the current draggable object
- element—the original draggable element (not the one in the drag container, which is the original drag target)
- result—an open-ended string representing the result of the drag operation

By convention, if the drag ends in a successful drop, and the drop handler is able to duplicate the dragged content, the drop handler will set the CAF.Draggable.current.result field to "move." This notifies the CAF.Draggable object that it may delete the original draggable element (effectively performing a "move" operation). The CAF.Draggable object will do this if the CAF.Draggable's move option is set to true.

The CAF.Draggable class also has two static functions, model() and rows(). These are especially useful in drop handler functions (see "CAF.Droppable Class" on page 225) to determine what is in the drag container. The model() function models the drag container's content as a single element (useful for handling the drop of a simple control like an image or text control); the rows() function models the drag container's content as an array of table rows (useful for handling the drop of a complex drop like a panel containing multiple controls, or a table or a list row).

Note that, like with other "Scriptaculous Effects" on page 198, CAF.Draggable (and CAF.Droppable) objects require the "scriptaculous/effects" package to be loaded before they can be referenced. Therefore you should wrap any effects or drag-drop JavaScript code in an anonymous function, and pass that function to the Library.load() function:

```
Library.load('scriptaculous/effects', function() {
   new CAF.Draggable(id, { cursor: "move" });
});
```

CAF.Droppable Class

The CAF.Droppable class encapsulates and extends the functionality of the "Scriptaculous Effects" on page 198 Droppables.add() method. Instead of using the Droppables.add() method and its options to register a droppable element, like the following:

```
Droppables.add(id, { onDrop: function() {} });
```

you can use a syntax that mirrors that of the Draggable object constructor, like the following:

```
Object.extend(new CAF.Droppable(id), {
    onDrop: function(src, dst) {
        CAF.Droppable.prototype.onDrop.apply(this, arguments);
        CAF.Draggable.current.result = ""; // cancel move
    }
});
```

The CAF.Droppable onDrop() method copies the value of sub-controls with the same local ID from the dropped element into the sub-controls of the CAF.Droppable's target element (the drop target). (Note that JSF allows multiple controls to use the same ID, as long as each of those controls is in a different naming-container; see "Control ID Reference" on page 192.) For example, say we have two DIVs, each in a different JSF naming-container, each with three text controls; the first DIV is the target of a CAF.Draggable object, the second the target of a CAF.Droppable. The IDs of the text controls in the first DIV are "name," "description," and "notes;" the IDs of the text controls in the second DIV are "item-number," "name," and "description." If the user drags and drops the first DIV onto the second, the CAF.Droppable's onDrop() method will copy the first DIV's "name" and "description" text controls of the second (but it will not modify the second DIV's "item-number" text control).

If the CAF.Droppable's onDropSetValue option is set to true, the onDrop() method will simply copy the value of the dropped control to the value of the target control (ignoring sub-controls). For example, say we have two image elements, the first image is the target of a CAF.Draggable object, the second image is the target of a CAF.Droppable (and with the onDropSetValue option set to true). If the user drags and drops the first image onto the second, the onDrop() method will simply copy the first image control's value (the image element's URL) to the second image.

The easiest way to customize a CAF.Droppable object's drop handling is to specify a function for the handleDrop option. This function will be invoked by the CAF.Droppable's onDrop() method before it does anything else. The handleDrop callback function is passed the same two arguments as the original onDrop() function: src and dst. The src argument is the dragged element, which (if the dragged element is from a CAF.Draggable) should be the drag container; the dst argument is the CAF.Droppable's target element (the drop target). The handler should return true if it fully handled the drop, and false to allow the default onDrop() processing to execute (which copies the content of the dropped control to the target control).

The following example demonstrates using a custom handleDrop handler to override the default CAF.Droppables onDrop() behavior. It gets the value of the "name" sub-control in the dropped control (if available), and sets the drop target control's value to the value of the dropped "name" sub-control:

```
new CAF.Droppable(id, {
    handleDrop: function(src, dst) {
    var rows = CAF.Draggable.rows(src);
    if (rows.length) {
        var nameId = rows[0].getControlId("name");
        if (nameId) {
            CAF.model(dst).setValue(CAF.model(nameId).getValue());
        }
        }
        return false;
    }
});
```

Another useful callback is configurable by means of the allowDrop option. It's a function that is invoked by the CAF.Droppable's onActivate() method. The allowDrop callback is passed two arguments: src and dst. The src argument is the dragged element, which (if the dragged element is from a CAF.Draggable) should be the drag container; the dst argument is the target element (the drop target). The handler should return true if the src element can be dropped and false if it cannot.

The following example demonstrates using a custom allowDrop function. It returns true if the drag container contains the phrase "bad mojo," and false if it does not contain "bad mojo":

```
new CAF.Droppable(id, {
    allowDrop: function(src, dst) {
        return !/bad mojo/i.test(src.innerHTML);
    }
});
```

Filterable Controls

JSF listbox, swapbox, and dropdown controls all have the same basic behavior and share a common base component and data model (the "select" data model). JSF table, list, and tree controls also share a common base component and data model (the "table" data model). Software AG has extended both the select and the table data models with an interface that allows the list of items exposed by the model to be filtered. Portlet developers can create a custom data model implementation with special logic for filtering, or they can use the default implementations.

The default select data model filter implementation loops through every item in the model, and tries to match the item's label with the filter expression using the standard

commonsearch keyword rules (* = wildcard, space = and, quotes = exact phrase). The default table data model filter implementation allows the portlet developer to specify a JSF binding expression to apply to each row to produce a string to match against; it loops through every item in the model and tries to match this string to the filter expression using the standard commonsearch keyword rules. This JSF binding expression allows a portlet developer choose what row content to filter.

Client-side filtering uses the standard commonsearch keyword rules in a similar manner to the default server-side filtering implementation, but will not allow portlet developers to customize the filtering logic.

Portlet developers can write JavaScript to call the filter method on any select or table control based on arbitrary user input. There is also a standard "Filter Input" on page 314 control; developers need only specify the ID of the select or table control to which it should apply.

These select controls have client-side filtering capability:

- "Autocomplete Text Input" on page 290
- "Checkbox Group" on page 294
- "Combobox" on page 297
- "Dropdown" on page 305
- "Extended Select-Many Listbox" on page 308
- "Extended Select-One Listbox" on page 310
- "Radio Button Group" on page 330
- "Select-Many Listbox" on page 335
- "Select-One Listbox" on page 342
- "Swapbox" on page 349

These table controls have client-side filtering capability:

- "Async List" on page 398
- "Async Listbox" on page 401
- "Async Tabs" on page 403
- "Async Table" on page 531
- "Async Categorized Table" on page 526
- "Categorized Table" on page 553
- Listbox" on page 406
- "Table" on page 570

■ "Tabs" on page 411

These select controls have server-side filtering capability:

- "Autocomplete Text Input" on page 290
- "Extended Select-Many Listbox" on page 308
- "Extended Select-One Listbox" on page 310
- "Swapbox" on page 349

These table controls have server-side filtering capability:

- "Async List" on page 398
- "Async Listbox" on page 401
- "Async Tabs" on page 403
- "Async Table" on page 531
- "Async Categorized Table" on page 526

Related Topics

- ▶ "CAF.Select.Model" on page 228
- ▶ "Examples of Filtering" on page 229

CAF.Select.Model

The following method has been added to the client-side CAF.Select.Model class (note that the CAF.Table.Model class extends this class):

```
/**
 * Filters the control's options with the specified expression string.
 * If the control is bound to a model with server-side filtering
enabled,
 * the control's options will be filtered on the server;
 * otherwise they will be filtered one the client.
 *
 * @param expr Expression string. Null or empty ("") signals show the default set of options.
 */
CAF.Select.Model.prototype.filter = function(expr) {};
```

Examples of Filtering

These topics provide simplified examples of filtering. Before you try them, you should be familiar with the creation of portlet applications and how to develop portlets using the design canvas.

- "Filtered Listbox (Medium List)" on page 229
- "Filtered Listbox (Large List)" on page 230
- "Filtered Swapbox (Medium List)" on page 230
- "Filtered Swapbox (Large List)" on page 231
- "Filtered Dropdown (Medium List)" on page 232
- "Filtered Combobox (Medium List)" on page 232
- "Autocomplete Text Field (Large List)" on page 233
- "Filtered Table (Current Page)" on page 233
- "Filtered Table (Entire Table)" on page 233

Filtered Listbox (Medium List)

The resulting list box displays a small to medium number of items (10-100). An input field below the list box allows the user to type in a filter expression. As the user types into the filter input field, items are immediately added to or removed from the listbox. The list is filtered on the client; no asynchronous requests to the server are made.

- 1 Create a select model from a Web service or database query, IS service, or static list, using the Bindings view's New Content Provider wizard.
- 2 Drag an Extended Select-Many Listbox control (Control > Input > Extended Select-Many ListBox) from the Palette view to the design canvas, and an Option Group control (Control > Input > Option Group) into the Extended Select-Many Listbox control.
- 3 Drag a Filter Input control (Control > Input > Filter Input) from the Palette view to the design canvas, below the Extended Select-Many Listbox control.
- 4 Bind the Option Group control's value to the select model created in step 1.
- 5 Bind the Extended Select-Many Listbox control's value to a property on the page bean (so as to capture the value(s) the user ultimately selects).
- 6 Set the ID property of the Extended Select-Many Listbox control and the For property of the Filter Input control to "my-listbox" (connecting the Filter Input control to the Extended Select-Many Listbox control).

Filtered Listbox (Large List)

The resulting list box displays a large number of items (100+). An input field below the list box allows the user to type in a filter expression. When the user types an expression into the filter field and clicks the Go button, items are added to or removed from the list box. The list is filtered on the server, so there might be a delay between the time the user clicks the Go button and when the filtering takes effect (a progress icon should indicate that the request is processing).

- 1 Create a select model from a Web service or database query, IS service, or static list, using the Bindings view's New Content Provider wizard. Set the select model's Filterable property to true, enabling server-side filtering on the default select model implementation.
- 2 Drag an Extended Select-Many Listbox control (Control > Input > Extended Select-Many ListBox) from the Palette view to the design canvas, and an Option Group control (Control > Input > Option Group) into the Extended Select-Many Listbox control.
- 3 Drag a Filter Input control (Control > Input > Filter Input) from the Palette view to the design canvas, below the Extended Select-Many Listbox control.
- 4 Bind the Option Group control's value to the select model created in step 1.
- 5 Bind the Extended Select-Many Listbox control's value to a property on the page bean (so as to capture the value(s) the user ultimately selects).
- 6 Set the ID property of the Extended Select-Many Listbox control and the For property of the Filter Input control to "my-listbox" (connecting the Filter Input control to the Extended Select-Many Listbox control).

Alternatively, for optimum performance you can implement a custom select model instance that performs the filtering at the data store level (for example, in a custom SQL query).

Filtered Swapbox (Medium List)

The resulting swapbox displays a small to medium number of items (10-100). An input field below the swapbox allows the user to type in a filter expression. As the user types into the filter input field, items are immediately added to or removed from the swapbox, in both the available and selected lists. The list is filtered on the client; no asynchronous requests to the server are made.

- 1 Create a select model from a Web service or database query, IS service, or static list, using the Bindings view's New Content Provider wizard.
- 2 Drag a Swapbox control (Control > Input > Swapbox) from the Palette view to the design canvas, and an Option Group control (Control > Input > Option Group) into the Swapbox control.
- 3 Drag a Filter Input control (Control > Input > Filter Input) from the Palette view to the design canvas, below the Swapbox control.

- 4 Bind the Option Group control's value to the select model created in step 1.
- 5 Bind the Swapbox control's value to a property on the page bean (so as to capture the value(s) the user ultimately selects).
- 6 Set the ID property of the Swapbox control and the For property of the Filter Input control to "my-listbox" (connecting the Filter Input control to the Swapbox control).

Filtered Swapbox (Large List)

The resulting swapbox displays a large number of items (100+). An input field below the swapbox allows the user to type in a filter expression. When the user types an expression into the filter field and clicks the Go button, items are added to or removed from the swapbox, in both the available and selected lists. The list is filtered on the server, so there might be a delay between the time the user clicks the Go button and when the filtering takes effect (a progress icon should indicate that the request is processing).

- 1 Create a select model from a Web service or database query, IS service, or static list, using the Bindings view's New Content Provider wizard. Set the select model's Filterable property to true, enabling server-side filtering on the default select model implementation.
- 2 Drag a Swapbox control (Control > Input > Swapbox) from the Palette view to the design canvas, and an Option Group control (Control > Input > Option Group) into the Swapbox control.
- 3 Drag a Filter Input control (Control > Input > Filter Input) from the Palette view to the design canvas, below the Swapbox control.
- 4 Bind the Option Group control's value to the select model created in step 1.
- 5 Bind the Swapbox control's value to a property on the page bean (so as to capture the value(s) the user ultimately selects).
- 6 Set the ID property of the Swapbox control and the For property of the Filter Input control to "my-listbox" (connecting the Filter Input control to the Swapbox control).

Alternatively, for optimum performance you can implement a custom select model instance that performs the filtering at the data store level (for example, in a custom SQL query).

Filtered Dropdown (Medium List)

The resulting dropdown displays a small to medium number of items (10-100). When the user selects the dropdown (clicks it or tabs to it), and begins typing, the dropdown selects the item that starts with the text the user typed. Note that this is the default behavior of a native HTML dropdown. Also note that this does not allow the user to type in wildcards or match aspects of the item other than the start of its label.

- 1 Create a select model from a Web service or database query, IS service, or static list, using the Bindings view's New Content Provider wizard.
- 2 Drag a Dropdown control (Control > Input > Dropdown) from the Palette view to the design canvas, and an Option Group control (Control > Input > Option Group) into the Dropdown control.
- 3 Bind the Option Group control's value to the select model created in step 1.
- 4 Bind the Dropdown control's value to a property on the page bean (so as to capture the value(s) the user ultimately selects).

Filtered Combobox (Medium List)

The resulting combobox displays a small to medium number of items (10-100). When the user selects the combobox (clicks it or tabs to it), and begins typing, the combobox selects the item that starts with the text the user typed, autocompleting the remaining item text. Note that this does not allow the user to type in wildcards or match aspects of the item other than the start of its label.

- 1 Create a select model from a Web service or database query, IS service, or static list, using the Bindings view's New Content Provider wizard.
- 2 Drag a Combobox control (Control > Input > Combobox) from the Palette view to the design canvas, and an Option Group control (Control > Input > Option Group) into the Dropdown control.
- 3 Bind the Option Group control's value to the select model created in step 1.
- 4 Bind the Combobox control's value to a property on the page bean (so as to capture the value(s) the user ultimately selects).

Autocomplete Text Field (Large List)

The user can enter an arbitrary value into a text field. When the user selects the field (clicks it or tabs to it), a scrollable menu displays below the text field (like a combobox). As the user pauses while typing, items are added to or removed from the menu. The list is filtered on the server, so there might be a delay in updating the list to reflect the field's current text (a progress icon should indicate that the request is processing).

- 1 Create a select model from a Web service or database query, IS service, or static list, using the Bindings view's New Content Provider wizard. Set the select model's Filterable property to true, enabling server-side filtering on the default select model implementation.
- 2 Drag an Autocomplete Text Input control (Control > Input > Autocomplete Text Input) from the Palette view to the design canvas, and an Option Group control (Control > Input > Option Group) into the Autocomplete Text Input control.
- 3 Bind the Option Group control's value to the select model created in step 1.
- 4 Bind the Autocomplete Text Input control's value to a property on the page bean (so as to capture the value(s) the user ultimately selects).

Filtered Table (Current Page)

The resulting table displays one page of rows (for example, 1-100 of 10,000). An input field above the table allows the user to type in a filter expression. As the user types into the filter input field, items are immediately added to or removed from the table. The table is filtered on the client and only applies to the current page (rows 1-100, not 101-10000); no asynchronous requests to the server are made.

- 1 Create a table model from a Web service or database query, IS service, or static list using the Bindings view's New Content Provider wizard.
- 2 Drags Table control (Control > Table > Table) from the Palette view to the design canvas.
- 3 Drag a Filter Input control (Control > Input > Filter Input) from the Palette view to the design canvas, above the Table control.
- 4 Bind the Table control's value to the select model created in step 1.
- 5 Set the ID property of the Table control and the For property of the Filter Input to "my-table" (connecting the Filter Input control to the Table control).

Filtered Table (Entire Table)

The resulting Table displays one page of rows (for example, 1-10 of 10000). An input field above the table allows the user to type in a filter expression. When the user types an expression into the filter field and clicks the Go button, items are added to or removed from the table. The table is filtered on the server and applies to the entire table (rows 1-10000). There might be a delay between the time the user clicks the Go button and when the filtering takes affect (a progress icon should indicate that the request is processing).

This sample is similar to "Filtered Table (Current Page)" on page 233, but in the New Content Provider wizard (step 1), the developer sets the table model's "filterable" property to true and configures a binding-expression that produces the text for each row against which to filter (for example "#{row.name} #{row.description}"). Alternatively, for optimum performance, the developer can implement a custom table model instance that performs the filtering at the data store level (for example, in a custom SQL query).

- 1 Create a table model from a Web service or database query, IS service, or static list using the Bindings view's New Content Provider wizard. Set the select model's Filterable property to true, enabling server-side filtering on the default select model implementation. Configure a binding-expression that produces the text for each row against which to filter (for example #{row.name} #{row.description}).
- 2 Drag a Table control (Control > Table > Table) from the Palette view to the design canvas.
- 3 Drag a Filter Input control (Control > Input > Filter Input) from the Palette view to the design canvas, above the Table control.
- Bind the Table control's value to the select model created in step 1.
- 5 Set the ID property of the Table control and the For property of the Filter Input to "my-table" (connecting the Filter Input control to the Table control).

Alternatively, for optimum performance you can implement a custom select model instance that performs the filtering at the data store level (for example, in a custom SQL query).

Chapter 11. Behavior Controls

Custom Drag	. 236
Custom Drop	. 237
Drag To Move Rows	. 239
Drag To Reorder Columns	. 243
Drag To Resize	. 244
Drag To Resize Columns	. 245
Raise On Change	. 246
Synchronize Values	. 247

Custom Drag

com.webmethods.caf.faces.behavior.CustomDrag

CAF Base Controls Component Library

A behavior that allows a user to drag another control.

If the "move" property is set to true, and if the target control is successfully dragged and dropped (and the drop-target's handleDrop behavior sets the CAF.Draggable.current.result flag to "move"), the target control's HTML content is removed from the current page. Otherwise, dragging and dropping has no effect on the target control.

The "allowDrag" handler is invoked every time a user tries to drag this behavior's target control. The handler is passed two arguments: draggable and event. The draggable argument is the target control's CAF.Draggable object; the event argument is the browser mouse event that precipitates the drag. The handler should return true if the target control can be dragged and false if it cannot.

See "CAF.Draggable Class" on page 223 in "Client-Side Libraries" on page 217 for more low-level JavaScript details.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	Target control to which this applies. If unspecified, applies to this control's nearest ancestor table.		
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display control.		true
move	Move	boolean	True to remove the target control's HTML content if dragged and dropped successfully.		false
Expert Properties					
allowDrag	Allow Drag	String	Custom JavaScript	'allowDrag' handler.	
propagateEvent	Propagate Event	boolean	True to allow a mo precipitating drag event listeners.	ouse-event- to propagate to other	false

Name	Display Name	Туре	Description	Choices	Default			
Expert Display Properties								
cursor	Cursor	String	CSS style property to use for mouse pointer when user hovers mouse pointer over target control.	auto (standard cursor for control type), cross hair, default (arrow cursor), pointer (pointing hand), move, text (Ibeam), wait (clock), help (question-mark)	move			
	Children							
	None.							
	Facets							
	None.							

Custom Drop

com.webmethods.caf.faces.behavior.CustomDrop

CAF Base Controls Component Library

A behavior that allows a user to drop a draggable item onto another control.

This behavior's default drop handling is to copy the value of sub-controls with the same local ID from the dropped control into the sub-controls of the behavior's target control. (Note that JSF allows multiple controls to use the same ID, as long as each of those controls is in a different naming-container. See "Control ID Reference" on page 192.) For example, say we have two block panels, each in a different naming-container, each with three text controls; the first has a CustomDrag behavior and the second has a CustomDrop behavior. The IDs of the text controls in the first panel are "name," "description," and "notes;" the IDs of the text controls in the second panel are "itemnumber," "name," and "description." If the user drags and drops the first panel into the second, this behavior will copy the first panel's "name" and "description" text control values to the "name" and "description" text controls of the second (it will not modify the second panel's "item-number" text control).

The preceding example only copies output text control values, so no information about the drag-and-drop event will ever reach the server. But if we add a hidden input control with an id of "item-id" to each panel, when a user drags the first panel into the second, the value of this hidden input will be copied from the first panel to the second. And when the

second panel's enclosing form is submitted, the new value of the second panel's hidden input will be submitted to the server, allowing server-side code to do any processing that is needed as a result of the drag-and-drop event (for example, to notify a back-end data store that the second panel has a new item-id, and to display the new item-number, name, and description that corresponds to that item-id).

If this behavior's "onDropSetValue" is set to true, the behavior's drop handling changes to simply copying the value of the dropped control to the value of this behavior's target control (ignoring sub-controls). For example, say we have two image controls, the first with a CustomDrag behavior, and second with a CustomDrop behavior (and "onDropSetValue" set to true), and the user drags the first image control onto the second. Because the second image control's "onDropSetValue" behavior property is set to true, the behavior will simply copy the first image control's value (the image URL) to the second image.

To further customize this behavior, you can specify custom client-side JavaScript code to execute when the user drags another control over the target control (the "allowDrop" handler), and when the user drops another control on the target control (the "handleDrop" handler).

The "allowDrop" handler is invoked every time a user tries to drag another control over this behavior's target control (that is, whenever the dragged control enters the target control's boundaries). It is passed two arguments: src and dst. The src argument is the dragged html element, which is a container for a temporary copy of the original control's main html element; the dst argument is the target control's main html element. The handler should return true if the control can be dropped and false if it cannot.

The "handleDrop" handler is invoked every time a user tries to drop another control on this behavior's target control (but only if the "allowDrop" handler returned true or was not specified). It is passed two arguments: src and dst. The src argument is the dragged html element, which is a container for a temporary copy of the original control's main HTML element; the dst argument is the target control's main HTML element. The handler should return true if it fully handled the drop, and false to allow the default handleDrop processing to execute (which is to copy the content of the dropped control to the target control).

See "CAF.Draggable Class" on page 223 in "Client-Side Libraries" on page 217 for more low-level JavaScript details.

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	Target control to which this applies. If unspecified, applies to this control's nearest ancestor table.	
id	ID	String	Control id.	

Name	Display Name	Туре	Description	Default
rendered	Rendered	boolean	True to display control.	true
Expert Properties				
allowDrop	Allow Drop	String	Custom JavaScript 'allowDrop' handle	r.
handleDrop	Handle Drop	String	Custom JavaScript 'handleDrop' hand	ler.
onDropSetValue	On Drop Set Value	boolean	True to use simple copy of the control value from dragged control to dropped control as the default handleDrop behavior; false to use complex copy of sub-control values from dragged control to dropped control.	false
Expert Display Prop	erties			
hoverClass	Hover Class	String	CSS class name to apply to the target of the user hovers a dragged item over the control.	
	Children			
	None.			
	Facets			
	None.			

Drag To Move Rows

com.webmethods.caf.faces.behavior.DragToMoveRows

CAF Base Controls Component Library

A behavior that allows a user to add, remove, copy, or reorder the rows in a table by dragging them.

If the "reorder" property is set to true, the com.webmethods.caf.faces.data.IReorderableTableContentProvider interface of the target table's content provider is used to persist the new row order. The new row ordering is not sent to the server, however, until the table's enclosing form is posted to the server (and it passes validation).

If the "add" and/or "remove" properties are set to true, the com.webmethods.caf.faces.data.IUpdateableTableContentProvider interface of the target table's content-provider is used to persist the row additions/removals. The updated table

information is not sent to the server, however, until the table's enclosing form is posted to the server (and it passes validation).

If the "copy" property is set to true and the "remove" property is set to false, dragging a row from one table to another (assuming the other table is set to "add") will add a new row to the other table, without removing the original row from the first table. If the "remove" property is set to "true", the "copy" property is ignored and dragging a row from one table to another will remove the original row from the first table.

To populate the controls of a newly added row (dragged and dropped from another table), this behavior tries to copy the value of controls with the same local ID from the dropped row into the new row. (Note that JSF allows multiple controls to use the same id, as long as each of those controls is in a different naming-container -- and a table is a naming-container. See "Control ID Reference" on page 192). For example, say you have two tables, each table with three columns, each column with a single text control. The IDs of the text controls in the first table are "name," "description," and "notes;" the IDs of the text controls in the second table are "item-number," "name," and "description." If the user drags a row from the first table to the second, the new row in the second table will be populated with the dropped row's "name" and "description" text control values (and have an empty "item-number" text control).

The preceding example only copies output text control values, so when the new row is updated on the server, no information about the new row (other than that it exists) will be available to the server. But if we add a hidden input control with an ID of "item-id" to each table, when a user drags a row from the first table to the second, the value of this hidden input will be copied from the dropped row to the new row. And when the second table's enclosing form is submitted, the value of the new row's hidden input will be submitted to the server, allowing server-side code to do any processing that is needed to commit the changes to the table (for example, to notify a back-end data store that a new row has been added with the specified item-id, and to fill in the item-number, name, and description fields of the new row object).

Also see "Table Row Tools" on page 214 some more information about adding, removing, and reordering rows.

To further customize this behavior, you can specify custom client-side JavaScript code to execute when the user attempts to start dragging a row (the "allowDrag" handler), when the user drags a row over the table (the "allowDrop" handler), and when the user drops a row on the table (the "handleDrop" handler).

The "allowDrag" handler is invoked every time a user tries to drag a row in this behavior's target table. It is passed two arguments: draggable and event. The draggable argument is the dragged row's CAF.Draggable object; the event argument is the browser mouse event which precipitates the drag. The handler should return true if the row can be dragged and false if it cannot.

The "allowDrop" handler is invoked every time a user tries to drag a row over this behavior's target table (that is, whenever the dragged row enters the table's boundaries). It is passed two arguments: src and dst. The src argument is the dragged HTML element, which is a container for a temporary copy of the original row element (in a temporary

table); the dst argument is the HTML drop-target element, which is a container for the target table. The handler should return true if the row can be dropped and false if it cannot.

The "handleDrop" handler is invoked every time a user tries to drop a row on this behavior's target table (but only if the "allowDrop" handler returned true or was not specified). It is passed two arguments: src and dst. The src argument is the dragged HTML element, which is a container for a temporary copy of the original row element (in a temporary table); the dst argument is the HTML drop-target element, which is a container for the target table. The handler should return true if it fully handled the drop, and false to allow the default handleDrop processing to execute (which is, when the dropped row is from another table and "add" is allowed, to create a new row and copy the values of controls with the same local ID from the original dropped row to the new row; and when the dropped row is from the same table and "reorder" is allowed, to move the row to its new location).

See "CAF.Draggable Class" on page 223 in "Client-Side Libraries" on page 217 for more low-level JavaScript details.

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	Target control to which this applies. If unspecified, applies to this control's nearest ancestor table.	
id	ID	String	Control id.	
rendered	Rendered	boolean	True to display control.	true
add	Add	boolean	True to allow rows from other tables to be dragged into this table, adding a new row to the table.	false
copy	Сору	boolean	True to allow rows from this table to be dragged to other tables. This is automatically true if either the "remove" or the "reorder" property is true.	false

Name	Display Name	Туре	Description	Default			
remove	Remove	boolean	True to allow rows to be removed from this table by dropping them successfully onto another table.	false			
reorder	Reorder	boolean	True to allow the user to drag rows in this table to reorder them.	false			
Expert Properties							
allowDrag	Allow Drag	String	Custom JavaScript 'allowDrag' handler.				
allowDrop	Allow Drop	String	Custom JavaScript 'allowDrop' handler.				
handleDrop	Handle Drop	String	Custom JavaScript 'handleDrop' handler.				
Expert Display Properties							
hoverClass	Hover Class	String	CSS class name to apply to the target control when the user hovers a dragged item over the target control.				

Children

None.

Facets

None.

Drag To Reorder Columns

com. we bmethods. caf. faces. behavior. Drag To Reorder Columns

CAF Base Controls Component Library

A behavior that allows the user to reorder the columns of the target table by dragging the column headers.

Changes to the order of the columns can be persisted by binding the "columnDisplay" property to a portlet-preferences-bean property, and binding the "storePreferences" property to the storePreferences() method of a portlet-preferences-bean. In addition, manual processing is needed during the faces render phase to change the order of the columns in the table to match the persisted values.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	Target control to wh unspecified, applies ancestor table.		
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display control.		true
storePreferences	Store Preferences	Method Binding	Method to invoke to	persist portlet p	oreferences.
Display Properties					
columnDisplay	Column Display	String	Array of column IDs	s, in order from l	eft to right.
	Children				
	None.				
	Facets				
	None.				

Drag To Resize

com.webmethods.caf.faces.behavior.DragToResize

CAF Base Controls Component Library

A behavior that allows the user to resize the target control by dragging its edges. Changes to the size of the control are not persisted.

If the target control is a table, resizing the vertical size of the table changes the number of rows-per-page displayed by the table. This doesn't work well unless the table is an "Async Table" on page 531 with client-side-caching turned on (that is to say, set the Lazy Load property Client Side Cache to true).

Properties

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	Target control to which this applies. If unspecified, applies to this control's parent.	
id	ID	String	Control id.	
rendered	Rendered	boolean	True to display control.	true
resizeTable	Resize Table	boolean	True to resize the nearest table in the target control.	false
Display Properties				
resizeTop	Resize Top	boolean	True to allow the user to resize the target control by dragging its top edge.	false
resizeLeft	Resize Left	boolean	True to allow the user to resize the target control by dragging its left edge.	false
resizeBottom	Resize Bottom	boolean	True to allow the user to resize the target control by dragging its bottom edge.	false
resizeRight	Resize Right	boolean	True to allow the user to resize the target control by dragging its right edge.	false

Children

None.

Facets

None.

Drag To Resize Columns

com.webmethods.caf.faces.behavior.DragToResizeColumns

CAF Base Controls Component Library

A behavior that allows the user to resize the columns of the target table by dragging the column header borders.

Changes to the size of the columns can be persisted by binding the "columnWidths" property to a portlet-preferences-bean property, and binding the "storePreferences" property to the storePreferences() method of a portlet-preferences-bean. In addition, manual processing is needed during the faces render phase to set the widths of the columns in the table to match the persisted values.

The "columnWidths" property is persisted as a csv-style string representation of a map, mapping column IDs to column widths. For example, the following string defines the width of "colOne" to be "60%", the width of "colTwo" to be "40%", and the width of "colThree" to be "20px": "colOne=60%,colTwo=40%,colThree=20px".

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	Target control to which this applies. If unspecified, applies to this control's nearest ancestor table.		
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display control.		true
storePreferences	Store Preferences	Method Binding	Method to invoke to per-	sist portlet pre	eferences.
Display Properties					
columnWidths	columnWidths Column String Widths		Column widths as id=wi persisting new column to	* *	•
	Children				
	None.				
	Facets				
	None.				

Raise On Change

com. we bmethods. caf. faces. behavior. Raise On Change

CAF Base Controls Component Library

A behavior that raises a specified command when the value of the behavior's target changes.

Properties

Name	Display Name	Туре	Description	Default			
General Properties							
command	Command	Control ID Reference	Id of command to raise.				
id	ID	String	Control id.				
for	For	Control ID Reference	Target control to which this applies. If unspecified, applies to this control's parent.				
rendered	Rendered	boolean	True to display control.	true			

Children

None.

Facets

None.

Synchronize Values

com. we bmethods. caf. faces. behavior. Synchronize Values

CAF Base Controls Component Library

A behavior that synchronizes the (client-side) values of two or more controls.

Properties

Name	Display Name	Туре	Description	Default		
General Properties						
id	ID	String	Control id.			
for	For	Control ID Reference	Target control to which this applies. If unspecified, applies to this control's parent.			
rendered	Rendered	boolean	True to display control.	true		
with	With	Control ID References	Ids of other controls with which to synchronize. Separate multiple ids with commas.			

Children

None.

Facets

None.

Chapter 12. Command Controls

Async Command Button	250
Async Command Icon	252
Async Command Interval	255
Async Command Link	256
Async Hidden Command	259
Command Button	261
Command Icon	263
Command Link	265
Form	268
Hidden Command	271
Servlet Command Form	272

Async Command Button

 $com. we bmethods. caf. faces. command. as {\it ync.} Button$

CAF Base Controls Component Library

A button that invokes a server-side action when clicked by means of an asynchronous request. This control must be contained by a Command Form ancestor.

Name	Display Name	Туре	Description	Choices	Default		
General Properties							
id	ID	String	Control ID.				
refresh	Refresh	"Control ID Reference" on page 192	ID of control to refresh when asynchronous request completes. Empty signals refresh the entire form. "_none" signals refresh nothing. To refresh multiple controls, use a comma-separated list of references.				
rendered	Rendered	boolean	True to display	True to display control.			
toggle	Toggle	"Control ID Reference" on page 192	ID of the control to toggle. To toggle multiple controls, use a comma-separated list of references.				
Display Properties							
disabled	Disabled	boolean	True to display as "grayed-out".		false		
width	Width	String	Pre-defined button2 (~2 characters wide), button3, button button4, button5, button6, button7, width. button8, button9, button10, button12, button14, button16, button18, button20 (~20 characters wide)				
Action Properties							
action	Action	javax.faces.el. Method Binding	Simple (zero-argument) action callback.				
actionListener	Action Listener	javax.faces.el. Method Binding	Standard action callback.				
immediate	Immediate	boolean	True to process ActionListeners in the Apply false Request Values phase; false to process in the Invoke Application phase.				

Name	Display Name	Туре	Description	Choices	Default			
Value Properties								
value	Value	Object	Button label.					
Expert Display Pro	perties							
accesskey	Access Key	String	Keyboard sho	tcut letter.				
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL	(right-to-left)			
lang	Language	String	Language code	e for content.				
style	CSS Style	String	HTML "style"	attribute value.				
styleClass	CSS Class	String	HTML "class"	attribute value.				
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.					
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).					
Expert Client Side	Events							
onblur	Blur	script	Control losing	focus.				
onfocus	Focus	script	Control gainin	g focus.				
onclick	Click	script	Control clicked.					
ondblclick	Double Click	script	Control double	e-clicked.				
onkeydown	Key Down	script	Key pressed w	hen control is focused.				
onkeypress	Key Press	script	Key pressed a	nd released when control i	s focused.			
onkeyup	Key Up	script	Key released v	when control is focused.				
onmousedown	Mouse Down	script	Mouse button	pressed when over contro	1.			
onmousemove	Mouse Move	script	Mouse moved	when over control.				
onmouseout	Mouse Out	script	Mouse moved	out of control boundaries				
onmouseover	Mouse Over	script	Mouse moved	into control boundaries.				
onmouseup	Mouse Up	script	Mouse button	released when over contro	ol.			

Children

Any. Children are displayed as button label (after the content of the "value" property).

Control Parameter children are added as request parameters to the submitted form (if the button is clicked).

Facets

None.

Async Command Icon

com.webmethods.caf.faces.command.async.lcon

CAF Base Controls Component Library

An icon that invokes a server-side action when clicked by means of an asynchronous request. This control must be contained by a Command Form ancestor.

Name	Display Name	Туре	Description	Choices	Default	
General Properties						
id	ID	String	Control ID			
refresh	Refresh	"Control ID Reference" on page 192	ID of control to refresh when an asynchronous request completes. Empty signals refresh the entire form. "_none" signals refresh nothing. To refresh multiple controls, use a comma-separated list of references.			
rendered	Rendered	boolean	True to display control. tru		true	
toggle	Toggle	"Control ID Reference" on page 192	ID of the control to toggle. To toggle multiple controls, use a comma-separated list of references.			
Display Properties						
disabled	Disabled	boolean	True to display as "grayed-out." f		false	
alt	Alternate Text	String	Alternate text to display if image cannot be displayed.		e displayed.	
url	Icon URL String URL of (base) image source.					
mousedownUrl	Url Mouse Down String URL of optional image to display when mouse is pressed over icon.		ouse is			
mouseoverUrl	Mouse Over Icon URL	String	URL of optional image to display when mouse hovers over icon.			

Name	Display Name	Туре	Description Choices Default
Action Properties			
action	Action	javax.faces.el. Method Binding	Simple (zero-argument) action callback.
actionListener	Action Listener	javax.faces.el. MethodBindi ng	Standard action callback.
immediate	Immediate	boolean	True to process ActionListeners in the false Apply Request Values phase; false to process in the Invoke Application phase.
Value Properties			
value	Value	Object	Icon label (displayed before the icon).
Expert Properties			
charset	Charset	String	Character encoding of linked resource.
coords	Coordinates	String	Client-side image map coordinates.
hreflang	Href Language	String	Language of linked resource.
rel	Relationship	String	Link relationship from the current resource to the linked resource.
rev	Reverse Link	String	Link relationship from the linked resource to the current resource.
shap	Shape	String	Client-side default, rect, circle, poly image map shape.
type	Content type	String	Content type of linked resource.
Expert Display Prop	erties		
accesskey	Access Key	String	Keyboard shortcut letter.
dir	Directionality	String	Text direction LTR (left-to-right), RTL hint. (right-to-left)
lang	Language	String	Language code for content.
style	CSS Style	String	HTML "style" attribute value.
styleClass	CSS Class	String	HTML "class" attribute value.
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.

Name	Display Name	Туре	Description	Choices	Default
title	Element Title	String	HTML "title" attr tooltip).	ribute value (often displayed	d as a
Expert Client Side E	vents				
onblur	Blur	script	Control losing fo	ocus.	
onfocus	Focus	script	Control gaining	focus.	
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-o	clicked.	
onkeydown	Key Down	script	Key pressed who	en control is focused.	
onkeypress	Key Press	script	Key pressed and	released when control is fo	cused.
onkeyup	Key Up	script	Key released wh	en control is focused.	
onmousedown	Mouse Down	script	Mouse button pr	ressed when over control.	
onmousemove	Mouse Move	script	Mouse moved w	hen over control.	
onmouseout	Mouse Out	script	Mouse moved or	ut of control boundaries.	
onmouseover	Mouse Over	script	Mouse moved in	to control boundaries.	
onmouseup	Mouse Up	script	Mouse button re	leased when over control.	

Any. Children are displayed as the icon label (after the icon).

Control Parameter children are added as request parameters to the submitted form (if the button is clicked).

Facets

Async Command Interval

com. we bmethods. caf. faces. command. as ync. Interval

CAF Base Controls Component Library

A client-side script that invokes a server-side action on a timed interval via an asynchronous request. This control must be contained by a Command Form ancestor.

Name	Display Name	Туре	Description Default
General Properties	S		
id	ID	String	Control ID.
refresh	Refresh	"Control ID Reference" on page 192	ID of control to refresh when asynchronous request completes. Empty signals refresh the entire form. "_none" signals refresh nothing. To refresh multiple controls, use a comma-separated list of references.
toggle	Toggle	"Control ID Reference" on page 192	ID of the control to toggle. To toggle multiple controls, use a comma-separated list of references.
interval	Interval	int	Interval in 60 seconds between refreshes.
rendered	Rendered	boolean	True to display true control.
waitUntil Reloaded	Wait Until Reloaded	boolean	Wait until the false control finishes reloading to begin the next interval.
Display Properties	1		
disabled	Disabled	boolean	True to make false un-invocable.

Name	Display Name	Туре	Description Default
Action Properties			
action	Action	javax.faces.el. Method Binding	Simple (zero-argument) action callback.
actionListener	Action Listener	javax.faces.el. Method Binding	Standard action callback.
immediate	Immediate	boolean	True to process false ActionListeners in the Apply Request Values phase; false to process in the Invoke Application phase.

Control Parameter children are added as request parameters to the submitted form (if the command is invoked).

Facets

None.

Async Command Link

com. we bmethods. caf. faces. command. as ync. Link

CAF Base Controls Component Library

A link that invokes a server-side action when clicked via an asynchronous request. This control must be contained by a Command Form ancestor.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		

Name	Display Name	Туре	Description Choices	Default
refresh	Refresh	"Control ID Reference" on page 192	ID of control to refresh when an asynchr request completes. Empty signals refresh form. "_none" signals refresh nothing. To multiple controls, use a comma-separate references.	the entire o refresh
rendered	Rendered	boolean	True to display control.	true
toggle	Toggle	"Control ID Reference" on page 192	ID of the control to toggle. To toggle mul controls, use a comma-separated list of re	
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out".	
Action Properties				
action	Action	javax.faces.el. Method Binding	Simple (zero-argument) action callback.	
actionListener	Action Listener	javax.faces.el. Method Binding	Standard action callback.	
immediate	Immediate	boolean	True to process ActionListeners in the Apply Request Values phase; false to process in the Invoke Application phase.	false
Value Properties				
value	Value	Object	Link label.	
Expert Properties				
charset	Charset	String	Character encoding of linked resource.	
coords	Coordinates	String	Client-side image map coordinates.	
hreflang	Href Language	String	Language of linked resource.	

Name	Display Name	Туре	Description	Choices	Default
rel	Relationship	String	Link relationship from the current resource to the linked resource.		
rev	Reverse Link	String	Link relationship from the linked resource to the current resource.		
shap	Shape	String	Client-side image map shape.	default, rect, circle, poly	
type	Content type	String	Content type of linked resource.		
Expert Display P	Properties				
accesskey	Access Key	String	Keyboard short	cut letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL (right-to-left)	
lang	Language	String	Language code	for content.	
style	CSS Style	String	HTML "style" at	ttribute value.	
styleClass	CSS Class	String	HTML "class" at	ttribute value.	
tabindex	Tab Index	int		y) sequence, from 1 to r ence, negative integer	
title	Element Title	String	HTML "title" att tooltip).	tribute value (often dis	played as a
Expert Client Sic	de Events				
onblur	Blur	script	Control losing f	ocus.	
onfocus	Focus	script	Control gaining	focus.	
onclick	Click	script	Control clicked.		

Name	Display Name	Туре	Description Choices Default
ondblclick	Double Click	script	Control double-clicked.
onkeydown	Key Down	script	Key pressed when control is focused.
onkeypress	Key Press	script	Key pressed and released when control is focused.
onkeyup	Key Up	script	Key released when control is focused.
onmousedown	Mouse Down	script	Mouse button pressed when over control.
onmousemove	Mouse Move	script	Mouse moved when over control.
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.
onmouseover	Mouse Over	script	Mouse moved into control boundaries.
onmouseup	Mouse Up	script	Mouse button released when over control.

Any. Children are displayed as link label (after the content of the "value" property).

Control Parameter children are added as request parameters to the submitted form (if the button is clicked).

Facets

None.

Async Hidden Command

com.webmethods.caf.faces.command.async.Hidden

CAF Base Controls Component Library

A client-side script that invokes a server-side action when called by some other client-side code via an asynchronous request. Must be contained by a Command Form ancestor.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	

Name	Display Name	Туре	Description Default
refresh	Refresh	"Control ID Reference" on page 192	ID of control to refresh when an asynchronous request completes. Empty signals refresh the entire form. "_none" signals refresh nothing. To refresh multiple controls, use a comma- separated list of references.
rendered	Rendered	boolean	True to display true control.
toggle	Toggle	"Control ID Reference" on page 192	ID of the control to toggle. To toggle multiple controls, use a comma-separated list of references.
Display Properties			
disabled	Disabled	boolean	True to make unfalse invocable.
Action Properties			
action	Action	javax.faces.el. Method Binding	Simple (zero-argument) action callback.
actionListener	Action Listener	javax.faces.el. Method Binding	Standard action callback.
immediate	Immediate	boolean	True to process false ActionListeners in the Apply Request Values phase; false to process in the Invoke Application phase.

Control Parameter children are added as request parameters to the submitted form (if the command is invoked).

Facets

Command Button

javax.faces.HtmlCommandButton

CAF JSF Base Controls Component Library

A button that invokes a server-side action when clicked. Must be contained by a Command Form ancestor. A standard JavaServer Faces control.

Standard JSF Properties

Name	Display Name	Туре	Description Choices	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out"	. false
Action Properties				
action	Action	javax.faces.el. Method Binding	Simple (zero-argument) action	callback.
actionListener	Action Listener	javax.faces.el. Method Binding	Standard action callback.	
immediate	Immediate	boolean	True to process ActionListeners Apply Request Values phase; fa process in the Invoke Application	alse to
Value Properties				
value	Value	Object	Button label.	
Expert Display Prop	erties			
accesskey	Access Key	String	Keyboard shortcut letter.	
dir	Directionality	String	Text direction LTR (left-to-r hint. (right-to-left)	ight), RTL
image	Image	int	Server-side path to an image to the button label.	display in
lang	Language	String	Language code for content.	
style	CSS Style	String	HTML "style" attribute value.	

Name	Display Name	Туре	Description Choices Default
styleClass	CSS Class	String	HTML "class" attribute value.
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).
Expert Client Side E	Events		
onblur	Blur	script	Control losing focus.
onfocus	Focus	script	Control gaining focus.
onclick	Click	script	Control clicked.
ondblclick	Double Click	script	Control double-clicked.
onkeydown	Key Down	script	Key pressed when control is focused.
onkeypress	Key Press	script	Key pressed and released when control is focused.
onkeyup	Key Up	script	Key released when control is focused.
onmousedown	Mouse Down	script	Mouse button pressed when over control.
onmousemove	Mouse Move	script	Mouse moved when over control.
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.
onmouseover	Mouse Over	script	Mouse moved into control boundaries.
onmouseup	Mouse Up	script	Mouse button released when over control.

Extended CAF Properties

Name	Display Name	Туре	Description	Choices	Default
Display Properties					
width	Width	String	Pre-defined button width.	button2 (~2 character button3, button4, but button7, button8, but button12, button14, button18, button20 (~ wide)	tton5, button6, ton9, button10, button16,

Any. Children are displayed as button label (after the content of the "value" property).

Control Parameter children are added as request parameters to the submitted form (if the button is clicked).

Standard JSF Facets

None.

Extended CAF Facets

None.

Command Icon

com.webmethods.caf.faces.command.Commandlcon

CAF Base Controls Component Library

An icon that invokes a server-side action (when clicked). Must be contained by a "Form" on page 268 ancestor.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control	l.	true
Display Properties					
disabled	Disabled	boolean	True to display as "gra	yed-out".	false
alt	Alternate Text	String	Alternate text to displated displayed.	ny if image c	annot be
url	Icon URL	String	URL of (base) image so	ource.	
mousedownUrl	Mouse Down Icon URL	String	URL of optional image mouse is pressed over		when
mouseoverUrl	Mouse Over Icon URL	String	URL of optional image mouse hovers over ico	1 ,	when
Action Properties					
action	Action	javax.faces.el.Method Binding	Simple (zero-argumen	t) action call	back.

Name	Display Name	Туре	Description	Choices	Default
actionListener	Action Listener	javax.faces.el.Method Binding	Standard action callba	ack.	
immediate	Immediate	boolean	True to process ActionListeners in false the Apply Request Values phase; false to process in the Invoke Application phase.		
Value Properties					
value	Value	Object	Icon label (displayed	before the ico	on).
Expert Properties					
charset	Charset	String	Character encoding o	f linked reso	arce.
coords	Coordinates	String	Client-side image ma	p coordinates	5.
hreflang	Href Language	String	Language of linked re	esource.	
rel	Relationship	String	Link relationship from the current resource to the linked resource.		
rev	Reverse Link	String	Link relationship from the current resource.	n the linked 1	resource to
shap	Shape	String	Client-side image map shape.	default, re	ect, circle,
type	Content type	String	Content type of linked	d resource.	
Expert Display Prop	perties				
accesskey	Access Key	String	Keyboard shortcut let	ter.	
dir	Directionality	String	Text direction hint.	LTR (left- RTL (righ	
lang	Language	String	Language code for co	ntent.	
style	CSS Style	String	HTML "style" attribut	e value.	
styleClass	CSS Class	String	HTML "class" attribut	e value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attribute as a tooltip.	e value (often	displayed
Expert Client Side	e Events				
onblur	Blur	script	Control losing focus.		

Name	Display Name	Туре	Description	Choices	Default
onfocus	Focus	script	Control gaining focus.		
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clicked	d.	
onkeydown	Key Down	script	Key pressed when cor	ntrol is focus	ed.
onkeypress	Key Press	script	Key pressed and relea focused.	sed when co	ntrol is
onkeyup	Key Up	script	Key released when co	ntrol is focus	sed.
onmousedown	Mouse Down	script	Mouse button pressed	when over o	control.
onmousemove	Mouse Move	script	Mouse moved when o	ver control.	
onmouseout	Mouse Out	script	Mouse moved out of o	control bound	ders.
onmouseover	Mouse Over	script	Mouse moved into con	ntrol bounde	ers.
onmouseup	Mouse Up	script	Mouse button released	d when over	control.

Any. Children are displayed as the icon label (after the icon).

Control Parameter children are added as request parameters to the submitted form (if the button is clicked).

Facets

None.

Command Link

javax.faces.HtmlCommandLink

CAF JSF Base Controls Component Library

A link that invokes a server-side action when clicked. Must be contained by a Command Form ancestor. A standard JavaServer Faces control.

Standard JSF Properties

Name	Display Name	Туре	Description Choices	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description	Choices	Default
Action Properties					
action	Action	javax.faces.el. Method Binding	Simple (zero-argu	iment) action callback.	
actionListener	Action Listener	javax.faces.el. Method Binding	Standard action ca	allback.	
immediate	Immediate	boolean	Apply Request Va	ctionListeners in the alues phase; false to oke Application phase.	false
Value Properties					
value	Value	Object	Link label.		
Expert Properties					
charset	Charset	String	Character encodir	ng of linked resource.	
coords	Coordinates	String	Client-side image	map coordinates.	
hreflang	Href Language	String	Language of linked resource.		
rel	Relationship	String	Link relationship from the current resource to the linked resource.		
rev	Reverse Link	String	Link relationship current resource.	from the linked resource	ce to the
shap	Shape	String	Client-side image map shape.	default, rect, circle, poly	
target	Target	String	Name of frame or window whose content will be replaced with the linked resource.	_blank (new window (current frame), _par frame), _top (current other	ent (parent
type	Content type	String	Content type of li	nked resource.	
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcu	it letter.	
dir	Directionality	String	Text direction LTR (left-to-right), hint. RTL (right-to-left)		
lang	Language	String	Language code fo	r content.	

Name	Display Name	Туре	Description Choices Default		
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attribute value.		
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).		
Expert Client Side I	Events				
onblur	Blur	script	Control losing focus.		
onfocus	Focus	script	Control gaining focus.		
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clicked.		
onkeydown	Key Down	script	Key pressed when control is focused.		
onkeypress	Key Press	script	Key pressed and released when control is focused.		
onkeyup	Key Up	script	Key released when control is focused.		
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved when over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		
Expert Display Prop	perties				
accesskey	Access Key	String	Keyboard shortcut letter.		
dir	Directionality	String	Text direction LTR (left-to-right), hint. RTL (right-to-left)		
image	Image	int	Server-side path to an image to display in the button label.		
lang	Language	String	Language code for content.		
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attribute value.		
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).		

Extended CAF Properties

Name	Display Name	Туре	Description	Choices	Default		
Display Properties							
disabled	Disabled	boolean	True to display as	"grayed-out".	false		
	Children						
	Any. Children are displayed as link label (after the content of the "value" property).						
	Control Param link is clicked).		n are added as reque	st parameters to the sul	omitted form (if the		
	Standard JSF Fa	cets					
	None.						
	Extended CAF Fa	acets					
	None.						

Form

javax.faces.HtmlForm

CAF JSF Base Controls Component Library

A standard JavaServer Faces form.

Standard JSF Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cont	rol.	true
Expert Properties					
accept	Accept	Collection	List of content types the server accepts for uploaded files.		
acceptcharset	Accept Charset	Collection	List of character enc	odings the server a	ccepts.

Name	Display Name	Туре	Description	Choices	Default
enctype	Encoding Type	String	Form content type.	application/x- www-form- urlencoded, multipart/form- data	application/x- www-form- urlencoded
target	Target	String	Name of frame or window whose content will be replaced with the response from this form.	_blank (new wind frame), _parent (p. (current window),	arent frame), _top
Expert Display Prop	erties				
dir	Directionality	String	Text direction hint.	LTR (left-to-right),	RTL (right-to-left)
lang	Language	String	Language code for c	ontent.	
style	CSS Style	String	HTML "style" attribu	ıte value.	
styleClass	CSS Class	String	HTML "class" attribu	ıte value.	
title	Element Title	String	HTML "title" attribu	te value.	
Expert Client Side E	vents				
onblur	Blur	script	Control losing focus		
onfocus	Focus	script	Control gaining focu	ıs.	
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-click	ed.	
onkeydown	Key Down	script	Key pressed when co	ontrol is focused.	
onkeypress	Key Press	script	Key pressed and rele	eased when control i	s focused.
onkeyup	Key Up	script	Key released when o	control is focused.	
onmousedown	Mouse Down	script	Mouse button presse control.	ed when over	
onmousemove	Mouse Move	script	Mouse moved when	over control.	
onmouseout	Mouse Out	script	Mouse moved out of boundaries.	f control	
onmouseover	Mouse Over	script	Mouse moved into c	ontrol boundaries.	
onmouseup	Mouse Up	script	Mouse button releas control.	ed when over	

Extended CAF Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
defaultCommand	Default Command	String	ID of command control key when form is focus	l to submit if user presse sed.	s enter
defaultFocus	Default Focus	String	ID of control to focus when page is loaded.	_first (first control in form), other	_first
requireSession Token	Require Session Token	boolean	True to require a session form (which inhibits C Forgery attacks). Note makes it impractical to contained by the Form submitting the HTML control (for example, manually build an HTM Web server to invoke the form to the server to	that, by design, this invoke a command control except by form rendered by the nakes it impractical to ML form on some other	true
validateOnBlur	Validate On Blur	boolean	True to automatically when they lose focus, finputs when their valu	alse to validate form	false

Children

Any.

Standard JSF Facets

None.

Extended CAF Facets

Hidden Command

com.webmethods.caf.faces.command.HiddenCommand

Base Controls Component Library

A client-side script that invokes a server-side action when called by some other client-side code. Must be contained by a Command Form ancestor.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to make un- invocable.	false
Action Properties				
action	Action	javax.faces.el. Method Binding	Simple (zero-argument callback.) action
actionListener	Action Listener	javax.faces.el. MethodBindin g	Standard action callback.	
immediate	Immediate	boolean	True to process ActionListeners in the Apply Request Values phase; false to process in the Invoke Application phase.	false

Children

Control Parameter children are added as request parameters to the submitted form (if the command is invoked).

Facets

Servlet Command Form

com.webmethods.caf.faces.command.ServletCommandForm

Base Controls Component Library

A JSF form that when submitted, submits to the JSF servlet, instead of to the JSF portlet.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	ntrol.	true
defaultCommand	Default Command	String	ID of command co key when form is t	ntrol to submit if use focused.	er presses enter
defaultFocus	Default Focus	String	ID of control to focus when page is loaded.	_first (first control in form), other	_first
Expert Properties					
accept	Accept	Collection	List of content types the server accepts for uploaded files.		
acceptcharset	Accept Charset	Collection	List of character er	ncodings the server a	ccepts.
enctype	Encoding Type	String	Form content type.	application/x- www-form- urlencoded, multipart/form- data	application/x -www-form- urlencoded
target	Target	String	Name of frame or window whose content will be replaced with the response from this form.	_blank (new window), _self (current frame), _parent (parent frame), _top (current window), other	
Expert Display Proper	ties				
dir	Directionality	String	Text direction hint.	LTR (left-to-right), left)	RTL (right-to-
lang	Language	String	Language code for	content.	

Name	Display Name	Туре	Description	Choices	Default
style	CSS Style	String	HTML "style" attril	oute value.	
styleClass	CSS Class	String	HTML "class" attril	bute value.	
title	Title	String	HTML "title" attrib	ute value.	
Expert Client Side Eve	ents				
onsubmit	Submit	script	Form about to be s	ubmitted.	
onreset	Reset	script	Form about to be re	eset.	
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clic	cked.	
onkeydown	Key Down	script	Key pressed when	control is focused.	
onkeypress	Key Press	script	Key pressed and re	eleased when contro	ol is focused.
onkeyup	Key Up	script	Key released when	control is focused.	
onmousedown	Mouse Down	script	Mouse button pres	sed when over cont	rol.
onmousemove	Mouse Move	script	Mouse moved whe	en over control.	
onmouseout	Mouse Out	script	Mouse moved out	of control bounders	6.
onmouseover	Mouse Over	script	Mouse moved into	control bounders.	
onmouseup	Mouse Up	script	Mouse button relea	ased when over con	trol.

Any.

Facets

Chapter 13. Dialog Controls

Modal Dialog	276
Modeless Dialog	278
Portal Resource Picker Dialog	280
Principal Picker Dialog	283
Progress Dialog	286

Modal Dialog

com.webmethods.caf.faces.panel.ModalDialog

Base Controls Component Library

A panel that encapsulates the style and behavior of a modal dialog. A modal dialog forces the user to choose an option in the dialog (usually by clicking a button) before the user can do anything outside of the dialog (contrast this with a "Modeless Dialog" on page 278). This control is initially hidden; when toggled visible, it disables all other controls on the page until it is toggled hidden again.

Like other "hideable" controls, this control can be toggled between visible and hidden through client-side JavaScript code. Special "toggle" controls encapsulate this code within controls that can be created and configured visually, using the webMethods CAF. See "Hideable Controls" on page 194 for more information about hideable controls. See "Toggle Controls" on page 197 for more information about toggle controls.

This control has four facets: "title", "submit", "cancel", and "other". The last three facets are intended to encapsulate the default layout for the submit (or OK), cancel, and miscellaneous other buttons. For example, if you place a cancel button in the "cancel" facet, and an OK button in the "submit" facet, by default the modal dialog will layout the cancel button in the bottom-right corner of the dialog, with the OK button directly to its left. The "title" facet is intended to display a title for the dialog (if a title is appropriate).

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to include cont (note that setting th prevent the control visible).		true
Display Properties					
defaultCommand	Default Command	"Control ID Reference " on page 192	ID of command con presses enter key w is focused.	trol to submit if user hen control content	

Name	Display Name	Туре	Description	Choices	Default
defaultFocus	Default Focus	"Control ID Reference " on page 192	ID of control to focus when control is toggled visible.	_first (first input control contained by this control), other	_first
disableWhenHidden	Disabled When Hidden	boolean	True to disable cont when control is hid them from submitti	1	false
Lazy Load Properties					
lazy	Lazy Load	boolean	True to load control asynchronously on		false
progressDelay	Progress Delay	int	Milliseconds to wai progress dialog.	t before showing	0
progressMsg	Progress Message	String	Message to display in progress dialog.		Defaults to localized message; in English, "Loading "
refreshOnShow	Refresh On Show	boolean		sible; false to refresh ol is initially hidden	false
suppressInputs	Suppress Inputs	Collection	validation/update p	ls for which to disable processing when handl est (comma separated)	
twoPass	Two Pass	boolean	True to load contenteven if control is initially visible.	tially visible; false to	false
Expert Display Properties					
progressFlashOnLoad	Flash On Load	boolean	True to flash contro finishes loading asy		true
width	Width	String	Width of dialog in p	pixels (100px) or perce	nt (100%).

Any. Children of this control are displayed as the dialog's content.

Facets

Name	Display Name	Description	Allowed Types
cancel	Cancel	Cancel button container.	Any (usually a Toggle Button or a Command Button).
other	Other	Miscellaneous button container.	Any.
submit	Submit	Submit (or OK) button container.	Any (usually a Toggle Button or a Command Button).
title	Title	Dialog title.	Any.

Modeless Dialog

com.webmethods.caf.faces.panel.ModelessDialog

Base Controls Component Library

A panel that encapsulates the style and behavior of a modeless dialog. A modeless dialog allows the user to continue to manipulate controls outside of the dialog, without having to choose an option in the dialog (usually by clicking a button) right away (contrast this with a "Modal Dialog" on page 276). This control is initially hidden.

Like other "hideable" controls, this control can be toggled between visible and hidden through client-side JavaScript code. Special "toggle" controls encapsulate this code within controls that can be created and configured visually, using the webMethods CAF. See "Hideable Controls" on page 194 for more information about hideable controls. See "Toggle Controls" on page 197 for more information about toggle controls.

This control has four facets: "title", "submit", "cancel", and "other". The last three facets are intended to encapsulate the default layout for the submit (or OK), cancel, and miscellaneous other buttons. For example, if you place a Cancel button in the "cancel" facet, and an OK button in the "submit" facet, by default the modeless dialog will lay out the cancel button in the far-right corner of the dialog, with the OK button directly to its left. The "title" facet is intended to display a title for the dialog (if a title is appropriate).

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to include con page (note that sett: prevent the control visible).	ing this to false will	true
Display Properties					
defaultCommand	Default Command	"Control ID Reference" on page 192		ntrol to submit if user atrol content is focuse	
defaultFocus	Default Focus	"Control ID Reference" on page 192	ID of control to focus when control is toggled visible.	_first (first input control contained by this control), other	_first
disableWhenHidden	Disabled When Hidden	boolean	True to disable controls when controls preventing them from	rol is hidden,	false
Lazy Load Properties					
lazy	Lazy Load	boolean	True to load control asynchronously on		false
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog.		0
progressMsg	Progress Message	String	Message to display in progress dialog.		Defaults to localized message; in English, "Loading. "
refreshOnShow	Refresh On Show	boolean	True to refresh cont control is toggled v refresh content only initially hidden and visible.	isible; false to y if control is	false

Name	Display Name	Туре	Description	Choices	Default
suppressInputs	Suppress Inputs	Collection	List of ids of controls for which to disable validation/update processing when handling an asynchronous request (comma separated).		ndling an
twoPass	Two Pass	boolean	even if control is	ent asynchronously, initially visible; false synchronously if y visible.	false
Expert Display Properties	3				
progressFlashOn Load	Flash On Load	boolean	True to flash con finishes loading	trol when content asynchronously.	true

Any. Children of this control are displayed as the dialog's content.

Facets

Name	Display Name	Description	Allowed Types
cancel	Cancel	Cancel button container.	Any (usually a Toggle Button or a Command Button).
other	Other	Miscellaneous button container.	Any.
submit	Submit	Submit (or OK) button container.	Any (usually a Toggle Button or a Command Button).
title	Title	Dialog title.	Any.

Portal Resource Picker Dialog

com. we bmethods. caf. faces. panel. Portal Resource Picker Modal Dialog

MWS Control Component Library

A modal dialog that allows the user to select one or more My webMethods Server resources (folders, pages, portlets, and so forth).



Note: This control is not portable to third-party portals. You can use it only in portlets published to instances of My webMethods Server.

The selected resources are referenced by the control's "targetValue" property. For a single-select picker, the "targetValue" property can be bound to the following data types:

- com.webmethods.caf.faces.data.portal.PortalItemModel (a model object representing the MWS resource)
- com.webmethods.portal.system.IURI (the My webMethods Server ID of the resource)
- java.lang.String (a string representation of the My webMethods Server ID of the resource)

For a multi-select picker, the "targetValue" property can be bound to a list or array of the above types.

Like other "hideable" controls, this control can be toggled between visible and hidden through client-side JavaScript code. Special "toggle" controls encapsulate this code within controls that can be created and configured visually, using the webMethods CAF. See "Hideable Controls" on page 194 for more information about hideable controls. See "Toggle Controls" on page 197 for more information about toggle controls.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to include control in rendered page (note that setting this to false will prevent the control from being toggled visible).	true
refreshOnApply	Refresh On Apply	Control ID Reference	ID of control to refresh when user selects principals (defaults to this control).	
Display Properties				
userSearch Allowed	User Search Allowed	boolean	True to allow selecting users.	true
groupSearch Allowed	Group Search Allowed	boolean	True to allow selecting groups.	true
roleSearch Allowed	Role Search Allowed	boolean	True to allow selecting roles.	true

Name	Display Name	Туре	Description	Default
Action Properties				
applyButton Action	Apply Action	Method Binding	Action to invoke when selects principals.	user
Value Properties				
multiple Selections Allowed	Multiple	boolean	True to allow selecting multiple principals.	true
rootURI	Root Container	String	MWS ID of root of the resource tree to display in the dialog.	folder. public
targetValue	Target Value Binding	Object	Selected principals.	
Lazy Load Propertie	?S			
lazy	Lazy Load	boolean	True to load control's content asynchronously on demand.	false
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog.	0
progressMsg	Progress Message	String	Message to display in progress dialog.	Defaults to localized message; in English, "Loading "
refreshOnShow	Refresh On Show	boolean	True to refresh content every time control is toggled visible; false to refresh content only if control is initially hidden and then toggled visible.	false

Name	Display Name	Туре	Description	Default		
suppressInputs	Suppress Inputs	Collection	List of ids of controls for which to disable validation/update processing when handling an asynchronous request (comma separated).			
twoPass	Two Pass	boolean	True to load content asynchronously, even if control is initially visible; false to render control synchronously if control is initially visible.	false		
Expert Display Properties						
progressFlash OnLoad	Flash On Load	boolean	True to flash control when content finishes loading asynchronously.	true		

None.

Facets

None.

Principal Picker Dialog

com.webmethods.caf.faces.panel.PrincipalPickerModalDialog

MWS Control Component Library

A modal dialog that allows the user to select one or more principals (users, groups, and roles).



Note: This control is not portable to third-party portals. You can use it only in portlets published to instances of My webMethods Server.

The selected principals are referenced by the control's "targetValue" property. For a single-select picker, the "targetValue" property can be bound to the following data types:

- com.webmethods.caf.faces.data.dir.IPrincipalProvider (a model object representing the principal)
- com.webmethods.portal.system.IURI (the My webMethods Server ID of the principal)
- java.lang.String (a string representation of the My webMethods Server ID of the principal)

For a multi-select picker, the "targetValue" property can be bound to a list or array of the above types.

You can use the com.webmethods.caf.faces.data.dir.PrincipalModelFactory and com.webmethods.caf.faces.data.dir.PrincipalModelList classes to help create individual principal model objects or lists of principal model objects.

Like other "hideable" controls, this control can be toggled between visible and hidden through client-side JavaScript code. Special "toggle" controls encapsulate this code within controls that can be created and configured visually, using the webMethods CAF. See "Hideable Controls" on page 194 for more information about hideable controls. See "Toggle Controls" on page 197 for more information about toggle controls.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to include true control in rendered page (note that setting this to false will prevent the control from being toggled visible).	
refreshOnApply	Refresh On Apply	Control ID Reference	ID of control to refresh when user selects principals (defaults to this control).	
Display Properties				
userSearch Allowed	User Search Allowed	boolean	True to allow selecting users.	true
groupSearch Allowed	Group Search Allowed	boolean	True to allow selecting groups.	true

Name	Display Name	Туре	Description	Default	
roleSearch Allowed	Role Search Allowed	boolean	True to allow selecting roles.	true	
Action Properties					
applyButton Action	Apply Action	Method Binding	Action to invoke when user selects principals.		
Value Properties					
multiple Selections Allowed	Multiple	boolean	True to allow selecting multiple principals.	true	
targetValue	Target Value Binding	Object	Selected principals.		
Lazy Load Properties					
lazy	Lazy Load	boolean	True to load control's content asynchronously on demand.	false	
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog.	0	
progressMsg	Progress Message	String	Message to display in progress dialog.	Defaults to localized message; in English, "Loading"	
refreshOnShow	Refresh On Show	boolean	True to refresh content every time control is toggled visible; false to refresh content only if control is initially hidden and then toggled visible.	false	
suppressInputs	Suppress Inputs	Collection	List of ids of controls for which to disable validation/update processing when handling an asynchronous request (comma separated).		

Name	Display Name	Туре	Description	Default
twoPass	Two Pass	boolean	True to load content asynchronously, even if control is initially visible; false to render control synchronously if control is initially visible.	false
Expert Display Prop	perties			
progressFlash OnLoad	Flash On Load	boolean	True to flash control when content finishes loading asynchronously.	true
Children				

None.

Facets

None.

Progress Dialog

com.webmethods.caf.faces.panel.ProgressDialog

Base Controls Component Library

A progress bar dialog that overlays another panel (or the entire page) when toggled visible through client-side JavaScript code. See "Hideable Controls" on page 194 for more information about hideable controls.

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	ID of control to overlay (or empty to page).	overlay entire
id	ID	String	Control ID.	

Name	Display Name	Туре	Description	Default
rendered	Rendered	boolean	True to include control in rendered page (note that setting this to false will prevent the control from being toggled visible).	true
Display Properties				
message	Message	String	Progress message to display.	Processing
Expert Display Properti	es			
centered	Centered	double	True to center content.	true
hideEffect	Hide Effect	String	Name of "Scriptaculous Effects" on page 198 to use to hide this control.	Effect.CAF.Fade
hideEffectOptions	Hide Effect Options	String	Custom options to pass to the hide effect, in JavaScript object notation (for example, "duration: 2.0, queue: 'front'").	
opacity	Opacity	double	Opacity of panel and contents, from 0.0 (completely transparent) to 1.0 (completely opaque).	1.0
showEffect	Show Effect	String	Name of "Scriptaculous Effects" on page 198 to use to show this control.	Effect.CAF.App ear
showEffectOptions	Show Effect Options	String	Custom options to pass to the show effect, in JavaScript object notation (for example, "duration: 2.0, queue: 'front'").	

None.

Facets

Chapter 14. Input Controls

Autocomplete Text Input	90
Checkbox	72
Checkbox Group	94
Combobox) 7
Date Input	99
Date Range Input)1
Dropdown)5
Extended Select-Many Listbox)8
Extended Select-One Listbox	10
File Input	12
Filter Input	14
Hidden Input	17
HTML Input	18
In Place Text Input	20
Multi-Line Text Input	22
Option	24
Option Group	26
Radio Button Group	30
Secret Input	33
Select-Many Listbox	35
Select-One Button	38
Select-One Link	40
Select-One Listbox	12
Select-One Tabs	45
Simple Schedule Input	17
Swapbox	19
Text Input	51

Autocomplete Text Input

com.webmethods.caf.faces.select.AutocompleteTextInput

Base Controls Component Library

Text-input with an auto-complete popup menu. Abstract behavior is equivalent to other single-select controls, such as a "Dropdown" on page 305. Selection options are specified by means of javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children.

If this control has a single Option Group, bound to a provider that implements the com.webmethods.caf.faces.data.IFilterableSelectItemGroupProvider, and the provider's isFilterable() method returns a value of true, filtering of the list of auto-completion values is done on the server, by the provider, using the provider's filter property value to filter the collection of items returned by its getSelectItemContentProviders() method. Note that the default ISelectItemGroupProvider implementation,

com.webmethods.caf.faces.data.object.DefaultSelectItemContentProvider, implements IFilterableSelectItemGroupProvider, but by default returns false for the isFilterable() method.

If the control is not bound to a provider that implements

IFilterableSelectItemGroupProvider, then auto-completion is done only on the client, using the standard commonsearch keyword rules (* = wildcard, space = and, quotes = exact phrase).

You can set the default filter for this control using this Control's Filter property value.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Properties

Name	Display Name	Туре	Description Choices	Default
General Properties				
id	ID	String	Control id.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" and uneditable.	false
readonly	Read Only	boolean	True to display as un-editable.	false

Name	Display Name	Туре	Description	Choices	Default
width	Width	String	Pre-defined input width.	input2 (~2 characters vinput4, input6, input8 input10, input20, input10 input40, input50 (~50 characters wide)	,
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instance t value to and from a	o use to convert the con String.	trol's
immediate	Immediate	boolean	the Apply Request \	eChangeListeners in Values phase; false to ss Validations phase.	false
other	Other	boolean	True to allow users t than the specified va		false
required	Required	boolean	True to display the o	ontrol as "required."	false
validator	Validator	javax.faces.valid ator.Validator	Validator instance to submitted value.	use to validate the cont	rol's
value	Value	Object	Current selection va	lue.	
valueChange Listener	Value Change Listener	javax.faces.el.Me thodBinding	Value change callba	ck.	
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcut le	etter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RT (right-to-left)	L
lang	Language	String	Language code for c	ontent.	
style	CSS Style	String	HTML "style" attrib	ute value.	
styleClass	CSS Class	String	HTML "class" attrib	ute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds an item to the tab sequence; a negative integer removes an item.		
title	Title	String	HTML "title" attributooltip).	ıte value (often displaye	d as a
Expert Value Prope	erties				
filter	Filter	String	Filter value.		

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Checkbox

javax.faces.HtmlSelectBooleanCheckbox

CAF JSF Base Controls Component Library

Single checkbox. The model is java.lang.Boolean (true if checked). A standard JavaServer Faces control.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Standard JSF Properties

Name	Display Name	Туре	Description Choices	Default	
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control.	true	
Display Properties					
disabled	Disabled	boolean	True to display as "grayed-out" and uneditable.		
readonly	Read Only	boolean	True to display as un-editable.	false	
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instance to use to convert the control's value to and from a String.		
immediate	Immediate	boolean	True to process ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		
required	Required	boolean	True to display control as "required".	false	
selected	Selected	boolean	True if checkbox is checked.		

Name	Display Name	Туре	Description	Choices	Default
validator	Validator	javax.faces. validator. Validator	Validator instand submitted value	ce to use to validate the control	s
value	Value	Object	Control value.		
valueChange Listener	Value Change Listener	javax.faces.el. Method Binding	Value change cal	llback.	
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shorte	rut letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL (right left)	t-to-
lang	Language	String	Language code f	or content.	
style	CSS Style	String	HTML "style" at	tribute value.	
styleClass	CSS Class	String	HTML "class" at	ttribute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" att tooltip.	ribute value (often displayed a	s a
Expert Client Side E	vents				
onblur	Blur	script	Control losing for	ocus.	
onfocus	Focus	script	Control gaining	focus.	
onchange	Change	script	Control value ch	anged.	
ondblclick	Double Click	script	Control double-	clicked.	
onkeydown	Key Down	script	Key pressed who	en control is focused.	
onkeypress	Key Press	script	Key pressed and	released when control is focus	ed.
onkeyup	Key Up	script	Key released wh	en control is focused.	
onmousedown	Mouse Down	script	Mouse button pr	ressed when over control.	
onmousemove	Mouse Move	script	Mouse moved w	hen over control.	
onmouseout	Mouse Out	script	Mouse moved or	ut of control boundaries.	
onmouseover	Mouse Over	script	Mouse moved in	to control boundaries.	
onmouseup	Mouse Up	script	Mouse button re	leased when over control.	
ondblclick	Double Click	script	Control double-	clicked.	

Extended CAF Properties

Name	Display Name	Туре	Description	Choices	Default
Display Properties					
label	Label	String	Checkbox label.		
	Children				
	None.				
	Standard JSF Facets				
	None.				

Checkbox Group

None.

javax.faces.HtmlSelectManyCheckbox

Extended My webMethods Facets

CAF JSF Base Controls Component Library

A group of checkboxes. Selection options are specified means of javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children. Note that the control's value (the currently selected options' values) is an array of selected values. A standard JavaServer Faces control.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Standard JSF Properties

Name	Display Name	Туре	Description	Choices	Default				
General Properties	General Properties								
id	ID	String	Control ID.						
rendered	Rendered	boolean	True to display control.		true				
Display Properties	6								
disabled	Disabled	boolean	True to display a editable.	s "grayed-out" and un-	false				

Name	Display Name	Туре	Description	Choices	Default
layout	Layout	String	Layout orientation.	lineDirection (horizontal), pageDirection (vertical)	lineDirection
readonly	Read Only	boolean	True to display a	s un-editable.	false
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instan and from a String	ce to use to convert the cong.	trol's value to
immediate	Immediate	boolean	the Apply Reque	ValueChangeListeners in est Values phase; false to ocess Validations phase.	false
required	Required	boolean	True to display c	ontrol as "required".	false
validator	Validator	javax.faces. validator. Validator	Validator instand value.	e to use to validate the cont	rol's submitted
value	Value	Object	Current selection	n values (as an array of valu	es).
valueChange Listener	Value Change Listener	javax.faces.el. Method Binding	Value change cal	lback.	
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortc	ut letter.	
border	Border	String	Border width in	pixels.	0
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL (right-to-left)	
disabledClass	Disabled CSS Class	String	CSS class to appl	y to disabled options.	
enabledClass	Enabled CSS Class	String	CSS class to appl	y to enabled options.	
lang	Language	String	Language code f	or content.	
style	CSS Style	String	HTML "style" at	tribute value.	
styleClass	CSS Class	String	HTML "class" at	tribute value.	
tabindex	Tab Index	int		r) sequence, from 1 to n. Zer gative integer removes item	
title	Element Title	String	HTML "title" att	ribute value (often displaye	ed as a tooltip.

Name	Display Name	Туре	Description	Choices	Default			
Expert Client Side Events								
onblur	Blur	script	Control losing fo	ocus.				
onfocus	Focus	script	Control gaining	focus.				
onclick	Click	script	Control clicked.					
onchange	Change	script	Control value changed.					
ondblclick	Double Click	script	Control double-clicked.					
onkeydown	Key Down	script	Key pressed whe	en control is focused.				
onkeypress	Key Press	script	Key pressed and	released when control is i	focused.			
onkeyup	Key Up	script	Key released wh	en control is focused.				
onmousedown	Mouse Down	script	Mouse button pr	essed when over control.				
onmousemove	Mouse Move	script	Mouse moved w	hen over control.				
onmouseout	Mouse Out	script	Mouse moved or	ut of control boundaries.				
onmouseover	Mouse Over	script	Mouse moved in	to control boundaries.				
onmouseup	Mouse Up	script	Mouse button re	leased when over control.				

Extended CAF Properties

Name	Display Name	Туре	Description	Choices	Default			
Display Properties								
width	Width	String	Pre-defined input width.	input2 (~2 characters wide), input4, input6, input8, input10, input20, input30, input40, input50 (~50 characters wide)				
Value Properties	S							
other	Other	boolean	True to allow users to enter values other than the specified values.		false			
sortByLabel	Sort By Label	boolean	True to sort options alp	True to sort options alphabetically by label text.				

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Standard JSF Facets

None.

Extended CAF Facets

None.

Combobox

com.webmethods.caf.faces.select.SelectOneCombobox

Base Controls Component Library

A single-select combination dropdown and text-input control. Selection options are specified means of javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Properties

Name	Display Name	Туре	e Description Choices		Default
General Properties					
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display contro	1.	true
Display Properties					
disabled	Disabled	boolean	True to display as "grayed-out" and uneditable.		false
readonly	Read Only	boolean	True to display as un-	editable.	false
width	Width	String	Pre-defined input width.	efined input input2 (~2 characters wid	
Value Properties					
converter	Converter	javax.faces.conver t.Converter	Converter instance to use to convert the control's value to and from a String.		

Name	Display Name	Туре	Description	Choices	Default
immediate	Immediate	boolean	True to process Value the Apply Request Va process in the Process	lues phase; false to	false
other	Other	boolean	True to allow users to than the specified value		false
required	Required	boolean	True to display contro	ol as "required."	false
sortByLabel	Sort By Label	boolean	True to sort options al label text.	phabetically by	false
validator	Validator	javax.faces.validat or.Validator	Validator instance to use to validate the control's submitted value.		
value	Value	Object	Current selection value.		
valueChange Listener	Value Change Listener	javax.faces.el.Met hodBinding	Value change callback.		
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcut let	ter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), (right-to-left)	RTL
lang	Language	String	Language code for co	ntent.	
style	CSS Style	String	HTML "style" attribu	te value.	
styleClass	CSS Class	String	HTML "class" attribu	te value.	
tabindex	Tab Index	int	The order in tab (key) sequence, from 1 to n. Zero adds an item to the tab sequence, a negative integer removes an item.		
title	Title	String	HTML "title" attribut tooltip).	e value (often display	red as a

javax.faces. Select
Item ("Option" on page 324) and javax.faces. Select
Items ("Option Group" on page 326) $\,$

Facets

None.

Date Input

com.webmethods.caf.faces.input.DateInput

Base Controls Component Library

A date input. The "value" property of this control must be a java.util.Date object.

The "value" property also may be bound to a java.lang.String — if the "valuePattern" property is set to the pattern which describes the value's format. This pattern must use the syntax described by the java.text.SimpleDateFormat Javadocs.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description Choices	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" and uneditable.	false
pattern	Pattern	String	Date display pattern (see java.text.SimpleDateFormat Javadocs).	Default ("") is current user's preferred date input pattern.
readonly	Read Only	boolean	True to display as un-editable.	false
showSeconds	Show Seconds	boolean	True to display seconds input (in addition to hour and minutes input).	false
showTime	Show Time	boolean	True to display time input (in addition to date input).	true
showTimezone	Show Time Zone	boolean	True to display time in the user's time zone with a time zone label; false to display time in UTC without a label.	true

Name	Display Name	Туре	Description	Choices	Default
width	Width	String	Pre-defined input width.	input2 (~2 characters wide), input4, input6, input8, input10, input20, input30, input40, input50 (~50 characters wide)	input6
Value Properties					
converter	Converter	javax.faces.con vert.Converter	Not used (date is automatically converted).		
immediate	Immediate	boolean	True to process ValueChangeLis teners in the Apply Request Values phase; false to process in the Process Validations phase.		false
required	Required	boolean	True to display control as "required".		false
validator	Validator	javax.faces.vali dator.Validator	Validator instance to use to validate the control's submitted value.		
value	Value	java.util.Date	Control value.		
valueChange Listener	Value Change Listener	javax.faces.el. MethodBindin g	Value change callback.		
valuePattern	Value Pattern	String	is bound to a strin	parse and format a value, g instead of a java.util.DoleDateFormat Javadocs	ate object
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shortcut letter.		

Name	Display Name	Туре	Description	Choices	Default
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attribute value.		
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
	Children				

None.

Facets

None.

Date Range Input

com.webmethods.caf.faces.input.DateRangeInput

Base Controls Component Library

A date-range input. The "value" property of this control must be a com.webmethods.caf.faces.data.object.DateRange object. A DateRange object has three properties, "fixedRange," "relativeRange," and "date." A DateRange can specify one of the following:

A fixed start and end date (ex "2000-01-01 to 2000-02-01"):

This property	Specifies this
fixedRange	A period of time in milliseconds
relativeRange	DateRange.FIXED
date	A start date (for a positive period of time; an end date for a negative period of time)

■ A fixed period of time, relative to the current date (ex "last thirty days")

This property	Specifies this
fixedRange	A period of time in milliseconds
relativeRange	DateRange.FIXED
date	Null

A relative period time, relative to the current date (ex "this month")

This property	Specifies this
fixedRange	0
relativeRange	A period of time (an enumerated value)
date	Null
This property	Specifies this
This property fixedRange	Specifies this 0
	· ·

The DateRange class supports the following "relativeRange" property values:

- DateRange.FIXED: Not a relative-range (range specified in milliseconds by "fixedRange" property).
- DateRange.THIS_DAY: Midnight this morning to the second before midnight tonight.
- DateRange.PREVIOUS_DAY: Midnight yesterday morning to the second before midnight last night.
- DateRange.NEXT_DAY: Midnight tomorrow morning to the second before midnight tomorrow night.
- DateRange.THIS_WEEK: Midnight on the first morning of the week (Sunday in the US) to the second before midnight on the last night of the week (Saturday in the US).
- DateRange.PREVIOUS_WEEK: Midnight on the first morning of the previous week (Sunday in the US) to the second before midnight on the last night of the previous week (Saturday in the US).
- DateRange.NEXT_WEEK: Midnight on the first morning of next week (Sunday in the US) to the second before midnight on the last night of next week (Saturday in the US).

- DateRange.THIS_MONTH: Midnight on the first of the month to the second before midnight on the last night of the month (28th, 29th, 30th, or 31st).
- DateRange.PREVIOUS_MONTH: Midnight on the first of the previous month to the second before midnight on the last night of previous month (28th, 29th, 30th, or 31st).
- DateRange.NEXT_MONTH: Midnight on the first of next month to the second before midnight on the last night of next month (28th, 29th, 30th, or 31st).
- DateRange.THIS_YEAR: Midnight on January 1 of this year to the second before midnight on December 31 of this year.
- DateRange.PREVIOUS_YEAR: Midnight on January 1 of the previous year to the second before midnight on December 31 of the previous year.
- DateRange.NEXT_YEAR: Midnight on January 1 of next year to the second before midnight on December 31 of next year.

A DateRange object can calculate the appropriate start and end dates for its date-range, via its calculateStart() and calculateEnd() methods. For an infinite range, calculateStart() and calculateEnd() will both return null. Otherwise, calculateStart() will always be less than or equal to the calculateEnd().

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Properti	ies				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display	y control.	true
Display Propertion	es				
disabled	Disabled	boolean	True to display un-editable.	y as "grayed-out" and	false
futureRanges	Future Ranges	boolean	True to show of from the futur	options for date ranges e.	false
pastRanges	Past Ranges	boolean	True to show of from the past.	options for date ranges	true
label	Label	String	Range dropdo	wn label.	Default ("") is localized label; in English, "Range".

Name	Display Name	Туре	Description	Choices	Default
layout	Layout	String	Picker layout style.	full, compact, collapsed	full
pattern	Pattern	String	Date display p java.text.Simpl Javadocs).		Default ("") is current user's preferred date input pattern.
readonly	Read Only	boolean	True to display	y as un-editable.	false
showTime	Show Time	boolean	True to display addition to da	y time input (in te input).	true
showTimezo ne	Show Time Zone	boolean	time zone with	True to display time in the user's time zone with a time zone label; false to display time in UTC without a label.	
width	Width	String	Pre-defined input width.	input2 (~2 characters wide), input4, input6, input8, input10, input20, input30, input40, input50 (~50 characters wide)	input6
Value Properties					
converter	Converter	javax.faces.convert. Converter	Not used (date	e-range is automatically	converted).
immediate	Immediate	boolean	Request Value	s Listeners in the Apply s phase; false to Process Validations	false
required	Required	boolean	True to display "required".	y control as	false
validator	Validator	javax.faces.validator. Validator		nce to use to validate ubmitted value.	
value	Value	com.webmethods.caf. faces.data.object. DateRange	Control value.		

Name	Display Name	Туре	Description	Choices	Default
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change o	callback.	
Expert Display P	roperties				
accesskey	Access Key	String	Keyboard sho	tcut letter.	
collapse	Collapse	boolean	True to display	initially as collapsed.	false
style	CSS Style	String	HTML "style"	attribute value.	
styleClass	CSS Class	String	HTML "class"	attribute value.	
tabindex	Tab Index	int	to n. Zero add	sey) sequence, from 1 s item to tab sequence, er removes item.	
	Children				

None.

Facets

None.

Dropdown

com.webmethods.caf.faces.toggle.ToggleMenu

CAF JSF Base Controls Component Library

A single-select dropdown. Selection options are specified via "Option" on page 324 and "Option Group" on page 326 children. A standard JavaServer Faces control.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Standard JSF Properties

Name	Display Name	Туре	Description Choices	Default
General Propert	ies			
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description	Choices	Default
Display Properties	;				
disabled	Disabled	boolean	True to display as editable.	"grayed-out" and un-	false
readonly	Read Only	boolean	True to display as un-editable. false		false
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance value to and from	e to use to convert the co a String.	ntrol's
immediate	Immediate	boolean	the Apply Reques	alueChangeListeners in to Values phase; false to cess Validations phase.	false
required	Required	boolean	True to display co	ntrol as "required".	false
validator	Validator	javax.faces.validator. Validator	Validator instance submitted value.	e to use to validate the co	ntrol's
value	Value	Object	Current selection	value.	
valueChangeLi stener	Value Change Listener	javax.faces.el.Method Binding	Value change call	back.	
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcu	t letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTI to-left)	L (right-
disabledClass	Disabled CSS Class	String	CSS class to apply	to disabled options.	
enabledClass	Enabled CSS Class	String	CSS class to apply	to enabled options.	
lang	Language	String	Language code fo	r content.	
style	CSS Style	String	HTML "style" att	ribute value.	
styleClass	CSS Class	String	HTML "class" att	ribute value.	
tabindex	Tab Index	int		sequence, from 1 to n. Zence, negative integer rem	
title	Element Title	String	HTML "title" attr tooltip.	ibute value (often displa	yed as a

Name	Display Name	Туре	Description	Choices	Default	
Expert Client Side	Events					
onblur	Blur	script	Control losing focus.			
onfocus	Focus	script	Control gaining for	ocus.		
onclick	Click	script	Control clicked.			
onchange	Change	script	Control value cha	nged.		
ondblclick	Double Click	script	Control double-clicked.			
onkeydown	Key Down	script	Key pressed when control is focused.			
onkeypress	Key Press	script	Key pressed and	released when control	is focused.	
onkeyup	Key Up	script	Key released whe	n control is focused.		
onmousedown	Mouse Down	script	Mouse button pressed when over control.			
onmousemove	Mouse Move	script	Mouse moved wh	nen over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.			
onmouseover	Mouse Over	script	Mouse moved into control boundaries.			
onmouseup	Mouse Up	script	Mouse button relection control.	eased when over		

Extended CAF Properties

Name	Display Name	Туре	Description	Choices	Default
Display Propert	ies				
width	Width	String	Pre-defined input width.	input2 (~2 characters v input4, input6, input8, input20, input30, inpu input50 (~50 character	input10, t40,
Value Properties	S				
other	Other	boolean	True to allow users to enter specified values.	r values other than the	false
sortByLabel	Sort By Label	boolean	True to sort options alphab	etically by label text.	false

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Standard JSF Facets

None.

Extended CAF Facets

None.

Extended Select-Many Listbox

com.webmethods.caf.faces.select.SelectManyListbox

Base Controls Component Library

A multi-select listbox that displays icons, descriptions, and extra-wide titles. Selection options are specified via "Option" on page 324 and "Option Group" on page 326 children. Note that the control's value (the currently selected options' values) is an array of selected values.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Properties

Name	Display Name	Туре	Description	Choices	Default
General Propert	ties				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control		true
Display Propert	ies				
disabled	Disabled	boolean	True to display as "grayed-out" and un-editable.		false
readonly	Read Only	boolean	True to display as un-e	ditable.	false
width	Width	String	Pre-defined input width.	input2 (~2 characters wide), input4, input6, input8, input10, input20, input30, input40, input50 (~50 characters wide)	

Name	Display Name	Туре	Description	Choices	Default
Value Properties	;				
converter	Converter	javax.faces.convert. Converter	Converter instance to use to convert the control's value to and from a String.		
immediate	Immediate	boolean	True to process false ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		false
other	Other	boolean	True to allow users to other than the specifie		false
required	Required	boolean	True to display contro	l as "required".	false
sortByLabel	Sort By Label	boolean	True to sort options all label text.	phabetically by	false
validator	Validator	javax.faces.validator. Validator	Validator instance to use to validate the control's submitted value.		
value	Value	Object	Current selection values (as an array of values).		
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change callback		
Expert Display P	Properties				
accesskey	Access Key	String	Keyboard shortcut lett	ter.	
dir	Directionalit y	String	Text direction hint.	LTR (left-to-rig (right-to-left)	ght), RTL
lang	Language	String	Language code for cor	ntent.	
size	Size	String	Height in lines.		
style	CSS Style	String	HTML "style" attribut	e value.	
styleClass	CSS Class	String	HTML "class" attribut	te value.	
tabindex	Tab Index	int	Order in tab (key) sequence, r		
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip.		
Expert Propertie	S				
filter	Filter	String	A filter expression. Us filterable data models. page 314.		

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Extended Select-One Listbox

com.webmethods.caf.faces.select.SelectOneListbox

Base Controls Component Library

A single-select listbox that displays icons, descriptions, and extra-wide titles. Selection options are specified via "Option" on page 324 and "Option Group" on page 326 children.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Properties

Name	Display Name	Туре	Description	Choices	Default
General Propert	ies				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control	l.	true
Display Properti	ies				
disabled	Disabled	boolean	True to display as "grayed-out" and un-editable.		false
readonly	Read Only	boolean	True to display as un-editable.		false
width	Width	String	Pre-defined input input2 (~2 chara input4, input6, input10, input2 input40, input5 characters wide		, input8, 20, input30, 50 (~50
Value Properties	S				
converter	Converter	javax.faces.convert. Converter	Converter instance to use to convert the control's value to and from a String.		
immediate	Immediate	boolean	True to process ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		false

Name	Display Name	Туре	Description	Choices	Default
other	Other	boolean		True to allow users to enter values false other than the specified values.	
required	Required	boolean	True to display control	as "required".	false
sortByLabel	Sort By Label	boolean	True to sort options alplabel text.	phabetically by	false
validator	Validator	javax.faces.validator. Validator	Validator instance to u submitted value.	se to validate the	e control's
value	Value	Object	Current selection value	es (as an array of	values).
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change callback.		
Expert Display P	Properties				
accesskey	Access Key	String	Keyboard shortcut lett	er.	
dir	Directionalit y	String	Text direction hint.	LTR (left-to-rig (right-to-left)	ght), RTL
lang	Language	String	Language code for con	tent.	
size	Size	String	Height in lines.		
style	CSS Style	String	HTML "style" attribute	e value.	
styleClass	CSS Class	String	HTML "class" attribute	e value.	
tabindex	Tab Index	int	Order in tab (key) sequitem to tab sequence, n		
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip.		
Expert Propertie	S				
filter	Filter	String	A filter expression. Use filterable data models. page 314.		

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

File Input

com.webmethods.caf.faces.input.File

Base Controls Component Library

A file upload control. The model for this control (the type of object to which you bind the control's "value" property) is a org.apache.commons.fileupload.FileItem.

For a FileItem object, you can get the file's content-type via the getContentType() method, the file's original name (from the client) via the getName() method, and the file's full content via the getInputStream() method. When finished with a FileItem, call its delete() method to release its associated resources (such as its temp file on disk or bytes in memory).

You cannot drop an input control outside of a Form control ("Form" on page 268).



Note: Users cannot upload files with asynchronous commands.

Name	Display Name	Туре	Description	Choices	Default
General Propertie	s				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display co	ontrol.	true
Display Properties	S				
disabled	Disabled	boolean	True to display as editable.	s "grayed-out" and un-	
readonly	Read Only	boolean	True to display as un-editable.		
width	Width	String	Pre-defined input width.	input2 (~2 characters input4, input6, input8 input10, input20, input10, input50 (~50 characters wide)	3,
Value Properties					
converter	Converter	javax.faces.convert. Converter	Not used.		
immediate	Immediate	boolean	the Apply Reques	alueChangeListeners in st Values phase; false to ocess Validations phase.	false

Name	Display Name	Туре	Description	Choices	Default
required	Required	boolean	True to display co	ntrol as "required".	false
validator	Validator	javax.faces.validator. Validator	Validator instance control's submitte	to use to validate the d value.	
value	Value	org.apache.commons .fileupload.FileItem	Control value.		
valueChangeLi stener	Value Change Listener	javax.faces.el.Method Binding	Value change call	oack.	
Expert Display Pro	perties				
accept	Accept	String		content-types, in CSV most browsers ignore	
accesskey	Access Key	String	Keyboard shortcu	t letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTI to-left)	_ (right-
lang	Language	String	Language code for	r content.	
style	CSS Style	String	HTML "style" attr	ribute value.	
styleClass	CSS Class	String	HTML "class" attr	ribute value.	
tabindex	Tab Index	int		sequence, from 1 to n. Zecce, negative integer remo	
title	Element Title	String	HTML "title" attritooltip.	bute value (often display	ed as a
Expert Client Side	Events				
onblur	Blur	script	Control losing foc	us.	
onfocus	Focus	script	Control gaining for	ocus.	
onclick	Click	script	Control clicked.		
onchange	Change	script	Control value char	nged.	
ondblclick	Double Click	script	Control double-cli	cked.	
onkeydown	Key Down	script	Key pressed when	control is focused.	
onkeypress	Key Press	script	Key pressed and r	eleased when control is	focused.
onkeyup	Key Up	script	Key released when	n control is focused.	
onmousedown	Mouse Down	script	Mouse button pre- control.	ssed when over	

Name	Display Name	Туре	Description	Choices	Default
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved when over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		

None.

Facets

None.

Filter Input

com. we bmethods. caf. faces. select. Filter Input

Base Controls Component Library

A standard input control for filtering a specified select or table control.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Select Filtering

The following select controls are filterable:

- "Autocomplete Text Input" on page 290
- "Combobox" on page 297
- "Extended Select-Many Listbox" on page 308

- "Extended Select-One Listbox" on page 310
- "Swapbox" on page 349

As the Combobox and the Autocomplete Text Input controls themselves allow users to type in an autocompleted value, you typically would not use a separate Filter Input control with these two controls (although you might use an IFilterableSelectItemGroupProvider provider with these controls as described below).

If a select control has a single "Option Group" on page 326, bound to a provider that implements the com.webmethods.caf.faces.data.IFilterableSelectItemGroupProvider, and the provider's isFilterable() method returns a value of true, the filtering is done on the server, by the provider, using the provider's filter property value to filter the collection of items returned by its getSelectItemContentProviders() method. Note that the default ISelectItemGroupProvider implementation,

com.webmethods.caf.faces.data.object.DefaultSelectItemContentProvider, implements IFilterableSelectItemGroupProvider, but by default returns a value of false for the isFilterable() method.

If a select control is not bound to a provider that implements IFilterableSelectItemGroupProvider, the filtering is done on the client, using the standard commonsearch keyword rules (* = wildcard, space = and, quotes = exact phrase).

Table Filtering

If a table control is bound to a

com.webmethods.caf.faces.data.IFilterableTableContentProvider, and the provider's isFilterable() method returns a value of true, the filtering is done on the server, by the provider, using the provider's filter property value to filter the rows over which the provider can iterate. With the default IFilterableTableContentProvider implementations, com.webmethods.caf.faces.data.object.FilterableListTableContentProvider and com.webmethods.caf.faces.data.object.FilterableSelectableListTableContentProvider, you must initialize the provider with a binding expression to use to calculate the value to filter on for each row. For example, you might use a filter value binding of "#{row.title} #{row.description}" to filter on the title and description fields of each row.

If a table control is not bound to an IFilterableTableContentProvider that is filterable, then filtering is done on the client (over the entire textual content of the row), using the standard commonsearch keyword rules (* = wildcard, space = and, quotes = exact phrase).

Select and Table Filtering

You can set the default filter for a table or select control by means of the select control's Filter property value.

Note that you can also use the CAF ClientSideModel to invoke filtering on a table or select control. See "Client-Side Model" on page 201 for more information.

Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	Control ID Reference	Table or select control	to filter.	
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display contro	1.	true
Display Properties					
disabled	Disabled	boolean	True to display as "graeditable.	ayed-out" and un-	false
readonly	Read Only	boolean	True to display as un-	editable.	false
showGoButton	Show Go Button	boolean	True to show Go Button (filter when user clicks Go); false to not show Go Button (filter as user types).		false
width	Width	String	Pre-defined input input2 (~2 characters width. wide), input4, input6, input8, input10, input2(input30, input40, input40 (~50 characters wide)		it6, put20, nput50
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shortcut let	ter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), (right-to-left)	RTL
lang	Language	String	Language code for cor	ntent.	
style	CSS Style	String	HTML "style" attribut	te value.	
styleClass	CSS Class	String	HTML "class" attribut	te value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds an item to a tab sequence, a negative integer removes an item.		
title	Title	String	HTML "title" attribute tooltip).	e value (often display	ed as a

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Hidden Input

javax.faces.hidden

CAF JSF Base Controls Component Library

A hidden input field. A standard JavaServer Faces control.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Properties	5				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display contr	ol.	true
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance to value to and from a S	o use to convert the con String.	ntrol's
immediate	Immediate	boolean	True to process Value the Apply Request V process in the Proces	alues phase; false to	false
required	Required	boolean	True to display control as "required".	false	
validator	Validator	javax.faces.validator. Validator	Validator instance to submitted value.	use to validate the cor	ntrol's
value	Value	Object	Control value.		
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change callbac	k.	

None.

Facets

None.

HTML Input

com. we bmethods. caf. faces. input. HTMLInput

Base Controls Component Library

A WYSIWYG HTML editor.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Propertie	S				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cor	ntrol.	true
Display Properties	5				
advanced Editor	Advanced Editor	boolean	True to use editor with lots and lots of buttons.		false
disabled	Disabled	boolean	True to display as "grayed-out" and uneditable.		false
readonly	Read Only	boolean	True to display as	un-editable.	false
width	Width	String	Pre-defined input input2 (~2 characters with input4, input6, input8, input10, input20, input30 input40, input50 (~50 characters wide)		8, ut30,
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance to use to convert the control's value to and from a String.		
immediate	Immediate	boolean	True to process ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		false

Name	Display Name	Туре	Description	Choices	Default
required	Required	boolean	True to display control as "required".	false	
validator	Validator	javax.faces.validator. Validator	Validator instance submitted value.	to use to validate the	control's
value	Value	Object	Control value.		
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change callb	pack.	
Expert Display Pr	operties				
accesskey	Access Key	String	Keyboard shortcu	t letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), F (right-to-left)	RTL
lang	Language	String	Language code for	content.	
style	CSS Style	String	HTML "style" attr	ibute value.	
styleClass	CSS Class	String	HTML "class" attr	ibute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attribute value (often displayed as tooltip.		

None.

Facets

None.

In Place Text Input

com. we bmethods. caf. faces. input. In Place Text

Base Controls Component Library

A single-line text input that is displayed like a static text output until clicked.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Properti	es				
doubleClick	Double Click	boolean	True to make editable with a double-click, false to make editable with a single-click.		false
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	True to display control.	
Display Propertie	es				
disabled	Disabled	boolean	True to display as editable.	"grayed-out" and un-	false
readonly	Read Only	boolean	True to display as un-editable.		false
width	Width	String	Pre-defined input input2 (~2 characters width. input4, input6, input10, input20, input40, input50 (~50 characters wide)		ut8, iput30,
Value Properties					
immediate	Immediate	boolean	True to process ValueChangeListeners fa in the Apply Request Values phase; false to process in the Process Validations phase.		false
required	Required	boolean	True to display control as "required".		false
selected	Selected	boolean	True if checkbox is checked.		
validator	Validator	javax.faces. validator. Validator	Validator instance to use to validate the control's submitted value.		control's
value	Value	Object	Control value.		

Name	Display Name	Туре	Description	Choices	Default
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change callback.		
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcut letter.		
dir	Directionality	String	Text direction LTR (left-to-right), RTL hint. (right-to-left)		
lang	Language	String	Language code for o	content.	
maxlength	Max Length	int	Max number of allo signals unbounded.		s than one
size	Size	int	Number of characters wide to display field ("width" property is preferred).		
style	CSS Style	String	HTML "style" attrib	oute value.	
styleClass	CSS Class	String	HTML "class" attrib	oute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip.		
Expert Client Side	Events				
onblur	Blur	script	Control losing focus	S.	
onfocus	Focus	script	Control gaining foc	us.	
onclick	Click	script	Control clicked.		
onchange	Change	script	Control value chang	ged.	
ondblclick	Double Click	script	Control double-click	ked.	
onkeydown	Key Down	script	Key pressed when control is focused.		
onkeypress	Key Press	script	Key pressed and released when control is focused.		
onkeyup	Key Up	script	Key released when control is focused.		
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved when	n over control.	
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		

Name	Display Name	Туре	Description	Choices	Default
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		
	Children				
	None.				
	Facets				
	None.				

Multi-Line Text Input

javax.faces.HtmlInputTextarea

CAF JSF Base Controls Component Library

Multi-line text input.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Standard JSF Properties

Name	Display Name	Туре	Description	Choices	Default		
General Properties							
id	ID	String	Control ID.				
rendered	Rendered	boolean	True to display contro	1.	true		
Display Properties	S						
disabled	Disabled	boolean	True to display as "gra	ayed-out".	false		
readonly	Read Only	boolean	True to display as un-editable. fal		false		
Value Properties							
converter	Converter	javax.faces. convert. Converter	Converter instance to to and from a String.	use to convert the contro	l's value		
immediate	Immediate	boolean	±	ChangeListeners in the phase; false to process ons phase.	false		
required	Required	boolean	True to display contro	l as "required".	false		

Name	Display Name	Туре	Description	Choices	Default
validator	Validator	javax.faces. validator. Validator	Validator instance to u submitted value.	se to validate the control	l's
value	Value	Object	Control value.		
valueChange Listener	Value Change Listener	javax.faces.el.Me thod Binding	Value change callback.		
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcut lett	er.	
cols	Columns	int	Width in characters ("v	width" property is prefer	rred).
dir	Directionalit y	String	Text direction hint. LTR (left-to-right), RTL (r to-left)		
lang	Language	String	Language code for con	itent.	
rows	Rows	int	Height in characters.	5	
style	CSS Style	String	HTML "style" attribut	e value.	
styleClass	CSS Class	String	HTML "class" attribut	e value.	
tabindex	Tab Index	int		uence, from 1 to n. Zero a negative integer removes	
title	Element Title	String	HTML "title" attribute tooltip.	value (often displayed a	is a
Expert Client Side	Events				
onblur	Blur	script	Control losing focus.		
onfocus	Focus	script	Control gaining focus.		
onclick	Click	script	Control clicked.		
onchange	Change	script	Control value changed		
ondblclick	Double Click	script	Control double-clicked	l.	
onkeydown	Key Down	script	Key pressed when con	trol is focused.	
onkeypress	Key Press	script	Key pressed and release	sed when control is focus	sed.
onkeyup	Key Up	script	Key released when cor	ntrol is focused.	
onmousedown	Mouse Down	script	Mouse button pressed	when over control.	
onmousemove	Mouse Move	script	Mouse moved when o	ver control.	
onmouseout	Mouse Out	script	Mouse moved out of co	ontrol boundaries.	

Name	Display Name	Туре	Description Choices	Default
onmouseover	Mouse Over	script	Mouse moved into control boundaries.	
onmouseup	Mouse Up	script	Mouse button released when over control.	

Extended CAF Properties

Name	Display Name	Туре	Description	Choices	Default
Display Prope	erties				
width	Width	String	Pre-defined input width.	input2 (~2 chara input4, input6, i input20, input30 input50 (~50 cha	nput8, input10, 0, input40,

Children

None.

Standard JSF Facets

None.

Extended CAF Facets

None.

Option

javax.faces.SelectItem

CAF JSF Base Controls Component Library

An individual option in a select control (such as "Dropdown" on page 305, "Radio Button Group" on page 330, "Extended Select-Many Listbox" on page 308, "Checkbox Group" on page 294, and others). A standard JavaServer Faces control. An option can be configured via any of these properties:

- value property, in the form of a com.webmethods.caf.faces.data.ISelectItemProvider object
- value property, in the form of a javax.faces.model.SelectItem object
- itemLabel, itemValue, itemDescription, itemDisabled, itemStyle, and itemIcon properties

Every option should be configured with at least a label and a value (although the value can be null).

You cannot drop an input control outside of a Form control ("Form" on page 268).

Also see "Option Group" on page 326.

Standard JSF Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
itemDescription	Item Description	String	Extra descriptive text.	
itemLabel	Item Label	boolean	Option label.	
Value Properties				
itemValue	Item Value	Object	Option value.	
value	Value	com.webmethods.caf. faces.data.ISelectItem Provider (preferred) or javax.faces.model. SelectItem	Option configuration.	
Expert Display Prop	erties			
itemDisabled	Item Disabled	boolean	True to display as "grayed-out" and un-editable.	false

Extended CAF Properties

Name	Display Name	Туре	Description
Expert Display Prope	erties		
itemStyle	Item CSS Style	String	HTML "style" attribute value.
itemIcon	Item Icon	Image URL	Icon image URL.

Children

None.

Standard JSF Facets

Extended CAF Facets

None.

Option Group

javax.faces.SelectItems

CAF JSF Base Controls Component Library

A group of options in a select control ("Dropdown" on page 305, "Radio Button Group" on page 330, "Select-Many Listbox" on page 335, "Checkbox Group" on page 294, and others). A standard JavaServer Faces control. An option group can be configured via the value property, using any of these:

- single com.webmethods.caf.faces.data.ISelectItemProvider or javax.faces.model.SelectItem object
- array of com.webmethods.caf.faces.data.ISelectItemProvider or javax.faces.model.SelectItem objects
- collection of com.webmethods.caf.faces.data.ISelectItemProvider or javax.faces.model.SelectItem objects
- map of String option labels to Object option values

Note that since the ISelectItemGroupProvider interface extends the ISelectItemProvider interface, any particular ISelectItemProvider option actually can be a ISelectItemGroupProvider group containing a sub-collection of ISelectItemProvider options. The same goes for the javax.faces.model.SelectItems class (SelectItems extend SelectItem).

Every individual option should be configured with at least a label and a value (although the value can be null). However, option groups do not need to be configured with either a label or a value. An option group with a label usually is rendered as a nested group.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Also see "Option" on page 324.

Name	Display Name	Туре	Description	Default
General Proper	rties			
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description	Default
Value Properties	S			
value	Value	com.webmethods.caf.faces. data.ISelectItemProvider (preferred), array of ISelectItemProviders, collection of ISelectItemProviders, javax.faces.model.SelectItem, array of SelectItems, collection of SelectItems, or Map of label/value pairs	Group configuration.	

Name	Display Name	Туре	Description	Default			
Display Properties							
description	Description	String	Extra descriptive text.				
label	Label	boolean	Group label.				
Expert Display F	Properties						
disabled	Disabled	boolean	True to display group label as "grayed-out" and un-editable.	false			
icon	Icon	Image URL	Group icon URL.				
style	CSS Style	String	HTML "style" attribute value.				

Children

None.

Standard JSF Facets

None.

Extended CAF Facets

Radio Button

com.webmethods.caf.faces.select.SelectBooleanRadioButton

CAF JSF Base Controls Component Library

A single radio button. Usually you would use several together, grouped via the "name" property. The model is java.lang.Boolean (true if checked).

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control id.		
name	Name	String	Name of radio-butto	n group.	
rendered	Rendered	boolean	True to display contr	rol.	true
Display Properties					
disabled	Disabled	boolean	True to display as "g editable.	grayed-out" and un-	false
label	Label	String	Radio button label.		
readonly	Read Only	boolean	True to display as ur	n-editable.	false
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instance to use to convert the control value to and from a String.		ntrol's
immediate	Immediate	boolean	True to process Valu the Apply Request V process in the Process		false
required	Required	boolean	True to display cont	rol as "required."	false
selected	Selected	boolean	True if radio button	is selected.	
validator	Validator	javax.faces. validator. Validator	Validator instance to use to validate the control's submitted value.		ntrol's
value	Value	Object	Control value.		
valueChange Listener	Value Change Listener	javax.faces.el. MethodBinding	Value change callbac	ck.	

Name	Display Name	Туре	Description	Choices	Default
Expert Display Prop	perties				
accesskey	Access Key	String	Keyboard shortcut le	etter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), R (right-to-left)	ΓL
lang	Language	String	Language code for co	ontent.	
style	CSS Style	String	HTML "style" attrib	ute value.	
styleClass	CSS Class	String	HTML "class" attrib	ute value.	
tabindex	Tab Index	int		quence, from 1 to n. Ze , negative integer remo	
title	Title	String	HTML "title" attribu tooltip.	te value (often displaye	ed as a
Expert Client Side E	Events				
onblur	Blur	script	Control losing focus.		
onfocus	Focus	script	Control gaining focu	s.	
onchange	Change	script	Control value change	ed.	
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-click	ed.	
onkeydown	Key Down	script	Key pressed when co	ontrol is focused.	
onkeypress	Key Press	script	Key pressed and rele	eased when control is fo	cused.
onkeyup	Key Up	script	Key released when c	ontrol is focused.	
onmousedown	Mouse Down	script	Mouse button presse	ed when over control.	·
onmousemove	Mouse Move	script	Mouse moved when over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button release	ed when over control.	

None.

Facets

Radio Button Group

javax.faces.HtmlSelectOneRadio

CAF JSF Base Controls Component Library

A group of radio buttons. Selection options are specified via "Option" on page 324 and "Option Group" on page 326 children. A standard JavaServer Faces control.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default		
General Properties	5						
id	ID	String	Control ID.				
rendered	Rendered	boolean	True to display contr	ol.	true		
Display Properties	Display Properties						
disabled	Disabled	boolean	True to display as "g and un-editable.	rayed-out"	false		
layout	Layout	String	Layout orientation.	lineDirectio n (horizontal), pageDirecti on (vertical)	lineDirection		
readonly	Read Only	boolean	True to display as un-editable.		false		
Value Properties							
converter	Converter	javax.faces.convert. Converter	Converter instance to value to and from a S		the control's		
immediate	Immediate	boolean	True to process ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		false		
required	Required	boolean	True to display control as "required".		false		
validator	Validator	javax.faces. validator.Validator	Validator instance to use to validate the control's submitted value.		the control's		
value	Value	Object	Current selection val	ue.			

Name	Display Name	Туре	Description	Choices	Default
valueChange Listener	Value Change Listener	javax.faces.el. MethodBinding	Value change callbac	k.	
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcut le	etter.	
border	Border	String	Border width in pixe	ls.	0
dir	Directionality	String	Text direction hint.	LTR (left-to- (right-to-left)	
disabledClass	Disabled CSS Class	String	CSS class to apply to	disabled optio	ns.
enabledClass	Enabled CSS Class	String	CSS class to apply to options.	enabled	
lang	Language	String	Language code for co	ontent.	
style	CSS Style	String	HTML "style" attribu	ıte value.	
styleClass	CSS Class	String	HTML "class" attribu	ute value.	
tabindex	Tab Index	int	Order in tab (key) sec item to tab sequence, item.	•	
title	Element Title	String	HTML "title" attribu tooltip.	te value (often	displayed as a
Expert Client Side	Events				
onblur	Blur	script	Control losing focus.		
onfocus	Focus	script	Control gaining focu	S.	
onclick	Click	script	Control clicked.		
onchange	Change	script	Control value change	ed.	
ondblclick	Double Click	script	Control double-click	ed.	
onkeydown	Key Down	script	Key pressed when confocused.	ontrol is	
onkeypress	Key Press	script	Key pressed and rele	ased when cor	trol is focused.
onkeyup	Key Up	script	Key released when confocused.	ontrol is	
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved when	over control.	

Name	Display Name	Туре	Description	Choices	Default
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button release control.	ed when over	

Name	Display Name	Туре	Description	Choices	Default
Display Propert	ies				
width	Width	String	Pre-defined input width.	input2 (~2 characters wide input6, input8, input10, input30, input40, input5 characters wide)	input20,
Value Properties	S				
other	Other	boolean	True to allow users to enter values other than the specified values.		false
sortByLabel	Sort By Label	boolean	True to sort options alphal	petically by label text.	false

Children

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Standard JSF Facets

None.

Extended CAF Facets

Secret Input

javax.faces.HtmlInputSecret

CAF JSF Base Controls Component Library

A password-style text input. A standard JavaServer Faces control.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
redisplay	Redisplay	boolean		l with actual value (a ssue); false to render with	false
rendered	Rendered	boolean	True to display con	trol.	true
Display Properties					
disabled	Disabled	boolean	True to display as '	ʻgrayed-out".	false
readonly	Read Only	boolean	True to display as ı	ın-editable.	false
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instance value to and from a	to use to convert the contr a String.	ol's
immediate	Immediate	boolean	the Apply Request	ueChangeListeners in Values phase; false to ess Validations phase.	false
required	Required	boolean	True to display con	trol as "required".	false
validator	Validator	javax.faces. validator. Validator	Validator instance submitted value.	to use to validate the contr	ol's
value	Value	Object	Control value.		
valueChange Listener	Value Change Listener	javax.faces.el. Method Binding	Value change callba	ack.	
Expert Display Prop	erties				

Name	Display Name	Туре	Description	Choices Default
accesskey	Access Key	String	Keyboard shortcut let	ter.
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL (right-to-left)
lang	Language	String	Language code for co	ntent.
maxlength	Max Length	int	Max number of allow signals unbounded.	ed characters. Less than one
size	Size	int	Number of characters property is preferred)	wide to display field ("width").
style	CSS Style	String	HTML "style" attribu	te value.
styleClass	CSS Class	String	HTML "class" attribu	te value.
tabindex	Tab Index	int		uence, from 1 to n. Zero adds negative integer removes item.
title	Element Title	String	HTML "title" attribut tooltip.	e value (often displayed as a
Expert Client Side E	Events			
onblur	Blur	script	Control losing focus.	
onfocus	Focus	script	Control gaining focus	S.
onchange	Change	script	Control value change	d.
onselect	Select	script	Control text selected.	
onclick	Click	script	Control clicked.	
ondblclick	Double Click	script	Control double-clicke	d.
onkeydown	Key Down	script	Key pressed when co	ntrol is focused.
onkeypress	Key Press	script	Key pressed and relea	ased when control is focused.
onkeyup	Key Up	script	Key released when co	ontrol is focused.
onmousedown	Mouse Down	script	Mouse button pressed	d when over control.
onmousemove	Mouse Move	script	Mouse moved when	over control.
onmouseout	Mouse Out	script	Mouse moved out of	control boundaries.
onmouseover	Mouse Over	script	Mouse moved into co	ntrol boundaries.
onmouseup	Mouse Up	script	Mouse button release	d when over control.

Name	Display Name	Туре	Description	Choices	Default
Display Pr	operties				
width	Width	String	Pre-defined input width.	input2 (~2 characters wide), input4, inj input8, input10, input20, input30, inpu input50 (~50 characters wide)	

Children

None.

Standard JSF Facets

None.

Extended CAF Facets

None.

Select-Many Listbox

javax.faces.HtmlSelectManyListbox

CAF JSF Base Controls Component Library

Multi-select listbox. Selection options are specified via "Option" on page 324 and "Option Group" on page 326 children. Note that the control's value (the currently selected options' values) is an array of selected values.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Propertie	s				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cont	trol.	true
Display Properties	S				
disabled	Disabled	boolean	True to display as "	grayed-out".	false
readonly	Read Only	boolean	True to display as u	n-editable.	false

Name	Display Name	Туре	Description	Choices	Default
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instance to value to and from a	to use to convert the c String.	control's
immediate	Immediate	boolean	True to process Valuin the Apply Requesto process in the Prophase.	st Values phase; false	false
required	Required	boolean	True to display cont	rol as "required".	false
validator	Validator	javax.faces.validator. Validator	Validator instance to submitted value.	o use to validate the c	ontrol's
value	Value	Object	Current selection va	lues (as an array of v	alues).
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change callba	ck.	
Expert Display Pro	operties				
accesskey	Access Key	String	Keyboard shortcut l	etter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), (right-to-left)	RTL
disabledClass	Disabled CSS Class	String	CSS class to apply to	o disabled options.	
enabledClass	Enabled CSS Class	String	CSS class to apply to	enabled options.	
lang	Language	String	Language code for o	content.	
size	Size	String	Height in lines.		
style	CSS Style	String	HTML "style" attrib	oute value.	
styleClass	CSS Class	String	HTML "class" attrib	oute value.	
tabindex	Tab Index	int	. 27	equence, from 1 to n. uence, negative integ	
title	Element Title	String	HTML "title" attributooltip.	ute value (often displa	nyed as a
Expert Client Side	Events				
onblur	Blur	script	Control losing focus	S.	
onfocus	Focus	script	Control gaining focu	ıs.	

Name	Display Name	Туре	Description	Choices	Default
onclick	Click	script	Control clicked.		
onchange	Change	script	Control value chang	ged.	
ondblclick	Double Click	script	Control double-click	ked.	
onkeydown	Key Down	script	Key pressed when c	ontrol is focused.	
onkeypress	Key Press	script	Key pressed and released when control is focused.		
onkeyup	Key Up	script	Key released when	control is focused.	
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved wher	over control.	
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button release control.	sed when over	

Name	Display Name	Туре	Description	Choices	Default			
Display Properti	Display Properties							
width	Width	String	Pre-defined input width.	input2 (~2 characters v input6, input8, input10 input30, input40, inpu characters wide)), input20,			
Value Properties	8							
other	Other	boolean	True to allow users to enter values other than the false specified values.		false			
sortByLabel	Sort By Label	boolean	True to sort options alphab	petically by label text.	false			

Children

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Standard JSF Facets

Extended CAF Facets

None.

Select-One Button

com.webmethods.caf.faces.select.SelectOneButton

Base Controls Component Library

A button that behaves like a select-one group. Clicking the button toggles it through the options in the group. Selection options are specified via "Option" on page 324 and "Option Group" on page 326 children.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Properties	S				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display co	ntrol.	true
Display Properties	5				
disabled	Disabled	boolean	True to display as editable.	"grayed-out" and un-	false
width	Width	String	Pre-defined button width.	button2 (~2 characters button3, button4, button6, button7, button9, button10, but button9, button10, but button14, button16, bu button20 (~20 character	on5, on8, ton12, utton18,
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance value to and from	e to use to convert the co a String.	ontrol's
immediate	Immediate	boolean	True to process ValueChangeListeners fa in the Apply Request Values phase; false to process in the Process Validations phase.		false
other	Other	boolean	True to allow user than the specified	s to enter values other values.	false
required	Required	boolean	True to display co	ntrol as "required".	false

Name	Display Name	Туре	Description	Choices	Default
sortByLabel	Sort By Label	boolean	True to sort option label text.	ns alphabetically by	false
validator	Validator	javax.faces.validator. Validator	Validator instance submitted value.	to use to validate the co	ontrol's
value	Value	Object	Current selection	value.	
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change calll	oack.	
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcu	t letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL (right-to-left)	
disabledClass	Disabled CSS Class	String	CSS class to apply	to disabled options.	
enabledClass	Enabled CSS Class	String	CSS class to apply to enabled options.		
lang	Language	String	Language code for content.		
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attribute value.		
tabindex	Tab Index	int		sequence, from 1 to n. 2 equence, negative intege	
title	Element Title	String	HTML "title" attri tooltip).	ibute value (often displa	yed as a
Expert Client Side	Events				
onblur	Blur	script	Control losing foc	us.	
onfocus	Focus	script	Control gaining for	ocus.	
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-cl	icked.	
onkeydown	Key Down	script	Key pressed wher	n control is focused.	
onkeypress	Key Press	script	Key pressed and r	released when control is	focused.
onkeyup	Key Up	script	Key released when control is focused.		
onmousedown	Mouse Down	script	Mouse button pre control.	ssed when over	

Name	Display Name	Туре	Description	Choices	Default
onmousemove	Mouse Move	script	Mouse moved wh	nen over control.	
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button rel- control.	eased when over	

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Select-One Link

com.webmethods.caf.faces.select.SelectOneLink

Base Controls Component Library

A link that behaves like a select-one group. Clicking the link toggles it through the options in the group. Selection options are specified via javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Choices	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" and un-editable.	false
Value Properties				
converter	Converter	javax.faces. convert. Converter	Converter instance to use to convert the value to and from a String.	e control's

Name	Display Name	Туре	Choices	Default
immediate	Immediate	boolean	True to process ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.	false
other	Other	boolean	True to allow users to enter values other than the specified values.	false
required	Required	boolean	True to display control as "required".	false
sortByLabel	Sort By Label	boolean	True to sort options alphabetically by label text.	false
validator	Validator	javax.faces. validator. Validator	Validator instance to use to validate the submitted value.	ne control's
value	Value	Object	Current selection value.	
valueChange Listener	Value Change Listener	javax.faces.el. MethodBinding	Value change callback.	
Expert Display Pro	perties			
accesskey	Access Key	String	Keyboard shortcut letter.	
dir	Directionality	String	Text direction hint. LTR (left-to-r (right-to-left)	ight), RTL
disabledClass	Disabled CSS Class	String	CSS class to apply to disabled options	•
enabledClass	Enabled CSS Class	String	CSS class to apply to enabled options.	
lang	Language	String	Language code for content.	
style	CSS Style	String	HTML "style" attribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to adds item to tab sequence, negative ir removes item.	
title	Title	String	HTML "title" attribute value (often distooltip).	splayed as a
Expert Client Side	Events			
onblur	Blur	script	Control losing focus.	
onfocus	Focus	script	Control gaining focus.	

Name	Display Name	Туре	Choices Default
onclick	Click	script	Control clicked.
ondblclick	Double Click	script	Control double-clicked.
onkeydown	Key Down	script	Key pressed when control is focused.
onkeypress	Key Press	script	Key pressed and released when control is focused.
onkeyup	Key Up	script	Key released when control is focused.
onmousedown	Mouse Down	script	Mouse button pressed when over control.
onmousemove	Mouse Move	script	Mouse moved when over control.
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.
onmouseover	Mouse Over	script	Mouse moved into control boundaries.
onmouseup	Mouse Up	script	Mouse button released when over control.

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Select-One Listbox

javax.faces.HtmlSelectOneListbox

CAF JSF Base Controls Component Library

A single-select listbox. Selection options are specified via javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children. A standard JavaServer Faces control.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control.	true	

Name	Display Name	Туре	Description	Choices	Default
Display Properties					
disabled	Disabled	boolean	True to display as "grayed-out" and unequitable.		
readonly	Read Only	boolean	True to display as un-editable.		false
Value Properties					
converter	Converter	javax.faces. convert.Converter	Converter instance value to and from a	to use to convert the con String.	ntrol's
immediate	Immediate	boolean	the Apply Request	ueChangeListeners in Values phase; false to ess Validations phase.	false
required	Required	boolean	True to display cont	trol as "required".	false
validator	Validator	javax.faces. validator. Validator	Validator instance to submitted value.	o use to validate the cor	ntrol's
value	Value	Object	Current selection value.		
valueChange Listener	Value Change Listener	javax.faces.el. MethodBinding	Value change callba	ck.	
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shortcut	letter.	
dir	Directionalit y	String	Text direction hint.	LTR (left-to-right), RT to-left)	L (right-
disabledClass	Disabled CSS Class	String	CSS class to apply t	o disabled options.	
enabledClass	Enabled CSS Class	String	CSS class to apply t	o enabled options.	
lang	Language	String	Language code for content.		
size	Size	String	Height in lines.		
style	CSS Style	String	HTML "style" attrib	oute value.	
styleClass	CSS Class	String	HTML "class" attrib	oute value.	
tabindex	Tab Index	int		equence, from 1 to n. Ze e, negative integer remo	

Name	Display Name	Туре	Description Choices Default		
title	Title	String	HTML "title" attribute value (often displayed as a tooltip).		
Expert Client Side I	Events				
onblur	Blur	script	Control losing focus.		
onfocus	Focus	script	Control gaining focus.		
onchange	Change	script	Control value changed.		
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clicked.		
onkeydown	Key Down	script	Key pressed when control is focused.		
onkeypress	Key Press	script	Key pressed and released when control is focused.		
onkeyup	Key Up	script	Key released when control is focused.		
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved when over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		

Name	Display Name	Туре	Description	Choices	Default	
Display Properties						
width	Width	String	Pre-defined input input2 (~2 characters wide), input width. input6, input8, input10, input20, input30, input40, input50 (~50 characters wide)		input20,	
Value Properties						
other	Other	boolean	True to allow users to enter values other than false the specified values.		false	
sortByLabel	Sort By Label	boolean	True to sort options alphabetically by label text. false			

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Standard JSF Facets

None.

Extended CAF Facets

None.

Select-One Tabs

com.webmethods.caf.faces.select.SelectOne Tabs

Base Controls Component Library

Tabs that behave like a select-one group. Selection options are specified via "Option" on page 324 and "Option Group" on page 326 children.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Propertie	es .				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display co	ntrol.	true
Display Propertie	S				
disabled	Disabled	boolean	True to display as "grayed-out" and uneditable.		false
layout	Layout	String	Tabs layout.	top, left, bottom, right	top
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance to use to convert the control's value to and from a String.		
immediate	Immediate	boolean	True to process ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		false

Name	Display Name	Туре	Description	Choices	Default
other	Other	boolean	True to allow users to enter values other than the specified values.		false
required	Required	boolean	True to display co	ontrol as "required".	false
sortByLabel	Sort By Label	boolean	True to sort optio label text.	ns alphabetically by	false
validator	Validator	javax.faces.validator. Validator	Validator instance submitted value.	e to use to validate the co	ontrol's
value	Value	Object	Current selection	value.	
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change call	back.	
Expert Display Pro	operties				
accesskey	Access Key	String	Keyboard shortcu	ıt letter.	·
dir	Directionality	String	Text direction LTR (left-to-right), hint. RTL (right-to-left)		
disabledClass	Disabled CSS Class	String	CSS class to apply to disabled options.		
enabledClass	Enabled CSS Class	String	CSS class to apply	y to enabled options.	
lang	Language	String	Language code fo	or content.	
style	CSS Style	String	HTML "style" att	tribute value.	
styleClass	CSS Class	String	HTML "class" att	tribute value.	
tabindex	Tab Index	int) sequence, from 1 to n. 2 sequence, negative intege	
title	Element Title	String	HTML "title" attr tooltip).	ribute value (often displa	yed as a
Expert Client Side	Events				
onblur	Blur	script	Control losing for	cus.	
onfocus	Focus	script	Control gaining f	ocus.	
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clicked.		
onkeydown	Key Down	script	Key pressed when control is focused.		

Name	Display Name	Туре	Description Choices	Default	
onkeypress	Key Press	script	Key pressed and released when control is focused.		
onkeyup	Key Up	script	Key released when control is focused.		
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved when over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Simple Schedule Input

com.webmethods.caf.faces.input.SimpleScheduleInput

Base Controls Component Library

A simple schedule input. The "value" property of this control must be a com.webmethods.caf.faces.data.object.Schedule object. The Schedule model allows for an interval of time (expressed by a com.webmethods.rtl.date.DateFields object), plus a relative starting date (the "at" property — also a DateFields object).

The DateFields object has the same fields as a java.util.Calendar object (Calendar.DAY_OF_YEAR, Calendar.HOUR_OF_DAY, Calendar.MINUTE, etc.), but unlike the Calendar, DateFields allows the fields to be set independently from one another. The parse() and format() methods of DateFields allow the fields to converted to and from a string representation. For example, the string representation for 5 days is represented as "5D"; the string representation for 12:30 AM is "00:30". See the DateFields javadocs for a complete description.

A schedule that executes every 5th day at 12:30 AM can be represented by a Schedule object with an interval of "5D" (5 days) and an "at" property of "00:30" (at 12:30 AM with 24-hour time). A schedule that executes every 5th day (without a specific start time) can be represented by a Schedule object with an interval of "5D" and no "at" property.

The simple schedule input control allows a user to configure only the day (Calendar.DAY_OF_YEAR or "xD"), hour (Calendar.HOUR_OF_DAY or "xx:00"), and minute (Calendar.MINUTE or "00:xx") fields of the schedule interval; and only the hour (Calendar.HOUR_OF_DAY or "xx:00") and minute (Calendar.MINUTE or "00:xx") fields of the schedule "at" property.

You cannot drop an input control outside of a Form control ("Form" on page 268).

JSF Properties

Name	Display Name	Туре	Description	Default	
General Properties	5				
id	ID	String	Control ID		
rendered	Rendered	boolean	True to display control.	true	
Display Properties	i				
disabled	Disabled	boolean	True to display as "grayed-out."	false	
readonly	Read Only	boolean	True to display as un-editable.	false	
Value Properties					
converter	Converter	javax.faces.convert. Converter	Not used (schedule is automatically converted).		
immediate	Immediate	boolean	True to process ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		
required	Required	boolean	True to display control as "required".	false	
validator	Validator	javax.faces.validator. Validator	Validator instance to use to validate submitted value.	e the control's	
value	Value	com.webmethods.caf. faces.data.object. Schedule	Control value.		
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change callback.		
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcut letter.		
style	CSS Style	String	HTML "style" attribute value.		

Name	Display Name	Туре	Description	Default
styleClass	CSS Class	String	HTML "class" attribute val	lue.
tabindex	Tab Index	int	Order in tab (key) sequence, adds item to tab sequence, removes item.	
	Children			
	None.			
	Facets			
	None.			

Swapbox

com. we bmethods. caf. faces. select. Select Many Swapbox

Base Controls Component Library

A swapbox (two listboxes with options that shuttle between). Selection options are specified via "Option" on page 324 and "Option Group" on page 326 children. Note that the control's value (the currently selected options' values) is an array of selected values.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description Choices	Default
General Properties	5			
id	ID	String	Control ID	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
availableLabel	Available Label	String	Label for available listbox.	Defaults to a localized string; in English "Available".
disabled	Disabled	boolean	True to display as "grayed-out" and un-editable.	false
filter	Filter	boolean	True to display filter fields.	false

Name	Display Name	Туре	Description	Choices	Default
readonly	Read Only	boolean	True to display a	s un-editable.	false
selectedLabel	Selected Label	String	Label for selecte	d listbox.	Defaults to a localized string; in English "Selected".
width	Width	String	Pre-defined input width.	ers wide), out8, input10, input40, acters wide)	
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instar value to and from	ice to use to convert n a String.	the control's
immediate	Immediate	boolean	True to process fa ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		false
move	Re-order	boolean	True to allow sel be moved up or	false	
other	Other	boolean	True to allow use other than the sp	ers to enter values pecified values.	false
required	Required	boolean	True to display o	control as	false
sortByLabel	Sort By Label	boolean	True to sort option by label text.	ons alphabetically	false
validator	Validator	javax.faces.validator. Validator	Validator instand submitted value	ce to use to validate	the control's
value	Value	Object	Current selection	n values (as an array	of values).
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change callback.		
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shorte	rut letter.	
dir	Directionality	String	Text direction LTR (left-to-right), RTL (right-hint. to-left)		
lang	Language	String	Language code f	or content.	

Name	Display Name	Туре	Description	Choices	Default
size	Size	String	Height in lines.		
style	CSS Style	String	HTML "style" at	ttribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.		
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).		
Expert Properties					
filter	Filter	String	A filter expressi filterable data m		er tables bound to

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Text Input

javax.faces.HtmlInputText

CAF JSF Base Controls Component Library

A single-line text input. A standard JavaServer Faces control.

You cannot drop an input control outside of a Form control ("Form" on page 268).

Name	Display Name	Туре	Description	Choices	Default
General Propertie	S				
id	ID	String	Control ID		
rendered	Rendered	boolean	True to display co	ontrol.	true

Name	Display Name	Туре	Description	Choices	Default
Display Properties	S				
disabled	Disabled	boolean	True to display as editable.	"grayed-out" and un-	false
readonly	Read Only	boolean	True to display as	un-editable.	false
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance value to and from	e to use to convert the case a String.	ontrol's
immediate	Immediate	boolean	in the Apply Requ	alueChangeListeners lest Values phase; false Process Validations	false
required	Required	boolean	True to display co	ontrol as "required".	false
validator	Validator	javax.faces.validator. Validator	Validator instance submitted value.	e to use to validate the c	ontrol's
value	Value	Object	Control value.		
valueChange Listener	Value Change Listener	javax.faces.el.Method Binding	Value change call	back.	
Expert Display Pro	operties				
accesskey	Access Key	String	Keyboard shortcu	ıt letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL (right-to-left)	
lang	Language	String	Language code for content.		
maxlength	Max Length	int	Max number of all signals unbounde	llowed characters. Less ted.	than one
size	Size	int	Number of charac ("width" property	eters wide to display field is preferred).	ld
style	CSS Style	String	HTML "style" attı	ribute value.	
styleClass	CSS Class	String	HTML "class" attr	ibute value.	
tabindex	Tab Index	int	` 2,	sequence, from 1 to n. 2 equence, negative integ	
title	Element Title	String	HTML "title" attri tooltip).	bute value (often displa	yed as a

Name	Display Name	Туре	Description	Choices	Default
Expert Client Side	Events				
onblur	Blur	script	Control losing focus.		
onfocus	Focus	script	Control gaining f	ocus.	
onchange	Change	script	Control value cha	inged.	
onselect	Select	script	Control text selected.		
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clicked.		
onkeydown	Key Down	script	Key pressed when control is focused.		
onkeypress	Key Press	script	Key pressed and	released when contro	l is focused.
onkeyup	Key Up	script	Key released whe	en control is focused.	
onmousedown	Mouse Down	script	Mouse button precontrol.	essed when over	
onmousemove	Mouse Move	script	Mouse moved wh	nen over control.	
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button relecontrol.	eased when over	

Name	Display Name	Туре	Description	Choices
Display Propert	ies			
width	Width	String	Pre-defined input width.	input2 (~2 characters wide), input4, input6, input8, input10, input20, input30, input40, input50 (~50 characters wide)

Children

None.

Standard JSF Facets

None.

Extended CAF Facets

Chapter 15. Output Controls

Applet	. 356
Button	. 357
Control Label	. 360
Custom Element	. 362
Flash	. 363
Formatted Message	. 365
Formatted Messages	. 366
Formatted Text	. 366
Frame	. 368
Header	. 369
Horizontal Rule	. 370
lcon	. 371
Image	. 374
Include HTML	. 375
Include Resource Bundle	. 376
Include Script	. 377
Include Stylesheet	. 378
Link	. 379
Message	. 381
Messages	. 383
Parameterized Text	. 384
Refresh Button	. 385
Refresh Icon	. 386
Refresh Interval	. 387
Refresh Link	. 388
Script Block	. 389
Select-One Output Text	. 389
Specific Formatted Messages	. 390
Specific Messages	. 391
Text	. 392
Truncated Text	. 394

Applet

com.webmethods.caf.faces.output.Applet

Base Controls Component Library

A control that displays a Java applet. Either the control's "value" property must specify the jar containing the applet, or the control's "href" facet must contain a "Portlet URL" on page 487 control that specifies the jar containing the applet, or — if the applet uses only a single class file — the control's "codebase" property must specify the base URL for the applet's class file.

The applet class itself should be specified via the control's "code" property (for example, as "MyApplet.class") unless the applet is a serialized object, in which case the class should be specified via the control's "object" property.

Name	Display Name	Туре	Description	Choices	Default		
Display Properties							
border	Border	String	Border width in	pixels.			
height	Height	String	Applet height ir "100%").	Applet height in pixels (ex "100") or percent (ex "100%").			
width	Width	String	Applet width in "100%").	Applet width in pixels (ex "100") or percent (ex "100%").			
General Properties							
code	Code	String	Applet class name (ex "MyApplet.class").				
id	ID	String	Control ID.	Control ID.			
minimum Version	Minimum Version	String	Minimum required JRE version.				
object	Object	String	Applet class nar	me, if serialized (ex s").			
rendered	Rendered	boolean	True to display	control.	true		
Value Properties							
value	Value	Object	URL of jar conta	aining applet.			
Expert Properties							
codebase	Code Base	String	Base URL of app	Base URL of applet class file and resources.			
mayscript	May Script	boolean	True to allow ap	pplet to access brow	ser objects.		

Name	Display Name	Туре	Description	Choices	Default	
name	Name	String	Programmatic applet name.			
Expert Display Prop	perties					
align	Alignment	String	Alignment of applet within window.	top, middle, botto	m, left, right	
marginheight	Margin Height	String	Applet margin height in pixels (margin is the space between border and content).			
marginwidth	Margin Width	String	Applet margin width in pixels (margin is the space between border and content).			
styleClass	CSS Class	String	HTML "class" attribute value.			
style	CSS Style	String	HTML "style" attribute value.			
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).			

UIParameter children specify applet parameters. All other children specify content to display if browser doesn't support plug-ins (aka alternate content).

Facets

Name	Display Name	Description	Allowed Types
href	Portlet URL	URL of jar containing applet. Overrides "value" property.	PortletURL

Button

com.webmethods.caf.faces.output.HtmlButton

Base Controls Component Library

A control that draws a simple HTML button. This control can be used to load a URL or to invoke client-side actions using com. webmethods.caf.faces.portleturl. Portlet Url Script children controls.

When used as an alternative to the Link control, loads the URL specified by this control's "value" parameter. The URL can be parameterized by adding javax.faces.Parameter children.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	trol.	true
Display Properties					
disabled	Disabled	boolean	True to display as "	grayed-out".	false
width	Width	String	Pre-defined button width.	button2 (~2 charace button3, button4, button6, button7, button9, button10 button14, button1 button18, button2 characters wide)	button5, button8, , button12, 6,
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance value to and from a	to use to convert the String.	e control's
label	Label	String	Label text (displaye	ed before contents of	children).
value	Value	Object	Link URL.		
Expert Properties					
target	Target	String	Name of frame or window whose content will be replaced with the linked resource.	_blank (new wind (current frame), _l (parent frame), _to window), other	parent
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shortcut	letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL (right-to-left)	
lang	Language	String	Language code for	content.	
style	CSS Style	String	HTML "style" attrib	oute value.	
styleClass	CSS Class	String	HTML "class" attrib	oute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		

Name	Display Name	Туре	Description	Choices	Default	
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).			
Expert Client Side E	Events					
onblur	Blur	script	Control losing focu	S.		
onfocus	Focus	script	Control gaining foo	cus.		
onclick	Click	script	Control clicked.			
ondblclick	Double Click	script	Control double-clicked.			
onkeydown	Key Down	script	Key pressed when control is focused.			
onkeypress	Key Press	script	Key pressed and released when control is focused.			
onkeyup	Key Up	script	Key released when control is focused.			
onmousedown	Mouse Down	script	Mouse button pressed when over control.			
onmousemove	Mouse Move	script	Mouse moved whe	n over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.			
onmouseover	Mouse Over	script	Mouse moved into	control boundarie	S.	
onmouseup	Mouse Up	script	Mouse button released when over control.			

Any. Children are displayed as button label (after value of "label" property).

javax.faces.Parameter children are added as request parameters to the link.

com.webmethods.caf.faces.portleturl.PortletUrlScript children may be used to hook into client-side javascript events (such as onclick).

Facets

Control Label

javax.faces.HtmlOutputLabel

CAF JSF Base Controls Component Library

A control that displays a label for a control. A standard JavaServer Faces control.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	"Control ID Reference" on page 192	ID of control that la	abel is for.	
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cor	ntrol.	true
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance value to and from a		rt the control's
value	Value	Object	Control value.		
Expert Display Prop	perties				
accesskey	Access Key	String	Keyboard shortcut	letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-ri (right-to-left)	ght), RTL
lang	Language	String	Language code for	content.	
style	CSS Style	String	HTML "style" attri	bute value.	
styleClass	CSS Class	String	HTML "class" attri	bute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attrib tooltip.	ute value (often	displayed as a
Expert Client Side E	Events				
onblur	Blur	script	Control losing focu	15.	
onfocus	Focus	script	Control gaining for	cus.	

Name	Display Name	Туре	Description	Choices	Default
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clic	ked.	
onkeydown	Key Down	script	Key pressed when	control is focused.	
onkeypress	Key Press	script	Key pressed and released when control is focused.		
onkeyup	Key Up	script	Key released when control is focused.		
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved whe	n over control.	
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		

Extended CAF Properties

Name	Display Name	Туре	Description
Display Properties			
separator	Separator	String	Separator character between label and control.

Children

None.

Standard JSF Facets

None.

Extended CAF Facets

Custom Element

com. we bmethods. caf. faces. output. Custom

Base Controls Component Library

A custom HTML element. Attributes can be specified by means of UIParameter children.

Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control id.		
name	Name	String	HTML element tag na	me for example, "table").	
rendered	Rendered	boolean	True to display control	l.	true
Expert Display Properties					
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTI to-left)	_ (right-
lang	Language	String	Language code for cor	itent.	
style	CSS Style	String	HTML "style" attribut	e value.	
styleClass	CSS Class	String	HTML "class" attribute value.		
title	Title	String	HTML "title" attribute value (often displayed as a tooltip).		s a

Children

Any. UIParameter children specify the HTML element's custom attributes.

Facets

Flash

com.webmethods.caf.faces.output.Flash

Base Controls Component Library

A control that displays a Flash movie. Either the control's "value" property must specify the movie source, or the control's "href" facet must contain a "Portlet URL" on page 487 control that specifies the movie source.

Name	Display Name	Туре	Description	Choices	Default
Display Properties					
height	Height	String	Movie height in pixels (ex "100") or percent (ex "100%").		
menu	Menu	String	True to allow full r	ight-click menu o	n movie.
loop	Loop	String	True to loop movie		
play	Play	String	True to auto-play n	novie.	
quality	Quality	String	Rendering quality.	low, autolow, n autohigh, high,	· ·
width	Width	String	Movie width in pix "100%").	xels (ex "100") or p	ercent (ex
General Properties					
expressInstall	Express Install	String	True to display the Flash Player Express Install movie (instead of the alternate content) if the browser's Flash Player version is below the minimum required version.		
id	ID	String	Control ID.		
minimum Version	Minimum Version	String	Minimum required	l Flash Player vers	sion.
rendered	Rendered	boolean	True to display con	itrol.	true
Value Properties					
value	Value	Object	URL of flash movie	2.	
Expert Properties					
allowscript access	Allow Script Access	boolean	True to allow movi	e to access brows	er objects.

Name	Display Name	Туре	Description	Choices	Default
base	Base	String	Base URL of movie	and resources.	
name	Name	String	Programmatic mov	vie name.	
Expert Display Prop	erties				
salign	Alignment	String	Alignment of t (top), l (left), b (bottom), r movie within (right), tl (top-left), tr (top-window. right), bl (bottom-left), br (bottom-right)		tr (top-
bgcolor	Background Color	String	Movie background	color.	
devicefont	Device Font	boolean	True to use client's false to use fonts en	fonts to render text, nbedded in movie.	false
vspace	Margin Height	String	Movie margin heig between border an	ht in pixels (margin is d content).	the space
hspace	Margin Width	String	Movie margin width in pixels (margin is the space between border and content).		
seamlesstabbing	Seamless Tabbing	boolean	True to allow seamless tabbing between true web page and movie.		
style	CSS Style	String	HTML "style" attril	oute value.	
styleClass	CSS Class	String	HTML "class" attril	oute value.	
swliveconnect	Seamless Tabbing	boolean	True to start Java w Flash Player.	hen loading the	false
title	Element Title	String	HTML "title" attrib tooltip).	ute value (often displ	ayed as a
wmode	Window Mode	String	Movie transparency and layering.	window (default), opaque (transparent sections of movie use movie's background color), transparent (transparent sections of movie show web page beneath movie)	window

UIParameter children specify movie parameters. All other children specify content to display if the browser does not have the minimum required version of the Flash Player installed (alternate content).

Facets

Name	Display Name	Description	Allowed Types
href	Portlet URL	URL of movie. Overrides "value" property.	PortletURL

Formatted Message

com.webmethods.caf.faces.output.ErrorMessage

Base Controls Component Library

Displays a message (such as validation messages) associated with a specific control with a fixed, specialized look-and-feel.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
for	For Component	Control ID Reference	ID of control for which to messages.	show
showDetail	showDetail	boolean	True to show detail message (in addition to summary message).	true

Children

None.

Facets

Formatted Messages

com.webmethods.caf.faces.output.ErrorMessages

Base Controls Component Library

Displays faces messages (such as validation messages) with a fixed, specialized look-and-feel.

Properties

Name	Display Name	Туре	Description	Default
General Properti	es			
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
globalOnly	Global Only	boolean	True to show only messages not associated with any control.	false
showDetail	showDetail	boolean	True to show detail message (in addition to summary message).	true

Children

None.

Facets

None.

Formatted Text

com. we bmethods. caf. faces. output. For matted Text

Base Controls Component Library

Displays the value of this control as formatted plain text (honoring newlines, tabs, and spaces).

Properties

Name	Display Name	Туре	Description	Default	
Display Propert	ies				
monospace	Monospace	boolean	True to display with a true monospaced font.		
softwrap	Softwrap	boolean	True to wrap text at the edge of the container.	true	
truncate	Truncate	int	Number of characters to displatruncating (zero = don't trunca		
Expert Display Properties					
styleClass	CSS Class	String	HTML "class" attribute value.		
style	CSS Style	String	HTML "style" attribute value.		
dir	Directionality	String	HTML "dir" attribute value used to indicate the directionality of the flow of the content for the current element.		
title	Element Title	String	HTML "title" attribute value (c displayed as a tooltip).	often	
lang	Language	String	HTML "lang" attribute value u specify the language of the end content.		
General Propert	ties				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control.	true	
Value Properties	s				
converter	Converter	javax.faces. convert. Converter	Converter instance to use to convert the control's value to and from a String.		
value	Value	Object	Text.		

Children

None.

Facets

Frame

com.webmethods.caf.faces.output.Frame

Base Controls Component Library

A control that displays another Web page within an IFrame. Either the control's "value" property must specify the URL of the page to frame, or the control's "href" facet must contain a "Portlet URL" on page 487 control that specifies the URL of the page to frame.

Name	Display Name	Туре	Description	Choices	Default
Display Properties					
frameborder	Border	String	1 to display border, 0 to hide border	1 (show border), 0 (border)	Thide
height	Height	String	Frame height in pixel "100%").	s (ex "100") or percen	t (ex
width	Width	String	Frame width in pixels "100%").	s (ex "100") or percent	(ex
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display contro	ol.	true
Value Properties					
value	Value	Object	URL of page to frame		
Expert Properties					
name	Name	String	Programmatic frame	name.	
Expert Display Prop	erties				
align	Alignment	String	Alignment of frame within window.	top, middle, bottom	n, left, right
marginheight	Margin Height	String	Frame margin height between border and c	• •	he space
marginwidth	Margin Width	String	Frame margin width between border and c		ne space
scrolling	Scrolling	String	Show scrollbars.	yes, no, auto	
styleClass	CSS Class	String	HTML "class" attribut	te value.	

Name	Display Name	Туре	Description	Choices	Default
style	CSS Style	String	HTML "style" attribu	te value.	
title	Element Title	String	HTML "title" attribute tooltip).	e value (often displaye	ed as a

None.

Facets

Name	Display Name	Description	Allowed Types
href	Portlet URL	URL of page to frame. Overrides "value" property.	PortletURL

Header

com.webmethods.caf.faces.output.Header

Base Controls Component Library

An HTML section header (such as H1-H6). The header's level (1-6) can be specified via the control's "level" property. The header's content may be specified by the control's "value" property (for plain text), or via the control's content.

Name	Display Name	Туре	Description	Choices	Default	
General Proper	ties					
id	ID	String	Control ID.			
rendered	Rendered	boolean	True to display co	ntrol.	true	
Display Propert	ies					
level	Level	Integer	Heading level.	1-6 (1=primary, 2=secondary, etc.)	3	
Value Propertie	s					
converter	Converter	javax.faces.convert. Converter	Converter instance to use to convert the control's value to and from a String.			
value	Value	Object	Heading text (displayed before control's children).			

Name	Display Name	Туре	Description	Choices	Default			
Expert Display I	Expert Display Properties							
dir	Directionality	String	Text direction hint.	LTR (left-to-right), to-left)	RTL (right-			
lang	Language	String	Language code for content.					
style	CSS Style	String	HTML "style" attribute value.					
styleClass	CSS Class	String	HTML "class" attribute value.					
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip.					

Command, Graphic, or Output controls (displayed after the control's "value" property).

Facets

None.

Horizontal Rule

com. we bmethods. caf. faces. output. Horizontal Rule

Base Controls Component Library

A simple horizontal section separator.

Name	Display Name	Туре	Description	Choices	Default
General Properties	S				
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display co	ontrol.	true
Expert Display Pro	operties				
dir	Directionality	String	Text direction hint.	LTR (left-to-rig to-left)	ght), RTL (right-
lang	Language	String	Language code fo	or content.	
style	CSS Style	String	HTML "style" attr	ribute value.	

Name	Display Name	Туре	Description	Choices	Default
styleClass	CSS Class	String	HTML "class" a	nttribute value.	
title	Element Title	String	HTML "title" at tooltip.	tribute value (often o	displayed as a
	Children				
	None.				
	Facets				
	None.				

Icon

com. we bmethods. caf. faces. output. I con

Base Controls Component Library

An icon, with an optional link to a URL specified by this control's "value" parameter. The URL may be parameterized by adding javax.faces.Parameter children.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	ntrol.	true
Display Properties					
disabled	Disabled	boolean	True to display as	"grayed-out".	false
alt	Alternate Text	String	Alternate text to d displayed.	isplay if image can	not be
url	Icon URL	String	URL of (base) imag	ge source.	
mousedownUrl	Mouse Down Icon URL	String	URL of optional in is pressed over ico		nen mouse
mouseoverUrl	Mouse Over Icon URL	String	URL of optional in hovers over icon.	nage to display wh	nen mouse

Name	Display Name	Туре	Description	Choices	Default
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance to use to convert the control's value to and from a String.		9
label	Label	String	Label text (display	ed before icon).	
value	Value	Object	Link URL.		
Expert Properties					
charset	Charset	String	Character encoding	g of linked resource.	
coords	Coordinates	String	Client-side image	map coordinates.	
hreflang	Href Language	String	Language of linked	d resource.	
rel	Relationship	String	Link relationship f the linked resource	rom the current reso	urce to
rev	Reverse Link	String	Link relationship from the linked resource to current resource.		rce to the
shap	Shape	String	Client-side image map shape.	default, rect, circle	, poly
target	Target	String	Name of frame or window whose content will be replaced with the linked resource.	_blank (new windo (current frame), _to (parent frame), _to (current window),	parent pp
type	Content type	String	Content type of linked resource.		
Expert Display Pro	perties				
accesskey	Access Key	String	Keyboard shortcut	letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), (right-to-left)	RTL
lang	Language	String	Language code for	content.	
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attribute value.		
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attrib a tooltip.	oute value (often disp	olayed as

Name	Display Name	Туре	Description Choices Default			
Expert Client Side	Events					
onblur	Blur	script	Control losing focus.			
onfocus	Focus	script	Control gaining focus.			
onclick	Click	script	Control clicked.			
ondblclick	Double Click	script	Control double-clicked.			
onkeydown	Key Down	script	Key pressed when control is focused.			
onkeypress	Key Press	script	Key pressed and released when control is focused.			
onkeyup	Key Up	script	Key released when control is focused.			
onmousedown	Mouse Down	script	Mouse button pressed when over control.			
onmousemove	Mouse Move	script	Mouse moved when over control.			
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.			
onmouseover	Mouse Over	script	Mouse moved into control boundaries.			
onmouseup	Mouse Up	script	Mouse button released when over control.			

Any. Children are displayed as the icon label (after the icon).

javax.faces.Parameter children are added as request parameters to the link.

com.webmethods.caf.faces.portleturl. ("Portlet URL" on page 487) children may be used to hook into client-side javascript events (onClick)

Facets

Image

javax.faces.image

CAF JSF Base Controls Component Library

Displays the image from an absolute URL or from a URL relative to the Web application. A standard javaServer Faces control.

Name	Display Name	Туре	Description	Choices	Default		
General Properties							
id	ID	String	Control ID.				
rendered	Rendered	boolean	True to display co	ontrol.	true		
Display Properties							
alt	Alternate Text	String	Alternate text to d	lisplay if image ca	nnot be displayed.		
height	Height	String	Applet height in p	Applet height in pixels (ex "100") or percent (ex "100%").			
width	Width	String	Applet width in pixels (ex "100") or percent (ex "100%").				
Value Properties							
value	Value	Object	Image URL.				
url	URL	Object	A synonym of the	value property			
Expert Display Prop	perties						
dir	Directionality	String	Text direction hint.	LTR (left-to-righ left)	t), RTL (right-to-		
ismap	Is Map	boolean	0	nat this image is to map. Such an ima hyperlink ("a").			
usemap	Use Map	String	The name of a client-side image map (an HTML "map" element) for which this element provides the image.				
lang	Language	String	Language code fo	or content.			
style	CSS Style	String	HTML "style" attr	HTML "style" attribute value.			
styleClass	CSS Class	String	HTML "class" attr	ribute value.			

Name	Display Name	Туре	Description Choices Default
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip.
Expert Client Side E	Events		
onclick	Click	script	Control clicked.
ondblclick	Double Click	script	Control double-clicked.
onkeydown	Key Down	script	Key pressed when control is focused.
onkeypress	Key Press	script	Key pressed and released when control is focused.
onkeyup	Key Up	script	Key released when control is focused.
onmousedown	Mouse Down	script	Mouse button pressed when over control.
onmousemove	Mouse Move	script	Mouse moved when over control.
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.
onmouseover	Mouse Over	script	Mouse moved into control boundaries.
onmouseup	Mouse Up	script	Mouse button released when over control.

None.

Facets

None.

Include HTML

com.webmethods.caf.faces.output.IncludeHTML

Base Controls Component Library

A control that includes the contents of an HTML file into the current page, specified by the control's Value property. The Value property must be a reference to an HTML file in the current Web application (relative to the Web application's WebContent directory). For example, the HTML file bar.html, located in the "foo" subdirectory of the WebContent directory, can be referenced by the value /foo/bar.html.

If the file is in the current Web application and you set the Evaluate property to true, JSF expressions embedded in the file will be evaluated on every request.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
evaluate	Evaluate	boolean	True to evaluate JSF expressions embedded in the external file.	false
id	ID	String	Control id.	
rendered	Rendered	boolean	True to display control.	true
Value Properties				
value	Value	Object	HTML file URL.	

Children

None.

Facets

None.

Include Resource Bundle

com.webmethods.caf.faces.output.IncludeResourceBundle

Base Controls Component Library

A control that includes a java.util.ResourceBundle into the current page, in the form of a basic client-side JavaScript object (a "client-side resource bundle"). The client-side object's properties map to the resource bundle's keys. The resource bundle name is specified via the control's "value" property.

The client-side object can be referenced via the CAF.getBundle(id) client-side JavaScript method. The id argument should be the client-side id of the control. For a control with an id of "myControlId", and a page bean with a name of "myPageBean", the following line (in a Script Block control) would store the client-side object in the MY_RESOURCES global variable, and then alert the resource keyed by the string "my-key":

var MY_RESOURCES = CAF.getBundle("#{myPageBean.clientIds['myControlId']}");
alert(MY_RESOURCES["my-key"]);

Properties

g Control ID.
g Control ID.
an True to display true control.
t Resource bundle name.

Children

None.

Facets

None.

Include Script

com.webmethods.caf.faces.output.IncludeScript

Base Controls Component Library

A control that includes a JavaScript file into current page, specified by the control's "value" property. The "value" property may be the URL of an external script file (such as "http://example.com/foo.js"); or, if the "value" property begins with a slash, it may be the reference to a JavaScript file in the current web application, relative to the web application's WebContent directory. For example, the file "bar.js", located in the "foo" subdirectory of the WebContent directory, may be referenced by the value "/foo/bar.js". If the file is in the current Web application, and you set the "evaluate" property to true, JSF expressions embedded in the file will be evaluated on every request.

Name	Display Name	Туре	Description	Default
General Properties				
evaluate	Evaluate	boolean	True to evaluate JSF expressions embedded in the external file.	false

Name	Display Name	Туре	Description	Default
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Value Properties				
value	Value	Object	JavaScript file URL.	

None.

Facets

None.

Include Stylesheet

com.webmethods.caf.faces.output.IncludeStylesheet

Base Controls Component Library

A control that includes a CSS stylesheet into current page, specified by the control's "value" property. The "value" property may be the URL of an external stylesheet (such as "http://example.com/foo.css"); or, if the "value" property begins with a slash, it may be the reference to a CSS stylesheet in the current Web application (relative to the Web application's WebContent directory). For example, the stylesheet "bar.css", located in the "foo" subdirectory of the WebContent directory, may be referenced by the value "/foo/bar.css". If the file is in the current Web application, and you set the "evaluate" property to true, JSF expressions embedded in the file will be evaluated on every request.

Name	Display Name	Туре	Description	Default
General Properties				
evaluate	Evaluate	boolean	True to evaluate JSF expressions embedded in the external file.	false
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description	Default
Value Properties				
value	Value	Object	Stylesheet URL.	
Children				
None.				
Facets				
None.				

Link

javax. faces. Html Output Link

CAF JSF Base Controls Component Library

A link to a URL specified by this control's "value" parameter. The URL may be parameterized by adding javax.faces.Parameter children.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	trol.	true
Value Properties					
converter	Converter	javax.faces. convert.Converter	Converter instance to use to convert the control's value to and from a String.		
label	Label	String	Label text (displayed before contents of children).		
value	Value	Object	Link URL.		
Expert Properties					
charset	Charset	String	Character encoding	of linked resource.	
coords	Coordinates	String	Client-side image n	nap coordinates.	
hreflang	Href Language	String	Language of linked	resource.	
rel	Relationship	String	Link relationship from the current resource to the linked resource.		rce to the

Name	Display Name	Туре	Description	Choices	Default
rev	Reverse Link	String	Link relationship from the linked resource to the current resource.		
shap	Shape	String	Client-side image map shape.	default, rect, circle,	poly
target	Target	String	Name of frame or window whose content will be replaced with the linked resource.	_blank (new windo (current frame), _pa (parent frame), _top window), other	irent
type	Content type	String	Content type of link	ked resource.	
Expert Display Prop	perties				
accesskey	Access Key	String	Keyboard shortcut	letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), I (right-to-left)	RTL
lang	Language	String	Language code for content.		
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attrib	oute value.	
tabindex	Tab Index	int		equence, from 1 to n. Z e, negative integer rer	
title	Element Title	String	HTML "title" attrib displayed as a toolt	•	
Expert Client Side I	Events				
onblur	Blur	script	Control losing focus.		
onfocus	Focus	script	Control gaining foo	cus.	
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clic	ked.	
onkeydown	Key Down	script	Key pressed when	control is focused.	
onkeypress	Key Press	script	Key pressed and re	leased when control is	s focused.
onkeyup	Key Up	script	Key released when	control is focused.	
onmousedown	Mouse Down	script	Mouse button pressed when over control.		l .
onmousemove	Mouse Move	script	Mouse moved whe	n over control.	
·	•		-	·	

Name	Display Name	Туре	Description Choices [Default
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.	
onmouseover	Mouse Over	script	Mouse moved into control boundaries.	
onmouseup	Mouse Up	script	Mouse button released when over control.	

Extended CAF Properties

Name	Display Name	Туре	Description	Choices	Default
Display Prope	rties				
disabled	Disabled	boolean	True to display as "gra	yed-out".	false

Children

Any. Children are displayed as link label (after value of "label" property).

javax.faces.Parameter children are added as request parameters to the link.

com.webmethods.caf.faces.portleturl.PortletUrlScript children can be used to hook into client-side javascript events (onClick)

Standard JSF Facets

None.

Extended CAF Facets

None.

Message

javax.faces.message

CAF JSF Base Controls Component Library

Message output for a component. A standard JavaServer Faces control.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description	Default
showDetail	showDetail	boolean	True to show detail false message.	
showSummary	Show Summary	boolean	True to show summary message.	true
tooltip	Tooltip	boolean	Flag indicating whether the detail portion of the message should be displayed as a tooltip.	false
for	For Component	Control ID Reference	ID of control for which to messages.	to show
Expert Display Prop	erties			
style	CSS Style	String	HTML "style" attribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.	
errorClass	Error CSS Class	String	CSS style class to apply message with a severity "ERROR".	
errorStyle	Error CSS Style	String	CSS style(s) to apply to message with a severity "ERROR".	
fatalClass	Fatal CSS Class	String	CSS style class to apply message with a severity "FATAL".	
fatalStyle	Fatal CSS Style	String	CSS style(s) to apply to message with a severity "FATAL".	
infoClass	Info CSS Class	String	CSS style class to apply message with a severity "INFO".	
infoStyle	Info CSS Style	String	CSS style(s) to apply to message with a severity "INFO".	

Name	Display Name	Туре	Description	Default
warnClass	Warn CSS Class	String	CSS style class to apply message with a severity "WARN".	
warnStyle	Warn CSS Style	String	CSS style(s) to apply to message with a severity "WARN".	

None

Standard JSF Facets

None.

Messages

javax.faces.message

CAF JSF Base Controls Component Library

Message output for a all components. A standard JavaServer Faces control.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display co	ntrol.	true
showDetail	showDetail	boolean	True to show deta	True to show detail message.	
showSummary	Show Summary	boolean	True to show sum	True to show summary message.	
tooltip	Tooltip	boolean		Flag indicating whether the detail portion of the message should be displayed as a tooltip.	
globalOnly	Global Only	boolean	True to show only with any control.	True to show only messages not associated with any control.	
layout	Layout	String	Layout style	list, table	list

Name	Display Name	Туре	Description	Choices	Default
Expert Display Prop	perties				
style	CSS Style	String	HTML "style" attribu	ite value.	
styleClass	CSS Class	String	HTML "class" attribu	ıte value.	
errorClass	Error CSS Class	String	CSS style class to apply to any message with a severity class of "ERROR".		
errorStyle	Error CSS Style	String	CSS style(s) to apply to any message with a severity class of "ERROR".		
fatalClass	Fatal CSS Class	String	CSS style class to apply to any message with a severity class of "FATAL".		
fatalStyle	Fatal CSS Style	String	CSS style(s) to apply to any message with a severity class of "FATAL".		
infoClass	Info CSS Class	String	CSS style class to ap severity class of "INI		with a
infoStyle	Info CSS Style	String	CSS style(s) to apply class of "INFO".	to any message wit	h a severity
warnClass	Warn CSS Class	String	CSS style class to apply to any message with a severity class of "WARN".		
warnStyle	Warn CSS Style	String	CSS style(s) to apply to any message with a severity class of "WARN".		

None

Standard JSF Facets

None.

Parameterized Text

A control that renders parameterized text. This is a core JSF control for which Software AG has done no customizations. For more information, see the Sun Microsystems documentation at these locations:

- ► http://java.sun.com/javaee/javaserverfaces/1.1_01/docs/renderkitdocs/HTML_BASIC/javax.faces.Outputjavax.faces.Format.html
- $\verb| http://java.sun.com/javaee/javaserverfaces/1.1/docs/api/javax/faces/component/html/HtmlOutputFormat.html| \\$

Refresh Button

com. we bmethods. caf. faces. output. refresh. Button

Base Controls Component Library

A button that asynchronously refreshes the content of a specified control.

Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	"Control ID Reference" on page 192	ID of control to re	efresh.	
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display co	ontrol.	true
Display Properties					
style	CSS Style	String	HTML "style" att	ribute value.	
styleClass	CSS Class	String	HTML "class" att	ribute value.	
disabled	Disabled	boolean	True to display as	s "grayed-out".	false
width	Width	String	Pre-defined button width.	button2 (~2 characters wide), button3, button4, button5, button6, button7, button8, button9, button10, button12, button14, button16, button18, button20 (~20 characters wide	
Value Properties					
label	Label	String	Label text (displa	yed before contents	of children).

Children

Any. Children are displayed as button label (after value of "label" property).

Facets

Refresh Icon

com. we bmethods. caf. faces. output. refresh. I con

Base Controls Component Library

An icon that asynchronously refreshes the content of a specified control.

Name	Display Name	Туре	Description	Default
General Properties				
for	For	"Control ID Reference" on page 192	ID of control to refresh.	
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out".	false
alt	Alternate Text	String	Alternate text to display if image cannot be displayed.	
url	Icon URL	String	URL of (base) image source.	
mousedownUrl	Mouse Down Icon URL	String	URL of optional image to display when mouse is pressed over icon.	
mouseoverUrl	Mouse Over Icon URL	String	URL of optional image to display when mouse hovers over icon.	
Value Properties				
label	Label	String	Label text (displayed icon).	before the

Any. Children are displayed as the icon label (after the icon).

Facets

None.

Refresh Interval

com. we bmethods. caf. faces. output. refresh. Interval

Base Controls Component Library

A client-side script that asynchronously refreshes the content of a specified control at a specified interval.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
for	For	"Control ID Reference" on page 192	ID of control to refresh.	
id	ID	String	Control ID.	
interval	Interval	int	Interval in seconds between refreshes.	60
rendered	Rendered	boolean	True to display control.	true
waitUntil Reloaded	Wait Until Reloaded	boolean	Wait until control finishes reloading to begin next interval.	false

Children

None.

Facets

Refresh Link

com. we bmethods. caf. faces. output. refresh. Link

Base Controls Component Library

A link that asynchronously refreshes the content of a specified control.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
for	For	"Control ID Reference" on page 192	ID of control to refresh.	
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out".	false
Value Properties				
label	Label	String	Label text (displayed l contents of children).	oefore

Children

Any. Children are displayed as link label (after value of "label" property).

Facets

Script Block

com.webmethods.caf.faces.output.Script

Base Controls Component Library

A control that adds a JavaScript block to the current page, containing the JavaScript code specified by the control's "value" property.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Value Properties				
value	Value	Object	JavaScript code.	

Children

None.

Facets

None.

Select-One Output Text

com.webmethods.caf.faces.select.SelectOneOutput

Base Controls Component Library

Text output that uses the select-one model. Model options are specified via javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children. The selected item (the item whose value matches this control's value) is the item displayed.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		

Name	Display Name	Туре	Description	Choices	Default
rendered	Rendered	boolean	True to display cor	ntrol.	true
Value Properties					
converter	Converter	javax.faces.convert. Converter	Converter instance to use to convert the control's value to and from a String.		
value	Value	Object	Current selection value.		
Expert Display Prop	perties				
dir	Directionality	String	Text direction LTR (left-to-right), RTL hint. (right-to-left)		RTL
lang	Language	String	Language code for	content.	
style	CSS Style	String	HTML "style" attri	bute value.	
styleClass	CSS Class	String	HTML "class" attribute value.		
title	Title	String	HTML "title" attribute value (often displayed as a tooltip).		ayed as a

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Specific Formatted Messages

com. we bmethods. caf. faces. output. Specific Formatted Messages

Base Controls Component Library

A control that displays faces messages for specific components, such as validation messages, with a fixed, specialized look and feel.

Name	Display Name	Туре	Description	Default
General Propertie	S			
id	ID	String	Control id.	

Name	Display Name	Туре	Description	Default
for	For	ControlID References	IDs of controls (or containers of controls) for who display messages. Separate multiple IDs with c	
rendered	Rendered	boolean	True to display control.	true
showDetail	showDetail	boolean	True to show a detail message, in addition to a summary message.	true

None.

Facets

None.

Specific Messages

com. we bmethods. caf. faces. output. Specific Messages

Base Controls Component Library

A control that displays faces messages for specific components, such as validation messages, with a customizable look and feel.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control id.		
for	For	ControlID References	IDs of controls (or conto display messages. S commas.		,
layout	Layout	String	Layout style.	list, table	list
rendered	Rendered	boolean	True to display contro	l.	true
showDetail	Show Detail	boolean	True to show detail message. fals		false
showSummary	Show Summary	boolean	True to show summar	y message.	true
tooltip	Tooltip	boolean	True to render details	as tooltip.	false

Name	Display Name	Туре	Description	Choices	Default	
Expert Display Pro	Expert Display Properties					
errorClass	Error CSS Class	String	CSS class for error i	nessages.		
errorStyle	Error CSS Style	String	CSS style for error i	nessages.		
fatalClass	Fatal CSS Class	String	CSS class for fatal error messages.			
fatalStyle	Fatal CSS Style	String	CSS style for fatal error messages.			
infoClass	Info CSS Class	String	CSS class for inform	national messages.		
infoStyle	Info CSS Style	String	CSS style for inform	national messages.		
style	CSS Style	String	CSS style for all me	ssages.		
styleClass	CSS Class	String	CSS class for all me	ssages.		
warnClass	Warn CSS Class	String	CSS class for warni	ng messages.		
warnStyle	Warn CSS Style	String	CSS style for warni	ng messages.		

None.

Facets

None.

Text

javax.faces.text

CAF JSF Base Controls Component Library

A basic text output control. A standard JavaServer Faces control.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
escape	Escape	boolean	True to HTML-encode the enclosed content.	true
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description Default		
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instance to use to convert the control's value to and from a String.		
value	Value	Object	Text to display.		
Expert Display Properties					
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attribute value.		
dir	Directionality	String	HTML "dir" attribute value used to indicate the directionality of the flow of the content for the curren element.	ne	
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).	en	
lang	Language	String	HTML "lang" attribute value used to specify the language of the enclosed content.	d	

Extended CAF Properties

Name	Display Name	Туре	Description	Default
Expert Pro	perties			
raw	Raw	boolean	True to omit enclosing HTML tags.	false

Children

None

Standard JSF Facets

Truncated Text

com. we bmethods. caf. faces. output. Truncated Text

Base Controls Component Library

A variation of the Output Text control that optionally truncates the text at the specified length and appends '...'. The full string is rendered as the 'title' attribute (tooltip) if no 'title' attribute is specified.

Name	Display Name	Туре	Description Default				
Display Properties							
truncateLen	Length	Integer	Number of characters to display before truncating				
Expert Display Pr	Expert Display Properties						
styleClass	CSS Class	String	HTML "class" attribute value.				
style	CSS Style	String	HTML "style" attribute value.				
dir	Directionality	String	HTML "dir" attribute value used to indicate the directionality of the flow of the content for the current element.				
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).				
lang	Language	String	HTML "lang" attribute value used to specify the language of the enclosed content.				
General Propertie	es						
id	ID	String	Control ID.				
escape	Escape	boolean	True to HTML- true encode the enclosed content.				
rendered	Rendered	boolean	True to display true control.				

Name	Display Name	Туре	Description	Default
Value Properties				
converter	Converter	javax.faces.convert. Converter	Converter instance to use to convert the control's value to and from a String.	
value	Value	Object	JavaScript code.	

None.

Facets

Chapter 16. List Controls

Async List	398
Async Listbox	401
Async Tabs	403
Listbox	406
Simple List	408
Tabs	411

Async List

com.webmethods.caf.faces.list.async.Simple

Base Controls Component Library

A list whose content comes from a com.webmethods.caf.faces.data.ITableContentProvider object. Standard list-like containers, such as java.util.List, are wrapped automatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset

Each item in the data model is mapped to a row in the list. The content of the list is rendered multiple times, once for each row displayed.

The difference between the async list and a regular list is that the async list pages asynchronously. See "Async Table" on page 531 for more information about configuring its asynchronous properties.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of fir display.	st row in page to	0
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control		true
rows	Rows	int	Number of rows in pag "all".	ge to display. Less tha	n one signals
Value Properties					
value	Value	ITableContent Provider	Data bound to table.		
var	Row Variable	String	Name of current row o row within column cor		ce current

Name	Display Name	Туре	Description	Choices	Default
Display Properties					
marker	Marker	String	String to place next to (for "vertical" or "horizontal" types) or between (for "horizontal-separated" type) each item. Does not apply to "ol" or "ul" types.	decimal (decimal roother (any raw ht	·
selected	Selected	boolean	True to display as "sele expression to select/un basis).		false
type	Туре	String	List type.	vertical, horizontal (marker aligned next to item), horizontal- separated (marker centered between items), ol (default ordered list), ul (default unordered list)	vertical
Lazy Load Propertie	es				
bufferChunk	Buffer Chunk	int	Chunk size in rows to client-side cache buffer buffer via one big chur	:. Zero signals fill	0
bufferMin	Buffer Minimum	int	Minimum size in rows of client-side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.		0
bufferMax	Buffer Maximum	int	Maximum size in rows buffer (hidden rows bu side of the visible table don't buffer.	iffered on either	0
clientSideCache	Client Side Cache	boolean	True to store rows in cle control or page is refre		false
flashOnChange	Flash On Change	boolean	True to flash individua changed when content asynchronously.		true

Name	Display Name	Туре	Description	Choices	Default
progressDelay	Progress Delay	int	Milliseconds to wait be progress dialog.	pefore showing	0
progressMsg	Progress Message	String	Message to display in	progress dialog.	Defaults to localized message; in English, "Loading"
suppressInputs	Suppress Inputs	Collection	List of ids of controls validation/update pro asynchronous reques	cessing when handlir	ig an
Expert Properties					
filter	Filter	String	A filter expression. Used to filter tables bound to filterable data models.		
Expert Display Prop	perties				
customRow Class	Custom Row CSS Class	String	Binding expression that specifies a custom CSS class name to add to the current row.		
rowClasses	Row CSS Classes	Collection	CSS class to use for each row, from top to bottom, repeating.		
style	CSS Style	String	HTML "style" attribut	e value.	
styleClass	CSS Class	String	HTML "class" attribut	e value.	
title	Title	String	HTML "title" attribute	e value.	

Any.

Facets

Async Listbox

com.webmethods.caf.faces.list.async.Listbox

Base Controls Component Library

A listbox whose content comes from a com.webmethods.caf.faces.data.ITableContentProvider object. Standard list-like containers, such as java.util.List, are wrapped automatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset

Each item in the data model is mapped to a row in the list. The content of the list is rendered multiple times, once for each row displayed.

The difference between the async list and a regular list is that the async list pages asynchronously. See "Async Table" on page 531 for more information about configuring its asynchronous properties.

Name	Display Name	Туре	Description	Choices	Default				
General Properties	General Properties								
first	First	int	Zero-based index of display.	f first row in page to	0				
id	ID	String	Control ID.						
rendered	Rendered	boolean	True to display control. true		true				
rows	Rows	int	Number of rows in page to display. Less than one signals "all".						
Value Properties									
value	Value	ITableContent Provider	Data bound to table.						
var	Row Variable	String	Name of current row object (used to reference current row within column content).						
Display Properties	5								
disabled	Disabled	boolean	True to display as " editable.	grayed-out" and un-	false				

Name	Display Name	Туре	Description	Choices	Default
selected	Selected	boolean	binding expression a per-row basis). See	True to display as "selected" (use a binding expression to select/deselect on a per-row basis). See TableRowTools for a separate technique to allow the user to select/deselect rows	
width	Width	String	Pre-defined input width.	input2 (~2 characters wide), input4, input6, input8, input10, input20, input30, input40, input50 (~50 characters wide)	input10
Lazy Load Proper	ties				
bufferChunk	Buffer Chunk	int	Chunk size in rows to use when filling the client-side cache buffer. Zero signals fill buffer via one big chunk.		0
bufferMin	Buffer Minimum	int	Minimum size in rows of client-side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.		0
bufferMax	Buffer Maximum	int	Maximum size in rows of client-side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.		0
clientSide Cache	Client Side Cache	boolean	True to store rows i until control or pag		false
flashOn Change	Flash On Change	boolean	True to flash individual rows that have changed when content finishes loading asynchronously.		true
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog.		0
progressMsg	Progress Message	String	Message to display in progress dialog.		Defaults to localized message; in English, "Loading"

Name	Display Name	Туре	Description	Choices	Default			
suppress Inputs	Suppress Inputs	Collection	validation/upda	List of ids of controls for which to disable validation/update processing when handling an asynchronous request (comma separated).				
Expert Display Pi	Expert Display Properties							
customRow Class	Custom Row CSS Class	String	0 1	Binding expression that specifies a custom CSS class name to add to the current row.				
rowClasses	Row CSS Classes	Collection	CSS class to use repeating.	CSS class to use for each row, from top to bottom, repeating.				
style	CSS Style	String	HTML "style" attribute value.					
styleClass	CSS Class	String	HTML "class" attribute value.					
title	Title	String	HTML "title" at	tribute value.				

Any.

Facets

None.

Async Tabs

com.webmethods.caf.faces.list.async.Tabs

Base Controls Component Library

A list of tabs whose content comes from a com.webmethods.caf.faces.data.ITableContentProvider object. Standard list-like containers, such as java.util.List, are wrapped automatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset

Each item in the data model is mapped to a tab in the list. The content of the list is rendered multiple times, once for each tab displayed.

The difference between the async tabs and the regular tabs is that the async tabs pages asynchronously. See "Async Table" on page 531 for more information about configuring its asynchronous properties.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of display.	first row in page to	0
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cont	rol.	true
rows	Rows	int	Number of rows in signals "all".	page to display. Less	than one
Value Properties					
value	Value	ITableContent Provider	Data bound to table		
var	Row Variable	String	Name of current row row within column	w object (used to reference content).	rence current
Display Properties					
layout	Layout	String	Tabs layout.	top, left, bottom, right	top
selected	Selected	boolean	True to display as "s binding expression t a per-row basis).		false
Lazy Load Propertie	S				
bufferChunk	Buffer Chunk	int	Chunk size in rows the client-side cache signals fill buffer via	buffer. Zero	0
bufferMin	Buffer Minimum	int	Minimum size in rows of client-side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.		0
bufferMax	Buffer Maximum	int	Maximum size in rows of client-side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.		0
clientSideCache	Client Side Cache	boolean	True to store rows in until control or page		false

Name	Display Name	Туре	Description	Choices	Default	
flashOnChange	Flash On Change	boolean	True to flash indivion changed when contasynchronously.		true	
progressDelay	Progress Delay	int	Milliseconds to wai progress dialog.	t before showing	0	
progressMsg	Progress Message	String	loc me En		Defaults to localized message; in English, "Loading"	
suppressInputs	Suppress Inputs	Collection	List of ids of controls for which to disable validation/update processing when handling an asynchronous request (comma separated).			
Expert Properties						
filter	Filter	String	A filter expression. Used to filter tables bound to filterable data models.			
Expert Display Prop	erties					
customRow Class	Custom Row CSS Class	String	Binding expression name to add to the	that specifies a custo current row.	m CSS class	
rowClasses	Row CSS Classes	Collection	CSS class to use for each row, from top to bottom, repeating.			
style	CSS Style	String	HTML "style" attrib	ute value.		
styleClass	CSS Class	String	HTML "class" attribute value.			
title	Title	String	HTML "title" attribu	ıte value.		

Any.

Facets

Listbox

com.webmethods.caf.faces.list.Listbox

Base Controls Component Library

A listbox whose content comes from a com.webmethods.caf.faces.data.ITableContentProvider object. Standard list-like containers, such as java.util.List, are wrapped automatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset

Each item in the data model is mapped to a row in the list. The content of the list is rendered multiple times, once for each row displayed.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of display.	first row in page to	0
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cont	rol.	true
rows	Rows	int	Number of rows in page to display. Less than one signals "all".		
Value Properties					
value	Value	ITableContent Provider	Data bound to table.		
var	Row Variable	String	Name of current rov reference current rov content).	, ,	
Display Properties					
disabled	Disabled	boolean	True to display as "g editable.	rayed-out" and un-	false

Name	Display Name	Туре	Description	Choices	Default	
selected	Selected	boolean	binding expression a per-row basis). Se a separate techniqu	True to display as "selected" (use a binding expression to select/deselect on a per-row basis). See TableRowTools for a separate technique to allow the user to select/deselect rows.		
width	Width	String	Pre-defined input width.	input2 (~2 characters wide), input4, input6, input8, input10, input20, input30, input40, input50 (~50 characters wide)	input10	
Expert Properties						
filter	Filter	String	A filter expression. filterable data mod	Used to filter tables bels.	oound to	
firstRequest Param	First Request Param	String	Name of request parameter that specifies table's first row. Used only for "restful" paging.			
rowsRequest Param	Rows Request Param	String		Name of request parameter that specifies number of rows for the table to display. Used only for "restful" paging.		
Expert Properties						
filter	Filter	String	A filter expression. filterable data mod	Used to filter tables bels.	oound to	
Expert Display Prop	perties					
customRow Class	Custom Row CSS Class	String	Binding expression class name to add t	that specifies a custo to the current row.	om CSS	
rowClasses	Row CSS Classes	Collection	CSS class to use for each row, from top to bottom, repeating.			
rowHeight	Row Height	String	Height of each row in pixels (20px) or ems (20em).			
style	CSS Style	String	HTML "style" attribute value.			
styleClass	CSS Class	String	HTML "class" attribute value.			
title	Title	String	HTML "title" attrib	HTML "title" attribute value.		

Any.

Facets

None.

Simple List

com.webmethods.caf.faces.list.Simple

Base Controls Component Library

A list whose content comes from a com.webmethods.caf.faces.data.ITableContentProvider object. Standard list-like containers, such as java.util.List, are wrapped automatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset
- javax.faces.model.DataModel

Each item in the data model is mapped to a row in the list. The content of the list will be rendered multiple times, one for each row displayed.

To enable "restful" paging, you must set the list's "firstParamName" property to the name of the request parameter from which the list derives its "first" property. For example, if the list binds the "myFirst" request parameter (where an example URL might be '/myTable?myFirst=10') to its "first" property, using a value binding expression "#{param.myFirst}", you would set the value of the list's "firstParamName" property to "myFirst". Note that in order to bind the value of a request parameter to the list's "first" property, you must manually convert the request parameter value to a primitive int (see the following getMyFirst() method). The list does not need to be contained within a Command Form to do "restful" paging.

To enable "restful" page sizing, you must set the list's "rowsParamName" property to the name of the request parameter from which the list derives its "rows" property. For example, if the list binds the "myRows" request parameter (where an example URL might be '/myTable?myRows=10') to its "rows" property, you would set the value of the list's "rowsParamName" property to "myRows". Note that in order to bind the value of a request parameter to the list's "rows" property, you must manually convert the request parameter value to a primitive int (see the following getMyRows() method). The list does not need to be contained within a Command Form to do "restful" page sizing.

```
// Convert "myFirst" request parameter to a primitive int.
public int getMyFirst() {
   FacesContext context = FacesContext.getCurrentInstance();
   String s = (String)
context.getExternalContext().getRequestParameterMap().get("myFirst");
   if (s != null)
       return Integer.parseInt(s);
   return 0;
// Convert "myRows" request parameter to a primitive int.
public int getMyRows() {
   FacesContext context = FacesContext.getCurrentInstance();
   String s = (String)
context.getExternalContext().getRequestParameterMap().get("myRows");
   if (s != null)
       return Integer.parseInt(s);
   return 10;
}
```

Name	Display Name	Туре	Description	Default
General Properties				
first	First	int	Zero-based index of first row in page to display.	0
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
rows	Rows	int	Number of rows in page to display. Less than one signals "all".	
Value Properties				
value	Value	ITableContent Provider	Data bound to table.	
var	Row Variable	String	Name of current row object (used to reference current row within column content).	

Name	Display Name	Туре	Description	Default
Display Properties				
marker	Marker	String	String to place next to (for "vertical" or "horizontal" types) or between (for "horizontal- separated" type) each item. Does not apply to "ol" or "ul" types.	
selected	Selected	boolean	True to display as "selected" (use a binding expression to select/unselect on a per-row basis).	false
type	Туре	String	List type.	vertical
Expert Properties				
firstRequest Param	First Request Param	String	Name of request parameter that specifies table's first row. Used only for "restful" paging.	
rowsRequest Param	Rows Request Param	String	Name of request parameter that specifies number of rows for the table to display. Used only for "restful" paging.	
Expert Display Pro	perties			
customRow Class	Custom Row CSS Class	String	Binding expression that specifies a custom CSS class name to add to the current row.	
rowClasses	Row CSS Classes	Collection	CSS class to use for from top to bottom repeating.	
style	CSS Style	String	HTML "style" attribute value.	

Name	Display Name	Туре	Description	Default
styleClass	CSS Class	String	HTML "class" attribute value.	
title	Title	String	HTML "title" attribute value.	

Any.

Facets

None.

Tabs

com.webmethods.caf.faces.list.Tabs

Base Controls Component Library

List of tabs whose content comes from either a com.webmethods.caf.faces.data.ITableContentProvider object. See com.webmethods."Simple List" on page 408 for information on data types and restful paging.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of display.	first row in page to	0
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	trol.	true
rows	Rows	int	Number of rows in signals "all".	page to display. Less	than one
Value Properties					
value	Value	ITableContentPr ovider	Data bound to table		
var	Row Variable	String	Name of current rocurrent row within	w object (used to refe column content).	rence

Name	Display Name	Туре	Description	Choices	Default
Display Properties					
layout	Layout	String	Tabs layout.	top, left, bottom, right	top
selected	Selected	boolean	True to display as binding expression a per-row basis).	"selected" (use a n to select/unselect on	false
Expert Properties					
filter	Filter	String	A filter expressior filterable data mod	ı. Used to filter tables l dels.	bound to
filterRequest Param	Filter Request Param	String		parameter that specifie Jsed only for "restful"	
firstRequest Param	First Request Param	String		parameter that specifie ly for "restful" paging.	s table's
rowsRequest Param	Rows Request Param	String		parameter that specifie ble to display. Used on	
Expert Display Prop	oerties				
customRow Class	Custom Row CSS Class	String	U 1	n that specifies a custo to the current row.	om CSS
rowClasses	Row CSS Classes	Collection	CSS class to use for repeating.	or each row, from top t	o bottom,
rowHeight	Row Height	String	Height of each rov	w in pixels (20px) or er	ns (20em).
style	CSS Style	String	HTML "style" attr	ibute value.	
styleClass	CSS Class	String	HTML "class" attr	ibute value.	
title	Title	String	HTML "title" attril	oute value.	

Any.

Facets

Chapter 17. Logic Controls

Async Iterator
Content Parameter
Control Parameter
Else
lf
Import View
Iterator
Load Resource Bundle
Naming Container

Async Iterator

com.webmethods.caf.faces.list.async.Raw

Base Controls Component Library

This control iterates over a com.webmethods.caf.faces.data.ITableContentProvider object, displaying the control's content once for each row in the provider. Standard list-like containers, such as java.util.List, are wrapped automatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset

The difference between Async Iterator and "Iterator" on page 421 is that Async Iterator pages asynchronously. See "Async Table" on page 531 for more information about configuring its asynchronous properties.

Name	Display Name	Туре	Description	Default
General Properties				
first	First	int	Zero-based index of first row in 0 page to display.	
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
rows	Rows	int	Number of rows in page to display signals "all".	. Less than one
Value Properties				
value	Value	ITableContent Provider	Data bound to table.	
var	Row Variable	String	Name of current row object (used t current row within column content	
Lazy Load Propertion	es			
bufferChunk	Buffer Chunk	int	Chunk size in rows to use when filling the client-side cache buffer. Zero signals fill buffer via one big chunk.	0

Name	Display Name	Туре	Description	Default
bufferMin	Buffer Minimum	int	Minimum size in rows of client- side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.	0
bufferMax	Buffer Maximum	int	Maximum size in rows of client- side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.	0
clientSideCache	Client Side Cache	boolean	True to store rows in client-side cache until control or page is refreshed.	false
flashOnChange	Flash On Change	boolean	True to flash individual rows that have changed when content finishes loading asynchronously.	true
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog.	0
progressMsg	Progress Message	String	Message to display in progress dialog.	Defaults to localized message; in English, "Loading"
suppressInputs	Suppress Inputs	Collection	List of ids of controls for which to c validation/update processing when asynchronous request (comma sepa	handling an
Expert Properties				
filter	Filter	String	A filter expression. Used to filter ta filterable data models.	bles bound to
	Children			
	Any.			
	Facets			

webMethods Designer - Composite Application Framework Help Version 7.1.1

Content Parameter

com.webmethods.caf.faces.logic.ContentParameter

CAF JSF Base Controls Component Library

A parameter that allows child content in place of a value property. It can be used with controls like the "Import View" on page 418 control to specify replacement content for incomplete sections of the imported view, or the "Parameterized Text" on page 384 control to specify message properties.

Properties

Name	Display Name	Туре	Description
General Properties	S		
name	Name	String	Parameter name.

Children

Any. Children are used as the parameter's value.

Facets

None.

Control Parameter

javax.faces.parameter

CAF JSF Base Controls Component Library

A control that adds a parameter to a containing control.

Name	Display Name	Туре	Description
General Properties			
name	Name	String	Parameter name.
value	Value	Object	Parameter value.

None.

Facets

None.

Else

com.webmethods.caf.faces.logic.Else

Base Controls Component Library

A panel that renders its children if its "value" property is true and its preceding "If" on page 418 or Else sibling is not true. (Note that the behavior is actually "else if", rather than an unconditional "else".)

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Value Properties				
value	Value	Object	True to render contents.	

Children

Any.

Facets

lf

com.webmethods.caf.faces.logic.lf

Base Controls Component Library

A panel that renders its children if its "value" property is true.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Value Properties				
value	Value	Object	True to render contents.	

Children

Any.

Facets

None.

Import View

com.webmethods.caf.faces.logic.lmport

Base Controls Component Library

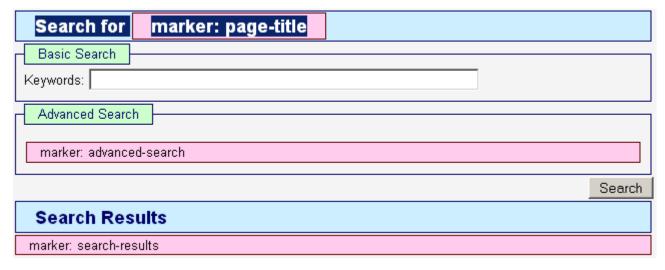
A control that imports the view specified by its "value" property directly into the current view tree. This allows you to interact with controls in the imported view as if they are present literally in the current view. This control is a naming container, meaning that it can disambiguate a control inside this container with the same ID as a control outside of it.

Imported views must be part of the same web application. A view "bar" located in the "foo" subdirectory of the web application's WebContent directory can be referenced by the value "/foo/bar.view".

Templating

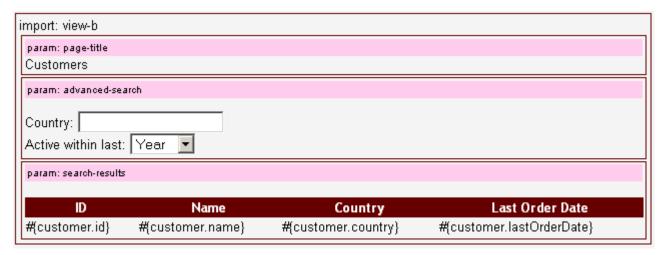
You can use the Import View control to import a template view, where the imported view contains incomplete sections (markers) into which the import control injects content specified on the importing view. A template view is a regular view used for the purpose of templating, and a marker control is any control (often a panel control) used for the purpose of marking a section to be replaced. You can specify the content to be injected by means of "Content Parameter" on page 416 children of the Import View control. For example, the following template view (View B) contains markers for the page title, the advanced-search property group, and the search-results table:

Search Page Template (View B)



The top-level instance view (View A) imports the template view, and specifies the content for the page title, advanced-search, and search-results sections:

Search Page Results (View A)



The name of a "Content Parameter" on page 416 in the Import View control should be the ID (or "Control ID Reference" on page 192) of the control in the imported view the Content Parameter's content will replace. If there is no Content Parameter present in the Import View control for a marker in the imported view (for example, if View A above did not have a search-results Content Parameter), the marker's content is rendered as is. In other words, the marker's content is the default content displayed if no Content Parameter for the marker has been specified.

You can use UIParameter controls (Control Parameters) in an Import View control to set property values on the imported view's page bean. The name of the UIParameter specifies the name of the page bean property, and the value specifies the value to which the property is set. In the above example, if View B's page bean had a property called pageTitle, a UIParameter with the name pageTitle could be used by View A to set that property (in the above example, it might set the property's value to "Customers"). View B could then use the page bean property to display the pageTitle, or use it for some other purpose, such as the input to a Web service.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Value Properties				
value	Value	Object	View to import.	

You can use UIParameter ("Content Parameter" on page 416) children to set property values of the imported view's page bean. Content Parameter children can be used to replace content in the imported view with the content of the Content Parameter.

Facets

None.

Iterator

com.webmethods.caf.faces.list.Raw

Base Controls Component Library

A control that iterates over a com.webmethods.caf.faces.data.ITableContentProvider object, displaying this control's content for each row in the provider. Standard list-like containers, such as java.util.List, are wrapped automatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset
- javax.faces.model.DataModel

To enable "restful" paging, you must set the list's "firstParamName" property to the name of the request parameter from which the list derives its "first" property. For example, if the list binds the "myFirst" request parameter (where an example URL might be '/myTable?myFirst=10') to its "first" property, using a value binding expression "#{param.myFirst}", you would set the value of the list's "firstParamName" property to "myFirst". Note that in order to bind the value of a request parameter to the list's "first" property, you must manually convert the request parameter value to a primitive int (see the following getMyFirst() method). The list does not need to be contained within a Command Form to do "restful" paging.

To enable "restful" page sizing, you must set the list's "rowsParamName" property to the name of the request param from which the list derives its "rows" property. For example, if the list binds the "myRows" request parameter (where an example URL might be '/myTable?myRows=10') to its "rows" property, you would set the value of the list's "rowsParamName" property to "myRows". Note that in order to bind the value of a request parameter to the list's "rows" property, you must manually convert the request parameter value to a primitive int (see the following getMyRows() method). The list does not need to be contained within a Command Form to do "restful" page sizing.

```
// Convert "myFirst" request parameter to a primitive int.
public int getMyFirst() {
   FacesContext context = FacesContext.getCurrentInstance();
   String s = (String) context.getExternalContext().getRequestParameterMap().get("myFirst");
   if (s != null)
      return Integer.parseInt(s);
   return 0;
}

// Convert "myRows" request parameter to a primitive int.
public int getMyRows() {
   FacesContext context = FacesContext.getCurrentInstance();
   String s = (String) context.getExternalContext().getRequestParameterMap().get("myRows");
   if (s != null) return Integer.parseInt(s); return 10;
}
```

Name	Display Name	Туре	Description	Default
General Properties	s			
first	First	int	Zero-based index of first row in page to display.	0
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
rows	Rows	int	Number of rows in page to display. Less that signals "all".	n one
Value Properties				
value	Value	ITableContent Provider	Data bound to table.	
var	Row Variable	String	Name of current row object (used to reference row within column content).	ce current
Expert Properties				
filter	Filter	String	Filter expression. Used to filter tables bound data models. See "Filter Input" on page 314.	to filterable
filterRequest Param	Filter Request Param	String	Name of the request parameter that specifies the table's filter expression. Used only for "restful" paging.	
firstRequest Param	First Request Param	String	Name of request parameter that specifies table's first row. Used only for "restful" paging.	

Name	Display Name	Туре	Description	Default
rowsRequest Param	Rows Request Param	String	Name of request parameter that specifies number of rows for the table to display. Used only for "restful" paging.	
raw	Raw	boolean	True to omit enclosing row HTML tags. Note that individual rows cannot be manipulated with client-side JavaScript while this is true.	false
	Children			
	Any.			
	Facets			

Load Resource Bundle

None.

com. we bmethods. caf. faces. logic. Load Resource Bundle

Base Controls Component Library

A control that loads a resource bundle into the current request scope so that it can be accessed in a binding expression following this component in the view. This control is deprecated; a better way to access a resource bundle is to add a method to your page bean that uses the

com.webmethods.caf.faces.bean.BaseFacesBean.getResourcesProvider(String bundleName) method to load a specified resource bundle as an IContentProvider. For example, the following method would load the bundle named "com.example.MyResourceBundle":

```
public ResourceBundleContentProvider getMyResourcesProvider() {
   return getResourcesProvider("com.example.MyResourceBundle");
}
```

You could then access resources in the bundle with expressions like the following (which access the resource in the bundle keyed with "foo"):

```
#{activePageBean.myResourcesProvider["foo"]}
```

Properties

Name	Display Name	Туре	Description	Default		
General Proper	General Properties					
id	ID	String	Control id.			
rendered	Rendered	boolean	True to load resource.	true		
Value Propertie	S					
basename	Bundle Name	String	Name of resource bundle.			
var	Variable Name	String	Name of variable to use to reference resource bundle.			

Children

None.

Facets

None.

Naming Container

CAF JSF Base Controls Component Library

A logical container that "namespaces" the client IDs of contained controls. This control modifies the way contained controls are referenced. See "Control ID Reference" on page 192.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Children

Any.

Facets

Chapter 18. Map Controls

Dynamic Map Marker List	426
Google Map	431
Google Map Key	435
Invoke Map Script	435
Map Marker	438
Map Marker Group	443
Map Navigation Control	444
Return Map Coords Script	445
Return Map Value Script	446

Dynamic Map Marker List

com.webmethods.caf.faces.gmap.GMarkerList

Base Controls Component Library

A dynamic list of markers for use with the "Google Map" on page 431 control. The data model for this control is an com.webmethods.caf.faces.data.ITableContentProvider (just like a table or list control). For each row in the table content-provider, a marker is rendered. The properties of this control can be bound to row properties, so that for example, if the provider's row object has street, city, state, and zipCode properties, the control's initialAddress property could be bound to the expression #{marker.street} #{marker.state} #{marker.state} #{marker.sipCode} (assuming the control's row variable is set to its default value of marker). If the provider in this example had two rows, one with a street value of "600 108th Ave NE", a city value of Bellevue, a state value of WA, and a zipCode property of 98004; and the other with a street value of "8822 Ridgeline Blvd," a city value of Littleton, a state value of CO, and a zipCode property of 80129, the marker-list would render two markers, one with an initialAddress of "600 108th Ave NE Bellevue WA 98004," and one with an initialAddress of "8822 Ridgeline Blvd Littleton CO 80129."

Each marker's location normally is specified using the Dynamic Map Marker List control's latitude and longitude property values. You can also specify marker locations using the initial Address property, if the latitude and longitude properties are both set to 0. The address can be formatted as a natural-language address with multiple lines concatenated into a single line with spaces (for example, "600 108th Ave NE, Bellevue, WA 98004 USA"). A two-letter ISO-3166 country code can specified with the initial Country property as a country/locale hint, although it is generally not needed.

The marker's icon can be specified using the Dynamic Map Marker List control's Icon category properties. You can use the standard CAF marker icons by configuring the icon property with an icon name (red, orange, yellow, green, cyan, blue, purple, or magenta), or you can use a custom icon by configuring the icon property with the URL of the icon (for example, "http://maps.google.com/mapfiles/arrow.png"). You can configure the marker's selected icon similarly using the selectedImage property. If you use a custom icon, you may also want to configure many of the expert Icon properties, such as iconSizeHeight, iconSizeWidth, shadow (the URL of the shadow image), shadowSizeHeight, shadowSizeWidth, and so forth. With the standard CAF marker icons, you can leave these properties blank.

The content of the marker-list is used to render each marker's information (bubble) window. A user can click on a marker to display the information window, and also to select the marker, if the control's clickable property is set to true for the marker. A user can drag the marker to a new location if the controls's draggable property is set to true for the marker.

Scripting

The Google Maps API is a full-featured, client-side JavaScript API, and is fully accessible for use with the CAF Google Maps controls. See http://www.google.com/apis/maps/ for the Google Maps API documentation.

To access the GMarker object (the Google Maps API marker object) associated with a particular marker in the marker-list, use the get() method the control's CAF client-side model object to get the CAF model object for the marker, and then the getOverlay() method on the CAF marker model object to get the GMarker object. The following example looks up the GMarker object with an id of myMarker from the marker-list control model with an ID of myMarkerList, and uses the Google Maps API to hide the marker:

```
var marker = CAF.model("#{activePageBean.clientIds["myMarkerList"]}").get("myMarker").getOverlay();
marker.hide();
```

The marker's ID can be configured on a table content-provider, commonly using a binding expression with the provider's rowId property. The marker can also be identified using its (zero-based) row index. The following example looks up the first GMarker object from the marker-list control model with an ID of myMarkerList, and uses the Google Maps API to hide the marker:

```
var marker = CAF.model("#{activePageBean.clientIds["myMarkerList"]}").get(0).getOverlay();
marker.hide();
```

Client-Side Model

The CAF client-side model object for this control extends from the CAF. Table. Model object (see "Client-Side Model" on page 201 for more information about the CAF client-side model). The object's value is the selected marker's ID, (or if the control's multiple property is set to true, an array of selected marker IDs). For example, you can set the selected marker of the marker-list with an ID of myMarkerList to the marker with an ID of myMarker with the following script:

```
CAF.model("#{activePageBean.clientIds["myMarkerList"]}").setValue("myMarker");
```

You can get the coordinates of a particular marker's location in string form (for example, 37.0625;-95.677068) by using the getCoords() method of the client-side model. The following script gets the coordinates of the marker with an ID of myMarker in a marker-list with an id of myMarkerList:

```
var coords = CAF.model("#{activePageBean.clientIds["myMarkerList"]}").get("myMarker").getCoords();
```

The CAF.GMap.stringToGLatLng() and CAF.GMap.gLatLngToString() functions convert string coordinates to and from a GLatLng object (the Google Maps API coordinates object). To set the coordinates, you must change the marker model's coordinates using the model's setCoords() method and then update the marker-list with the changed marker

model using the marker-list model's set() method. The following script sets the marker with an ID of myMarker in a marker-list with an ID of myMarkerList:

```
CAF.model("#{activePageBean.clientIds["myMarkerList"]}")
.set(CAF.model("#{activePageBean.clientIds["myMarker"]}").setCoords("37.0625;-95.677068"));
```

You can use the CAF client-side model to set a marker's location with a natural-language address. The following script sets the location of the marker with an ID of myMarker:

Script Controls

You can use the standard Script-category controls like "Return Value Script" on page 500 or "Invoke Script" on page 491 to manipulate the marker-list with the standard client-side actions (such as getValue or setValue). You can also use the custom "Return Map Value Script" on page 446 and "Invoke Map Script" on page 435 controls to access special maponly properties and actions of the individual markers in a marker-list such as getLatitude or setAddress. Additionally, the "Return Map Coords Script" on page 445 control allows latitude and longitude coordinates to be specified separately when invoking a marker's setCoords action.

JSF Properties

Name	Display Name	Туре	Description	Choices	Default			
General Properties	General Properties							
id	ID	String	Control id.					
rendered	Rendered	boolean	True to display the contro	ol.	true			
multiple	Multiple	boolean	True if multiple markers	may be selected.	false			
Display Properties	5							
title	Title	String	Marker ToolTip title.					
Value Properties								
initialAddress	Initial Address	String	Initial marker location if (for example, 600 108th A	O				
initialCountry	Initial Country	String	Two-letter country code (ISO 3166) hint for the initial address (for example, US).					
latitude	Latitude	double	Latitude of the marker location in degrees (for example, 37.0625).					

Name	Display Name	Туре	Description	Choices	Default
longitude	Longitude	double	Longitude of the marker location in degrees (for example, -95.677068).		
selected	Selected	boolean	True if the marker is selec	cted.	false
value	Value	ITableContent Provider	List of markers.		
var	Row Variable	String	Name of the variable with reference the current makes		marker
Icon Properties					
icon	Icon	String	URL of the marker icon (for example, http://maps.google.com/mapfiles/arrow.png), or a standard icon name (for example, red).	red, orange, yellow, green, cyan, blue, purple, magenta	red
iconSize Height	Icon Height	int	Icon height in pixels.		
iconSizeWidth	Icon Width	int	Icon width in pixels.		
selectedImage	Selected Image	String	URL of the icon to use when the marker is selected, or a standard icon name.	red, orange, yellow, green, cyan, blue, purple, magenta	yellow
Behavior Propertion	es				
clickable	Clickable	boolean	True to allow the user to select it or open the infor		false
draggable	Draggable	boolean	True to allow the user to drag the marker to fals a different location.		false
Expert Icon Prope	rties				
dragCross AnchorX	Drag Cross Anchor X	int	Horizontal offset of the canchor position in pixels.		icon
dragCross AnchorY	Drag Cross Anchor Y	int	Vertical offset of the cross image from the icon anchor position in pixels.		
dragCross Image	Drag Cross Image	String	URL of the image to display over the exact marker location when the marker is dragged.		
dragCross AnchorHeight	Drag Cross Anchor Height	int	Height of the cross image	e in pixels.	

Name	Display Name	Туре	Description	Choices	Default
dragCross AnchorWidth	Drag Cross Anchor Width	int	Width of the cross image in pixels.		
iconAnchorX	Icon Anchor X	int	Horizontal offset of the left edge of the icon from the marker location in pixels.		
iconAnchorY	Icon Anchor Y	int	Vertical offset of th marker location in	e top edge of the icon fro pixels.	m the
imageMap	Image Map	String	Comma-separated list of integers representing the x/y coordinates of the image map to specify the clickable part of the icon image in browsers other than Internet Explorer.		
infoWindow AnchorX	Info Window Anchor X	int		f the pointed tip of the in eft edge of the icon in pix	
infoWindow AnchorY	Info Window Anchor Y	int		e pointed tip of the inform op edge of the icon in pix	
mozPrint Image	Mozilla Print Image	String	URL of the icon to use for printing in Firefox.		
printImage	Print Image	String	URL of the icon to use for printing in browsers other than Firefox.		
printShadow	Print Shadow	String	URL of the image to use for the marker's shadow when printing.		
shadow	Shadow	String	URL of the image to use for the marker's shadow.		
shadowSize Height	Shadow Size Height	int	Height of the shade	ow in pixels.	
shadowSize Width	Shadow Size Width	int	Width of the shado	w in pixels.	
transparent	Transparent	String	URL of a virtually transparent version of the foreground icon image used to capture click events in Internet Explorer.		
Behavior Propertie	es				
autoPan	Auto Pan	boolean	True to auto-pan the marker near the	ne map if the user drags e map's edge.	true
bounceGravity	Bounce Gravity	int	Acceleration rate of bouncing.	f the marker when	1
bouncy	Bouncy	boolean		narker bounce up and r finishes dragging it.	true

Name	Display Name	Туре	Description	Choices	Default
dragCross Move	Drag Cross Move	boolean	<u> </u>	narker under the cursor alse to keep the drag-cross when dragging.	false
maxHeight	Max Height	int	Height in pixels for the marker to raise when dragged.		dragged.

Any. Children are rendered as the content of the marker's information (bubble) window.

Facets

None.

Google Map

com.webmethods.caf.faces.gmap.GMap

Base Controls Component Library

A Google map. To use this control, you must configure the control's key property with a valid Google Maps API key. You can register for a key at http://www.google.com/apis/maps/signup.html. When registering a key, you must specify the host name and port number of the server that will run the portlet or Web application; These values must be the host name and port number the user sees in the browser address bar when accessing the portlet or Web application. For example, if the user accesses the application using http://mws.example.com/, you need to use http://mws.example.com/ when registering for a key.

If the map is not rendered during the initial page load (that is, if the map is rendered only as part of an async refresh and not as part of the initial page), you must include a "Google Map Key" on page 435 control as part of the initial page, configured with the Google Maps API key. Otherwise, you can ignore the Google Map Key control.

The center of the map normally is specified using the latitude and longitude property values; however, it can also be specified in the following ways:

- The map's initialAddress property, if the latitude and longitude are both set to 0. You can format the address as a natural-language address; you can concatenate multiple lines into a single line with spaces (for example, "600 108th Ave NE, Bellevue, WA 98004 USA"). You can specify a two-letter ISO-3166 country code using the initialCountry property as a country/locale hint, though it is generally not needed.
- Auto-centering among the map's initial markers, if the map's latitude and longitude are both set to 0, and its initial Address is blank (and the map has markers).

You can add a variety of controls to the Google Map control:

- Use the "Map Navigation Control" on page 444 adds map navigation controls (such as a zoom control, map type control, and so forth).
- Use "Map Marker" on page 438 children of a "Map Marker Group" on page 443 control to add individual markers to the map.
- Use a "Dynamic Map Marker List" on page 426 control to add a list of markers from a dynamic data source (such as an array of rows from a SQL result set or an array of items from a Web service result) to the map.

Scripting

The Google Maps API is a full-featured, client-side JavaScript API, and is fully accessible for use with the CAF Google Maps controls. See http://www.google.com/apis/maps/ for the Google Maps API documentation.

To access the GMap2 object (the Google Maps API map object) associated with a Google Map control, use the map property of the control's CAF client-side model object. The following example looks up the GMap2 object of the Google Map control model with an ID of myMap, and uses the Google Maps API to zoom in one level:

```
var map = CAF.model("#{activePageBean.clientIds["myMap"]}").map;
map.zoomIn();
```

Client-Side Model

The CAF client-side model object for the Google Map control extends from the CAF.Input.Model object (see "Client-Side Model" on page 201 for more information about the CAF client-side model). The value of the model object is the coordinates of the map's center in string form (for example, 37.0625;-95.67706). The following script centers a Google Map control with an ID of myMap over the city of Bellevue, Washington:

```
CAF.model("#{activePageBean.clientIds["myMap"]}").setValue("37.0625;-95.677068");
```

The CAF.GMap.stringToGLatLng() and CAF.GMap.gLatLngToString() functions convert string coordinates to and from a GLatLng object (the Google Maps API coordinates object).

You can also use the CAF client-side model to set a map's center with a natural-language address. The following script will also center a Google Map control with an ID of myMap over the city of Bellevue, Washington:

```
CAF.model("#{activePageBean.clientIds["myMap"]}").setAddress("600 108th Ave NE, Bellevue, WA 98004");
```

You can also auto-center a map among the map's markers with the client-side model's autoCenter() method. The following script centers a Google Map control with an ID of myMap among the markers on the map:

```
CAF.model("#{activePageBean.clientIds["myMap"]}").autoCenter(true);
```

Pass a value of true to the autoCenter() method to automatically zoom so that all markers on the map are visible; pass a value of false (or nothing) to auto-center without changing the zoom level.

Script Controls

You can also use the standard Script-category controls like "Return Value Script" on page 500 or "Invoke Script" on page 491 to manipulate the map with the standard client-side actions (such as getValue or setValue). You can also use the custom "Return Map Value Script" on page 446 and "Invoke Map Script" on page 435 controls to access special map-only properties and actions such as getLatitude or setAddress. Additionally, the "Return Map Coords Script" on page 445 control allows latitude and longitude coordinates to be specified separately when invoking the setValue or setCoords action.

JSF Properties

Name	Display Name	Туре	Description	Choices	Default		
General Properties							
id	ID	String	Control id.				
rendered	Rendered	boolean	True to display contr	ol.	true		
key	Key	String	Google Maps API ke http://www.google.co				
Display Properties							
height	Height	String	Map height in pixels	(100) or percent (10	00%).		
width	Width	String	Map width in pixels	(100) or percent (10	0%).		
Value Properties							
initialAddress	Initial Address	String	Initial center of map example, "600 108th		Č ,		
initialCountry	Initial Country	String	Two-letter country co (for example, "US").	ode (ISO 3166) hint	for the initial address		
latitude	Latitude	double	Latitude of map cent	er in degrees (for ex	kample, 37.0625).		
longitude	Longitude	double	Longitude of map ce	nter in degrees (for	example, -95.677068).		

Name	Display Name	Туре	Description	Choices	Default		
type	Туре	String	Map type.	G_NORMAL_MAP, G_SATELLITE_MAP, G_HYBRID_MAP	G_NORMAL _MAP		
zoom	Zoom	int	Zoom level, from 1 (whole world) to about 20 (city block); 0 signals zoom-to-fit markers.				
Expert Display Pro	Expert Display Properties						
Continuous Zoom	Continuous Zoom	boolean	True to enable continuous smooth zooming.				
Dragging	Dragging	boolean	True to enable the user to re-center the map by dragging it.				
DoubleClick Zoom	Double Click Zoom	boolean	True to enable the us the map.	er to zoom in or out by do	uble-clicking		
InfoWindow	Info Window	boolean	True to enable an info	ormation (bubble) window	7.		
ScrollWheel Zoom	Scroll Wheel Zoom	boolean	True to enable the user to zoom the map using a mouse scroll wheel.				
style	CSS Style	String	HTML "style" attribute value.				
styleClass	CSS Class	String	HTML "class" attribute value.				
title	Element Title	String	HTML "element" att	ribute value, often display	ed as a ToolTip.		

"Map Navigation Control" on page 444, "Dynamic Map Marker List" on page 426, "Map Marker Group" on page 443, or "Invoke Map Script" on page 435.

Facets

Google Map Key

com.webmethods.caf.faces.gmap.GMapKey

Base Controls Component Library

A control that includes the Google Maps API into the page. This control is only needed if the main "Google Map" on page 431 control is not rendered during the initial page load.

JSF Properties

Name	Display Name	Туре	Description	Default		
General Properties						
id	ID	String	Control id.			
rendered	Rendered	boolean	True to display control.	true		
key	Key	String	Google Maps API key. Register for a key at http://www.google.com/apis/maps/signup.html .			

Children

None.

Facets

None.

Invoke Map Script

com.webmethods.caf.faces.gmap.GMapInvokeScript

Base Controls Component Library

A control that connects a standard client-side JavaScript control-model action to a parent control's (or specified "for" control's) client-side event. The following standard actions are available:

- raise: actuates the command control specified by the "control" property.
- setDisabled: enables or disables the input control specified by the "control" property, based on the value of the "value" property or "value" facet (true to disable, false to enable).
- setValue: sets the value of the control specified by the "control" property to the value of the "value" property or "value" facet.

- setVisible: shows or hides the control specified by the "control" property, based on the value of the "value" property or "value" facet (true to show, false to hide).
- toggle: shows the control specified by the "control" property if the control is hidden, hides the control specified by the "control" property if the control is visible

In addition, the following map-specific actions are available:

- setCoords: (map or marker) sets the map's center or the marker's location to the specified string coordinate pair (for example, "37.0625;-95.677068:).
- setAddress: (map or marker) sets the map's center or the marker's location to the specified natural-language address (for example, "600 108th Ave NE Bellevue, WA 98004").
- setZoom: (map-only) sets the map's zoom level to a number between 1 (whole world) and about 20 (city block).
- setMapType: (map-only) sets the map-type to either G_NORMAL_MAP, G_SATELLITE_MAP, or G_HYBRID_MAP.
- autoCenter: (map-only) centers the map among the map's markers.
- autoZoom: (map-only) zooms the map to fit the map's markers.
- setLabel: (marker-only) sets the marker's ToolTip to the specified value.
- setDescription: (marker-only) sets the contents of the marker's information (bubble) window to the specified value.
- showInfoWindow: (marker-only) opens the marker's information (bubble) window.
- hideInfoWindow: (marker-only) closes the marker's information (bubble) window.

Other script controls can be nested inside the "value" facet of this script control to provide a value for actions that require a value.

Name	Display Name	Туре	Description	Choices	Default
General Proper	ties				
id	ID	String	Optional cont	rol id.	
rendered	Rendered	boolean	true if the cor rendered.	trol is rendered; false if it is not	true
for	For	Control ID reference ("Control ID Reference" on page 192)		cript to an event for specified controscript control).	ol (defaults to

Name	Display Name	Туре	Description	Choices	Default
Script					
event	Event	String	The client event that should trigger this script.	standard control events:none, onclick, onchange, ondblclick, onmousedown, onmouseup, onmousemove, onmouseover, onmouseout, onkeydown, onkeypress, onkeyup, onRowChange, onValueChange, onDrag, onDrop.	onclick
				map events: map-addmaptype, map-removemaptype, map-click, map-dblclick, map-singlerightclick, map-movestart, map-move, map-moveend, map-zoomend, map-maptypechanged, map-infowindowopen, map-infowindowbeforeclose, map-infowindowclose, map-addoverlay, map-removeoveraly, map-clearoverlays, map-mouseover, map-mouseout, map-mousemove, map-dragstart, map-drag, map-dragend, map-load.	
				marker events: marker-click, marker-dblclick, marker- mousedown, marker-mouseup, marker-mouseover, marker- mouseout, marker- infowindowopen, marker- infowindowbeforeclose, marker- infowindowclose, marker- remove, marker-dragstart, marker-drag, marker-dragend, marker-visibilitychanged.	
action	Action	String	An action to invoke on the control model.	toggle, raise, setValue, setVisible, setDisabled, setCoords, setAddress, setZoom, setMapType, autoCenter, autoZoom, setLabel, setDescription, showInfoWindow, hideInfoWindow.	setValue

Name	Display Name	Туре	Description	Choices	Default
control	Control	String	action on. If a example to sh "Control Refe	ession or ID of the control to invoke ction is being invoked on multiple cow or hide multiple controls on the erence" on page 490 controls can be elivoke Map Script control to defir	controls (for page) dropped as
value	Value	String	value can be	assed to the control model action. Al taken from another script or the "Pa trol, and dropped in the "value" face	rameter" on

The "Control Reference" on page 490 control can specify multiple controls to invoke client-side model actions.

Facets

Name	Display Name	Description	Allowed Types
value	Value	Specifies a value for the actions that require a value (the "value" property can also be used to specify a value).	"Parameter" on page 494 "Custom Script" on page 490 "Return Value Script" on page 500 "Portlet Url Script" on page 496 "Return Map Value Script" on page 446 "Return Map Coords Script" on page 445

Map Marker

com.webmethods.caf.faces.gmap.GMarker

Base Controls Component Library

A control that provides an individual marker in a "Map Marker Group" on page 443 control. The marker's location normally is specified using its latitude and longitude property values but it can also be specified using the marker's initial Address property if the latitude and longitude are both set to 0. The address can be formatted as a natural-language address with multiple lines concatenated into a single line with spaces (for example, "600 108th Ave NE, Bellevue, WA 98004 USA). A two-letter ISO-3166 country code can be specified using the initial Country property as a country/locale hint, although it is generally not needed.

The marker's icon can be specified using the Icon category properties. You can use the standard CAF marker icons by configuring the marker's icon property with the icon name (red, orange, yellow, green, cyan, blue, purple, or magenta), or you can use a custom icon by configuring the icon property with the URL of the icon (for example,

http://maps.google.com/mapfiles/arrow.png). You can configure the marker's selected icon similarly using the selectedImage property. If you use a custom icon, you can also configure many of the expert Icon properties, such as iconSizeHeight, iconSizeWidth, shadow (the URL of the shadow image), shadowSizeHeight, shadowSizeWidth, and so forth. With the standard CAF marker icons, you can leave these properties blank.

The content of the marker is used to render the marker's information (bubble) window. A user can click a marker to display its information window, and also to select the marker, if the marker's clickable property is set to true. A user can drag the marker to a new location if the marker's draggable property is set to true.

Scripting

The Google Maps API is a full-featured, client-side JavaScript API, and is fully accessible for use with the CAF Google Maps controls. See http://www.google.com/apis/maps/ for the Google Maps API documentation.

To access the GMarker object (the Google Maps API marker object) associated with this control, use the getOverlay() method of the control's CAF client-side model object. The following example looks up the GMarker object of the MapMarker control model with an ID of myMarker, and uses the Google Maps API to hide the marker:

```
var marker = CAF.model("#{activePageBean.clientIds["myMarker"]}").getOverlay();
marker.hide();
```

Client-Side Model

The CAF client-side model object for this control extends from the CAF.Table.Row.Model object (see "Client-Side Model" on page 201 for more information about the CAF client-side model). The object's value is the marker's ID. You can get the coordinates of the marker's location, in string form (for example, 37.0625;-95.677068) by using the getCoords() method of the client-side model. The following script gets the coordinates of the marker with an ID of myMarker:

```
var coords = CAF.model("#{activePageBean.clientIds["myMarker"]}").getCoords();
```

The CAF.GMap.stringToGLatLng() and CAF.GMap.gLatLngToString() functions convert string coordinates to and from a GLatLng object (the Google Maps API coordinates object). To set the coordinates, you must change the marker model's coordinates setting using the model's setCoords() method and then update the marker-group with the changed marker model using the marker-group model's set() method. The following script sets the marker with an ID of myMarker in a marker-group with an ID of myMarkerGroup:

```
CAF.model("#{activePageBean.clientIds["myMarkerGroup"]}")
.set(CAF.model("#{activePageBean.clientIds["myMarker"]}").setCoords("37.0625;-95.677068"));
```

You can use the CAF client-side model to set a marker's location with a natural-language address. The following script sets the location of the marker with an ID of myMap:

Script Controls

You can use the standard Script-category controls like "Return Value Script" on page 500 or "Invoke Script" on page 491 to manipulate a marker with the standard client-side actions (such as getValue). You can also use the custom "Return Map Value Script" on page 446 and "Invoke Map Script" on page 435 controls to access special map-only properties and actions such as getLatitude or setAddress. Additionally, the "Return Map Coords Script" on page 445 control allows latitude and longitude coordinates to be specified separately when invoking the setValue or setCoords action.

JSF Properties

Name	Display Name	Туре	Description Choices		Default
General Properties	5				
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display control.		true
Display Properties	i				
title	Title	String	Marker ToolTip title.		
Value Properties					
initialAddress	Initial Address	String	Initial marker location if latitude=0 and longitude=0 (for example, 600 108th Ave NE, Bellevue, WA 98004).		
initialCountry	Initial Country	String	Two-letter country code (ISO 3166) hint for initial address (for example, US).		
latitude	Latitude	double	Latitude of marker location in	degrees (for example, 37	.0625).
longitude	Longitude	double	Longitude of marker location is 95.677068).	n degrees (for example,	-
selected	Selected	boolean	True if marker is selected.		false
Icon Properties					
icon	Icon	String	URL of the marker icon (for example, http://maps.google.com/mapf iles/arrow.png), or standard icon name (for example, red).	red, orange, yellow, green, cyan, blue, purple, magenta.	red

Name	Display Name	Туре	Description	Choices	Default
iconSize Height	Icon Height	int	Icon height in pixels.		
iconSizeWidth	Icon Width	int	Icon width in pixels.		
selectedImage	Selected Image	String	URL of icon to use when marker is selected, or standard icon name.	red, orange, yellow, green, cyan, blue, purple, magenta	yellow
Behavior Propertion	es				
clickable	Clickable	boolean	True to allow the user to click to open the information window.	he marker to select it or	false
draggable	Draggable	boolean	True to allow the user to drag t location.	he marker to a different	false
Expert Icon Prope	rties				
dragCross AnchorX	Drag Cross Anchor X	int	Horizontal offset of the cross image from the icon anchor position in pixels.		
dragCross AnchorY	Drag Cross Anchor Y	int	Vertical offset of the cross image from the icon anchor position in pixels.		
dragCross Image	Drag Cross Image	String	URL of the image to display over the exact marker location when the marker is dragged.		
dragCross AnchorHeight	Drag Cross Anchor Height	int	Height of the cross image in pi	xels.	
dragCros sAnchorWidth	Drag Cross Anchor Width	int	Width of the cross image in pix	cels.	
iconAnchorX	Icon Anchor X	int	Horizontal offset of the left edglocation in pixels.	ge of the icon from the m	arker
iconAnchorY	Icon Anchor Y	int	Vertical offset of the top edge of the icon from the marker location in pixels.		
imageMap	Image Map	String	Comma-separated list of integers representing the x/y coordinates of the image map to specify the clickable part of the icon image in browsers other than Internet Explorer.		
infoWindow AnchorX	Info Window Anchor X	int	Horizontal offset of the pointed tip of the information window from the left edge of the icon in pixels.		

Name	Display Name	Туре	Description Choices	Default
infoWindow AnchorY	Info Window Anchor Y	int	Vertical offset of the pointed tip of the information wir from the top edge of the icon in pixels.	ndow
mozPrint Image	Mozilla Print Image	String	URL of the icon to use for printing in Firefox.	
printImage	Print Image	String	URL of the icon to use for printing in browsers other the Firefox.	nan
printShadow	Print Shadow	String	URL of the image to use for the marker's shadow when	printing.
shadow	Shadow	String	URL of the image to use for the marker's shadow.	
shadowSize Height	Shadow Size Height	int	Height of the shadow in pixels.	
shadowSize Width	Shadow Size Width	int	Width of the shadow in pixels.	
transparent	Transparent	String	URL of a virtually transparent version of the foreground image used to capture click events in Internet Explorer	
Behavior Propertie	es			
autoPan	Auto Pan	boolean	True to auto-pan the map if the user drags the marker near the map's edge.	true
bounceGravity	Bounce Gravity	int	Acceleration rate of the marker when bouncing.	1
bouncy	Bouncy	boolean	True to make the marker bounce up and down after the user finishes dragging it.	true
dragCross Move	Drag Cross Move	boolean	True to keep the marker under the cursor when dragging; false to keep the drag-cross under the cursor when dragging.	false
maxHeight	Max Height	int	Height in pixels for the marker to raise when dragged.	

Any. Children will be rendered as the content of the marker's information (bubble) window.

Facets

Map Marker Group

com.webmethods.caf.faces.gmap.GMarkerGroup

Base Controls Component Library

A container for individual "Map Marker" on page 438 controls. To be used as a child of the "Google Map" on page 431 control.

Scripting

The CAF client-side model object for this control extends from the CAF. Table. Model object (see "Client-Side Model" on page 201 for more information about the CAF client-side model). The object's value is the selected marker's ID (or if this control's multiple property is set to true, an array of selected marker IDs). For example, you can set the selected marker of the marker-group with an ID of myMarkerGroup to the marker with an ID of myMarker with the following script:

CAF.model("#{activePageBean.clientIds["myMarkerGroup"]}").setValue("myMarker");

JSF Properties

Name	Display Name	Туре	Description	Default			
General Properties							
id	ID	String	Control id.				
rendered	Rendered	boolean	True to display the control.	true			
multiple	Multiple	boolean	True if multiple markers can be selected.	false			

Children

"Map Marker" on page 438.

Facets

Map Navigation Control

com.webmethods.caf.faces.gmap.GMapControl

Base Controls Component Library

A control that adds a navigation control to a Google map created using the "Google Map" on page 431 control. The following types are supported:

- GSmallMapControl: a condensed set of buttons for navigation (north, south, east, west) and zoom (in, out).
- GLargeMapControl: full-size navigation and zoom buttons.
- GSmallZoomControl: zoom-in and zoom-out buttons.
- GScaleControl: a geographic scale indicator (for example, 1 inch = 100 miles).
- GMapTypeControl: map-type buttons (normal, satellite, hybrid).
- GOverviewMapControl: a mini-map cut-out that displays a zoomed-out version of the main map view.

JSF Properties

Name	Display Name	Туре	Description	Choices	Default		
General Pro	perties						
id	ID	String	Control id.				
rendered	Rendered	boolean	True to display	control.	true		
type	Туре	String	Navigation control type.	GSmallMapControl, GLargeMapControl, GSmallZoomControl, GScaleControl, GMapTypeControl, GOverviewMapCo			
Display Pro	perties						
anchor	Anchor	String	Position of control on the map.	G_ANCHOR_TOP_RIGHT, G_ANCHOR_TOP_LEFT, G_ANCHOR_BOTTOM_RIGHT, G_ANCHOR_BOTTOM_LEFT	(defaults to the customary position for the specific control type)		
offsetX	Offset X	int	Horizontal offset from the corner of the map specified by the anchor property to this control, in pixels (for example, 10).				
offsetY	Offset Y	int		from the corner of the map specified by the control in pixels (for example, 10).	he anchor		

None.

Facets

None.

Return Map Coords Script

com. we bmethods. caf. faces. gmap. GMap Get Coords Script

Base Controls Component Library

A control that creates a client-side JavaScript function to return to the "Google Map" on page 431 control the specified latitude and longitude coordinate pair as a string. Usually this script is used to define value for the following script controls:

- "Parameter" on page 494
- "Invoke Script" on page 491
- "Invoke Map Script" on page 435

You can specify the latitude and longitude using the latitude and longitude properties or the latitude and longitude facets.

Properties

Name	Display Name	Туре	Description Default	
General Propert	ies			
id	ID	String	Control id.	
rendered	Rendered	boolean	True to display control.	true
Script Propertie	S			
latitude	Latitude	double	Latitude value.	
longitude	Longitude	double	Longitude value.	

Children

Facets

Name	Display Name	Description	Allowed Types
latitude	Latitude	Latitude value.	"Custom Script" on page 490 "Return Value Script" on page 500 "Return Map Value Script" on page 446
longitude	Longitude	Longitude value.	"Custom Script" on page 490 "Return Value Script" on page 500 "Return Map Value Script" on page 446

Return Map Value Script

com.webmethods.caf.faces.gmap.GMapGetValueScript

Base Controls Component Library

A control that creates a client-side JavaScript function to return values to a "Google Map" on page 431 control from client-side JavaScript control models. Usually this script is used to define value for the following script controls:

- "Parameter" on page 494
- "Invoke Script" on page 491
- "Return Map Value Script" on page 446

The following standard actions are available:

- isDisabled: true if the control is disabled; false if the control is enabled.
- isVisible: true if the control is visible; false if the control is hidden.
- getValue: gets the control's value.

In addition, the following map-specific actions are available:

- getCoords: (map or marker) gets the map's center or the marker's location as a string coordinate pair (for example, "37.0625;-95.677068").
- getLatitude: (map or marker) gets the latitude of the map's center or the marker's location as a floating-point number (for example, 37.0625).
- getLongitude: (map or marker) gets the longitude of the map's center or the marker's location as a floating-point number (for example, -95.677068).
- getZoom: (map-only) gets the map's zoom level.
- getMapType: (map-only) gets the map-type.
- getLabel: (marker-only) gets the marker's ToolTip.

- getDescription: (marker-only) gets the contents of the marker's information (bubble) window.
- isSelected: (marker-only) true if the marker is selected; false if the marker is not selected.

Name	Display Name	Туре	Description	Choices	Default
General Pro	perties				
id	ID	String	Optional control id.		
rendered	Rendered	boolean	Set to true if this control is rendered. true		
for	For	Control ID reference ("Control ID Reference" on page 192)	Connect the script to an event for the specified control (defaults to the parent of the script control).		
Script					
event	Event	String	The client event that should trigger this script. Not needed if this control is used as a child or facet of another script control.	onclick, onchange, ondblclick, onmousedown, onmouseup, onmousemove, onmouseover, onmouseout, onkeydown, onkeypress, onkeyup.	onclick
action	Action	String	The action to invoke on the control's model.	getValue, isVisible, isDisabled, getCoords, getLatitude, getLongitude, getZoom, getMapType, getLabel, getDescription, isSelected	getValue
control	Control	String	Binding expression or ID of the control to invoke the model action on. If the action is being invoked on multiple controls (for example to get the visible state of multiple controls on the page) "Control Reference" on page 490 controls can be dropped as children to the Return Map Value Script control to define multiple controls.		

The "Control Reference" on page 490 control can be used to specify multiple controls to get the value of.

Facets

Chapter 19. Panel Controls

Access Control Panel
Attachments Panel
Block Edge Panel
Block Panel
Disableable Panel
Grid Panel
Hideable Panel
Inline Hideable Panel
Inline Panel
Overlay Panel
Page Group
Popup Panel
Property Group
Property Line
Property Sub-Group
Scrolling Panel
Stack Panel
Static Cell
Static Row
Submit Group
Titlebar Tabs Wrapper
Tooltip

Access Control Panel

com. we bmethods. caf. faces. panel. Access Control Panel

Base Controls Component Library

Controls access (display) to a Panel based on Security Role Membership.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
allowed- security-role	Allowed Security Role	String	A optional binding expression to a Security Role that the current user must be a member of in order to see the contents of this Panel.	
denied-security- role	Denied Security Role	String	A optional binding ear a Security Role that the user must not be a morder to see the context.	he current ember of in

Children

Any.

Facets

Attachments Panel

com.webmethods.caf.faces.panel.AttachmentsPanel

Base Controls Component Library

A Complex control that allows a user to view a list of files, and optionally add to or remove from that list. The list of files is referenced by the control's attachmentsProvider property, which must be bound to a com.webmethods.caf.faces.data.attachments.IAttachmentsProvider instance.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to include control in rendered page.	true
attachments Provider	Attachments Provider	IAttachments Provider	Model object which provides the list of attachments.	
defaultEncoding	Default Encoding	String	Default encoding option in the Add Attachment dialog.	
readOnly	Read Only	boolean	True if user cannot modify list of attachments, only view them.	false

Children

None.

Facets

Block Edge Panel

com.webmethods.caf.faces.panel.BlockEdgePanel

Base Controls Component Library

Displays the contents of the panel in a block-level HTML element (a DIV). Controls in the 'leftEdge' facet are rendered docked to the left edge of the panel. Controls in the 'rightEdge' facet are rendered docked to the right edge of the panel.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Expert Display Prop	erties			
style	CSS Style	String	HTML "style" attribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.	
title	Title	String	HTML "title" attribu (often displayed as a	

Children

Any.

Facets

Name	Display Name	Description	Allowed Types
leftEdge	Left Edge	Container for controls that should be docked to the left edge of the panel	Any
rightEdge	Right Edge	Container for controls that should be docked to the right edge of the panel	Any.

Block Panel

com. we bmethods. caf. faces. panel. Block Panel

Base Controls Component Library

Displays the contents of the panel in a block-level HTML element (a DIV).

Properties

Name	Display Name	Туре	Description	Default	
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control.	true	
Expert Display Properties					
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attribute value.		
title	Title	String	HTML "title" attribute value (often displayed as a tooltip).		

Children

Any.

Facets

Disableable Panel

com.webmethods.caf.faces.panel.DisableablePanel

Base Controls Component Library

Displays the contents of the panel in a block-level HTML element (a DIV). This panel also has a "disabled" property, which affects the client-side "disabled" state of the contained controls (but not the server-side state). The client-side "disabled" state of this and other controls can be toggled with the setDisabled(disabled) method of the control's client-side model. See "Client-Side Model" on page 201 for more information.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display contained controls as "grayed-out" and uneditable.	false
Expert Display Prop	erties			
style	CSS Style	String	HTML "style" attribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.	
title	Title	String	HTML "title" attribute v displayed as a tooltip).	alue (often

Children

Any.

Facets

Grid Panel

javax.faces.grid

CAF JSF Base Controls Component Library

A grid container for other components. A standard JavaServer Faces control. This control is deprecated. Instead, use "Static Row" on page 474 and "Static Cell" on page 473.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	trol.	true
Display Properties					
disabled	Disabled	boolean	True to display contained controls as false "grayed-out" and un-editable.		false
Expert Display Prop	erties				
bgcolor	Background Color	Color	Table background color.		
border	Border	int	Border width in pixels, for both outer table border and borders between cells.		
cellpadding	Cell Padding	String	Padding within cell content between margin and text.		
cellspacing	Cell Spacing	String	Margin between cel	ll border and cell con	tent.
columnClasses	Column CSS Classes	Collection	CSS class to use for each column, from left to right (not repeating).		
customRow Class	Custom Row CSS Class	String	Binding expression that specifies a custom CSS class name to add to the current row.		
dir	Directionality	String	Text direction LTR (left-to-right), RTL (righ hint. to-left)		RTL (right-
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-foot	ter

Name	Display Name	Туре	Description	Choices	Default	
frame	Frame	String	Outer border style.	void (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)	void	
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-subl	header	
lang	Language	String	Language code for	content.		
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	portlet-section-bod section-alternate	y, portlet-	
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	void	
style	CSS Style	String	HTML "style" attri	bute value.		
styleClass	CSS Class	String	HTML "class" attri	bute value.		
summary	Summary	String	Summary text.			
title	Title	String	HTML "title" attrib	ute value.		
width	Width	String	Width of table in pi	xels ('100') or percent	('100%').	
Expert Client Side	Events					
onclick	Click	script	Control clicked.			
ondblclick	Double Click	script	Control double-clic	Control double-clicked.		

Name	Display Name	Туре	Description Choices Default		
onkeydown	Key Down	script	Key pressed when control is focused.		
onkeypress	Key Press	script	Key pressed and released when control is focused		
onkeyup	Key Up	script	Key released when control is focused.		
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved when over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundaries.		
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		

Any.

Facets

None.

Hideable Panel

com.webmethods.caf.faces.panel.HideablePanel

Base Controls Component Library

A Block panel that can be toggled between visible and hidden through client-side JavaScript code. Special "toggle" controls encapsulate this code within controls that can be created and configured visually, using the CAF Eclipse plug-in. See "Hideable Controls" on page 194 for more information about hideable controls. See "Toggle Controls" on page 197 for more information about toggle controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to include control in rendered page (note that setting this to false will prevent the control from being toggled visible).		true

Name	Display Name	Туре	Description Choices	Default
Display Properties				
default Command	Default Command	"Control ID Reference" on page 192	ID of command control to submit if use key when control content is focused.	r presses enter
defaultFocus	Default Focus	"Control ID Reference" on page 192	ID of control to focus when control is toggled visible. _first (first input control contained by this control), other	_first
disableWhen Hidden	Disabled When Hidden	boolean	True to disable contained input controls when control is hidden, preventing them from submitting.	false
visible	Visible	boolean	True to display control as initially visible.	true
Lazy Load Propertie	es .			
lazy	Lazy Load	boolean	True to load control's content asynchronously on demand.	false
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog.	0
progressMsg	Progress Message	String	Message to display in progress dialog.	Defaults to localized message; in English, "Loading"
refreshOnShow	Refresh On Show	boolean	True to refresh content every time control is toggled visible; false to refresh content only if control is initially hidden and then toggled visible.	false
suppressInputs	Suppress Inputs	Collection	List of ids of controls for which to disable validation/update processing when handling a asynchronous request (comma separated).	
twoPass	Two Pass	boolean	True to load content asynchronously, even if control is initially visible; false to render control synchronously if control is initially visible.	false

Name	Display Name	Туре	Description	Choices	Default			
Expert Display Prop	Expert Display Properties							
hideEffect	Hide Effect	String	Name of "Scriptace hide this control.	Name of "Scriptaculous Effects" on page 198 to use to hide this control.				
hideEffect Options	Hide Effect Options	String	Custom options to pass to the hide effect, in JavaScript object notation (ex "duration: 2.0, queue: 'front'").					
progressFlash OnComplete	Flash On Load	boolean	True to flash control when content true finishes loading asynchronously.					
progressUse HideShowEffect	Progress Use Hide/Show Effect	boolean	True to use custom effects also when h progress dialog.		false			
showEffect	Show Effect	String	Name of "Scriptacushow this control.	ılous Effects" on pag	e 198 to use to			
hideEffect Options	Show Effect Options	String	Custom options to pass to the show effect, in JavaScript object notation (ex "duration: 2.0, queue: 'front'").					
style	CSS Style	String	HTML "style" attri	bute value.				
styleClass	CSS Class	String	HTML "class" attribute value.					

Any.

Facets

Inline Hideable Panel

com.webmethods.caf.faces.panel.InlineHideablePanel

Base Controls Component Library

An inline panel that can be toggled between visible and hidden through client-side JavaScript code. Special "toggle" controls encapsulate this code within controls that can be created and configured visually, using the webMethods CAF. See "Hideable Controls" on page 194 for more information about hideable controls. See "Toggle Controls" on page 197s for more information about toggle controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to include control in rendered true page (note that setting this to false will prevent the control from being toggled visible).		
Display Properties					
default Command	Default Command	"Control ID Reference" on page 192	ID of command control to submit if user presses enter key when control content is focused.		
defaultFocus	Default Focus	"Control ID Reference" on page 192	ID of control to focus when control is toggled visible.	_first (first input control contained by this control), other	_first
disableWhen Hidden	Disabled When Hidden	boolean	True to disable contained input controls when control is hidden, preventing them from submitting.		false
visible	Visible	boolean	True to display convisible.	trol as initially	true
Lazy Load Propertie	S				
lazy	Lazy Load	boolean	True to load control's content asynchronously on demand.		false
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog.		0

Name	Display Name	Туре	Description	Choices	Default
progressMsg	Progress Message	String	Message to display	in progress dialog.	Defaults to localized message; in English, "Loading"
refreshOnShow	Refresh On Show	boolean	True to refresh cont control is toggled virefresh content only initially hidden and visible.	isible; false to if control is	false
suppressInputs	Suppress Inputs	Collection	validation/update p	ls for which to disable processing when hand est (comma separated	lling an
twoPass	Two Pass	boolean	True to load content even if control is init to render control sy control is initially v	tially visible; false nchronously if	false
Expert Display Prop	erties				
hideEffect	Hide Effect	String	Name of "Scriptacu hide this control.	lous Effects" on page	198 to use to
hideEffect Options	Hide Effect Options	String		pass to the hide effect duration: 2.0, queue:	•
progressFlash OnComplete	Flash On Load	boolean	True to flash contro finishes loading asy		true
progressUse HideShowEffect	Progress Use Hide/Show Effect	boolean	True to use custom effects also when hi progress dialog.		false
showEffect	Show Effect	String	Name of "Scriptacu show this control.	lous Effects" on page	198 to use to
hideEffect Options	Show Effect Options	String		pass to the show effect tation (ex "duration:	
style	CSS Style	String	HTML "style" attril	oute value.	
styleClass	CSS Class	String	HTML "class" attril	oute value.	

Any.

Facets

None.

Inline Panel

javax.faces.panel

CAF JSF Base Controls Component Library

An inline container for other components. A standard JavaServer Faces control.

Properties

Name	Display Name	Туре	Description	Choices	Default		
General Properties							
id	ID	String	Control ID.				
rendered	Rendered	boolean	True to display cor	ntrol.	true		
Expert Display Properties							
title	Element Title	String	HTML "title" attrib tooltip.	HTML "title" attribute value (often displayed as a tooltip.			
dir	Directionality	String	Text direction hint.	(==================================			
lang	Language	String	Language code for	Language code for content.			
style	CSS Style	String	HTML "style" attribute value.				
styleClass	CSS Class	String	HTML "class" attribute value.				

Children

Any.

Facets

Overlay Panel

com.webmethods.caf.faces.panel.OverlayPanel

Base Controls Component Library

A panel that overlays another panel (or the entire page) when toggled visible through client-side JavaScript code. Special "toggle" controls encapsulate this code within controls that can be created and configured visually, using the webMethods CAF. See "Hideable Controls" on page 194 for more information about hideable controls. See "Toggle Controls" on page 197 for more information about toggle controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	ID of control to ove page).	erlay (or empty to ov	erlay entire
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to include control in rendered true page (note that setting this to false will prevent the control from being toggled visible).		
Display Properties					
default Command	Default Command	"Control ID Reference" on page 192	ID of command control to submit if user presses entekey when control content is focused.		
defaultFocus	Default Focus	"Control ID Reference" on page 192	ID of control to focus when control is toggled visible.	_first (first input control contained by this control), other	_first
disableWhen Hidden	Disabled When Hidden	boolean	True to disable contained input controls when control is hidden, preventing them from submitting.		false
Lazy Load Propertie	es				
lazy	Lazy Load	boolean	True to load control's content false asynchronously on demand.		false
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog.		0

Name	Display Name	Туре	Description	Choices	Default	
progressMsg	Progress Message	String	Message to display	in progress dialog.	Defaults to localized message; in English, "Loading"	
refreshOnShow	Refresh On Show	boolean	True to refresh concontrol is toggled verifiesh content only initially hidden and visible.	risible; false to y if control is	false	
suppressInputs	Suppress Inputs	Collection	List of ids of controls for which to disable validation/update processing when handling an asynchronous request (comma separated).			
twoPass	Two Pass	boolean	True to load conter even if control is in to render control sy control is initially v	itially visible; false nchronously if	false	
Expert Display Prop	erties					
center	Center	double	True to center conte	ent.	true	
hideEffect	Hide Effect	String	Name of "Scriptaculous Effects" on page 198 to use to hide this control.	Effect.CAF.Fade		
hideEffect Options	Hide Effect Options	String		pass to the hide effect otation (ex "duration		
opacity	Opacity	double	Opacity of panel ar 0.0 (completely tran (completely opaque	nsparent) to 1.0	1.0	
progressFlash OnLoad	Flash On Load	boolean	True to flash contro finishes loading asy		true	
progressUse HideShowEffect	Progress Use Hide/Show Effect	boolean	True to use custom effects also when h progress dialog.		false	

Name	Display Name	Туре	Description	Choices	Default
showEffect	Show Effect	String	Name of "Scriptaculous Effects" on page 198 to use to show this control.	Effect.CAF.A	ppear
showEffect Options	Show Effect Options	String	Custom options to pass to the show effect, in JavaScript object notation (ex "duration: 2.0, queue: 'front'").		
style	CSS Style	String	HTML "style" attr	ibute value.	
styleClass	CSS Class	String	HTML "class" attr	ibute value.	
	Children				
	Any.				

Page Group

com.webmethods.caf.faces.panel.PageGroup

Base Controls Component Library

A panel that renders a portlet-like grouping UI around a large section of content.

Note that the "disabled" property of this control affects only the client-side "disabled" state of contained controls (and not the server-side state). The client-side "disabled" state of this and other controls can be toggled with the setDisabled(disabled) method of the control's client-side model. See "Client-Side Model" on page 201 for more information.

Properties

Facets None.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description	Default
Display Properties				
style	CSS Style	String	HTML "style" attribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.	
disabled	Disabled	boolean	True to display false contained controls as "grayed-out" and uneditable.	
label	Label	String	Group label (prepended facet content).	to label

Any.

Facets

	Name	Display Name	Description	Allowed Types
	label	Label	Group label (appended to label property content).	Any

Popup Panel

com.webmethods.caf.faces.panel.PopupPanel

Base Controls Component Library

A panel that pops up when a second control is clicked; the second control is specified by this control's "for" property.

The suggested CSS class for this control is "portlet-menu". To make a vertical list of links look more like a popup menu, try setting the CSS class to "caf-popup-menu".

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	ID of control for which (when clicked) to display the popup panel.		
hover	Hover	String	ID of control for wh to display the popu	nich, when the mouse p panel.	hovers over it,

Display Name	Туре	Description	Choices	Default
ID	String	Control ID.		
Rendered	boolean	True to display control. true		true
Position	boolean	Position, relative to the 'for' control, to display the popup panel.	left, top, right, bottom, over	bottom
erties				
Directionality	String	Text direction LTR (left-to-right), RTL (right-to-hint. left)		
Language	String	Language code for content.		
CSS Style	String	HTML "style" attribute value.		
CSS Class	String	HTML "class" attribute value.		
Element Title	String	HTML "title" attribute value (often displayed as a tooltip.		
	ID Rendered Position Perties Directionality Language CSS Style CSS Class	ID String Rendered boolean Position boolean Perties Directionality String Language String CSS Style String CSS Class String	ID String Control ID. Rendered boolean True to display con Position boolean Position, relative to the 'for' control, to display the popup panel. Perties Directionality String Text direction hint. Language String Language code for CSS Style String HTML "style" attributed to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel. Position Position, relative to the 'for' control, to display the popup panel.	ID String Control ID. Rendered boolean True to display control. Position boolean Position, relative left, top, right, to the 'for' control, bottom, over to display the popup panel. Perties Directionality String Text direction LTR (left-to-right), hint. left) Language String Language code for content. CSS Style String HTML "style" attribute value. CSS Class String HTML "class" attribute value. Element Title String HTML "title" attribute value (often displayed)

Any.

Facets

None.

Property Group

com. we bmethods. caf. faces. panel. Property Group

Base Controls Component Library

Panel that renders a grouping UI around a group of related PropertyLines ("Property Line" on page 469).

Note that the "disabled" property of this control affects only the client-side "disabled" state of contained controls (and not their server-side state). The client-side "disabled" state of this and other controls can be toggled with the setDisabled(disabled) method of the control's client-side model. See "Client-Side Model" on page 201 for more information.

Properties

Name	Display Name	Туре	Description	Default		
General Properties						
id	ID	String	Control ID.			
rendered	Rendered	boolean	True to display control.	true		
Display Properties						
style	CSS Style	String	HTML "style" attribute value.			
styleClass	CSS Class	String	HTML "class" attribute value.			
description	Description	String	Extended description.			
disabled	Disabled	boolean	True to display contained controls as "grayed-out" and un-editable.	false		
label	Label	String	Group label (preper label facet content).	nded to		
Expert Display Properties						
propertyLine LabelWidth	Property Line Label Width	String	Label width for contained property lines in pixels (50px) or percent (50%).			

Children

Any.

Facets

Name	Display Name	Description	Allowed Types
label	Label	Group label (appended to label property content).	Any

Property Line

com.webmethods.caf.faces.panel.PropertyLine

Base Controls Component Library

A panel that displays its contents on one line, with a label and optional extended description, like on a properties page. If the property-line contains a required control, it will indicate this in a skinnable way. If the property-line contains a control that has generated some error or info messages, the property-line will display those messages.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
style	CSS Style	String	HTML "style" attribute	value.
styleClass	CSS Class	String	HTML "class" attribute	value.
description	Description	String	Extended description.	
label	Label	String	Property label.	

Children

Any.

Facets

None.

Property Sub-Group

com.webmethods.caf.faces.panel.PropertySubGroup

Base Controls Component Library

A panel that renders a sub-grouping UI around a group of related PropertyLines ("Property Line" on page 469) or fields.

Note that the "disabled" property of this control affects only the client-side "disabled" state of contained controls (and not their server-side state). The client-side "disabled" state of this and other controls can be toggled with the setDisabled(disabled) method of

the control's client-side model. See "Client-Side Model" on page 201 for more information.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
style	CSS Style	String	HTML "style" attribute	value.
styleClass	CSS Class	String	HTML "class" attribute	value.
description	Description	String	Extended description.	
disabled	Disabled	boolean	True to display contained controls as "grayed-out" and uneditable.	false
label	Label	String	Group label (prepended facet content).	to label
Expert Display Prop	erties			
propertyLine LabelWidth	Property Line Label Width	String	Label width for containe lines in pixels (50px) or p (50%).	

Children

Any.

Facets

Name	Display Name	Description	Allowed Types
label	Label	Group label (appended to label property content).	Any

Scrolling Panel

com.webmethods.caf.faces.panel.ScrollingPanel

Base Controls Component Library

A block panel with a fixed width or height that scrolls to accommodate overflowing content. The height must be specified to scroll vertically; the width must be specified to scroll horizontally.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
height	Height	String	Panel height in pixels (percentage (100%).	100px) or
width	Width	String	Panel width in pixels (percentage (100%).	100px) or
Expert Display Prop	erties			
styleClass	CSS Class	String	HTML "class" attribute	e value.
style	CSS Style	String	HTML "style" attribute value.	
dir	Directionality	String	HTML "dir" attribute to indicate the direction the flow of the content current element.	nality of
title	Element Title	String	HTML "title" attribute value (often displayed as a tooltip).	
lang	Language	String	HTML "lang" attribute value used to specify the language of the enclosed content.	

Any.

Facets

None.

Stack Panel

com.webmethods.caf.faces.panel.StackPanel

Base Controls Component Library

A panel that renders only one of its children. This control's "value" property specifies the ID of the child to render. If no child ID is specified, the first child is rendered.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Value Properties				
value	Value	Object	ID of child to render.	

Children

Any.

Facets

Static Cell

com.webmethods.caf.faces.panel.StaticCell

Base Controls Component Library

Divides a "Static Row" on page 474 into cells. It is best to set the width of each Static Cell to a percentage such that the width of all the cells in row adds up to 100%. Cells without width settings may wrap in peculiar ways.

Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control.		true
Expert Display Properties					
align	Alignment	String	Horizontal text alignment.	left, center, right	left
height	Height	String	Height in pixels (100px) or perce	ent (100%).	
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attribute value.		
width	Width	String	Width in pixels (100px) or percent	nt (100%).	

Children

Any.

Facets

Static Row

com. we bmethods. caf. faces. panel. Static Row

Base Controls Component Library

Displays a static grid row, divided by StaticCells (see "Static Cell" on page 473.).

Properties

Name	Display Name	Туре	Description	Default
General Properti	es			
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Expert Display Properties				
cellmargin	Cell Margin	String	Margin for contained grid cells (1px).	
height	Height	String	Height in pixels (100px) or percent (100%).	
style	CSS Style	String	HTML "style" attribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.	
width	Width	String	Width in pixels (100px) or percent (100%).	

Children

"Static Cell" on page 473.

Facets

Submit Group

com.webmethods.caf.faces.panel.PropertySubmit

Base Controls Component Library

A panel that renders a grouping UI around a group of submit buttons. It encapsulates the standard location for the Submit, Cancel, Previous, and Next buttons, as well as the default width for the buttons. Submit, Cancel, Previous, and Next buttons should be placed in their respective facets; other buttons can be added as non-facet children of the panel.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
style	CSS Style	String	HTML "style" attribute	value.
styleClass	CSS Class	String	HTML "class" attribute	value.

Children

Any.

Facets

Name	Display Name	Description	Allowed Types
submit	Submit	Submit button container.	Any
cancel	Cancel	Cancel button container.	Any
previous	Previous	Previous button container.	Any
next	Next	Next button container.	Any

Titlebar Tabs Wrapper

com.webmethods.caf.faces.panel.TitlebarTabsWrapper

Base Controls Component Library

A panel placed around a "Tabs" on page 411 control that visually merges the Tabs with the Page Group label directly above.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Children

Any.

Facets

None.

Tooltip

com.webmethods.caf.faces.panel.Tooltip

Base Controls Component Library

A panel rendered as tooltip for another control, specified by this control's "for" property.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
click	Click	String	ID of control for which, clicked, to display the to	
for	For	String	ID of control for which (mouse hovers over it) to tooltip.	
id	ID	String	Control ID.	

Name	Display Name	Туре	Description	Default
rendered	Rendered	boolean	True to display control.	true
Expert Display Prop	erties			
style	CSS Style	String	HTML "style" attribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.	

Any.

Facets

Chapter 20. Portlet Controls

Extended Portlet Parameter
Extended Portlet URL
Portlet Include
Portlet Simple Link
Portlet Parameter
Portlet URL

Extended Portlet Parameter

com. we bmethods. caf. faces. portleturl. Extended Portlet Url Parameter

MWS Control Component Library

A control used to specify a portlet parameter for "Extended Portlet URL" on page 481.

Standard JSF Properties

Name	Display Name	Туре	Description	Default
General Properties				
array	Is Array	boolean	Whether supplied value is a commaseparated array	false
encoded	Encoded	boolean	Whether supplied value should be URL encoded.	true
name	Name	String	Name of the portlet preference to set	
value	Value	String	Value binding expression of the value to be set on the portlet	

Children

com.webmethods.caf.faces.portleturl.ExtendedPortletUrl ("Extended Portlet URL" on page 481) children may be used to set parameter value to be another portlet url. This may be used when the parameter to be passed to the target portlet contains the value of the URL for the callee to return back when its finished

Standard JSF Facets

None.

Extended CAF Facets

Extended Portlet URL

com. we bmethods. caf. faces. portleturl. Extended Portlet Url

MWS Control Component Library

This portlet URL control may be used only for My webMethods containers. It provides extensions to build portlet URLs that can address multiple portlets on the different portal page

com.webmethods.caf.faces.portleturl.ExtendedPortletUrlParameter ("Extended Portlet Parameter" on page 480) children controls are used to pass portlet parameters

Extended Portlet URL controls can be added as children to address multiple portlets on the page.

Name	Display Name	Туре	Description	Choices	Default	
General Properties						
id	ID	String	Control ID.			
rendered	Rendered	boolean	Not used because F rendered control	PortletUrl is not a	true	
Portlet Properties						
secure	Is Secure	boolean	Whether to use http generated portlet li		false	
portletMode	Portlet Mode	String	What is target portlet mode.		view	
type	Туре	String	Portlet link type	render, action	render	
windowState	Window State	String	What is desired portlet window state	normal,maximized, minimized	normal	
Extended Properties	S					
baseUrl	Base URL	String	Base value for this portlet URL. May also point to a portal page or individual portlet. When targetting the same portlet, may be left empty			
portlet	Portlet	String	Target portlet for this URL. When targetting the same portlet, may be left empty, otherwise should identify a portlet on the page by portlet ID or alias.			
targetAction	Target Action	String	Method binding extype URLs.	pression to invoke for "	'action"	

Name	Display Name	Туре	Description	Choices	Default	
keepState	Keep State	boolean	Forcefully keeps ' target portlet.	pageflow" state of the	false	
Legacy Portlet Properties						
portletLayout	Portlet Layout	String	Portlet layout to display for legacy webMethods portlets.			
portletMethod	Portlet Method	String	Portlet method to portlets.	invoke for legacy webN	lethods	

com.webmethods.caf.faces.portleturl.ExtendedPortletUrlParameter ("Extended Portlet Parameter" on page 480) to specify parameters for this portlet URL.

com.webmethods.caf.faces.portleturl.ExtendedPortletUrl to add other portlet URLs when multiple portlets need to be addressed.

Standard JSF Facets

None.

Extended CAF Facets

None.

Portlet Include

com. we bmethods. caf. faces. component. portlet. Portlet Include

MWS Control Component Library

This control dynamically embeds the content for an external portlet. The HTML content of the included portlet is injected right into the content of the parent portlet. Both webMethods 6.5 (legacy) and 7.x JSR 168 portlet content may be included using PortletInclude. This control can be used only for portlets deployed in webMethods containers; it is not supported for third-party containers.

ControlParameter controls added as children are used to pass preference values to the embedded portlet, where the name of the parameter is the name of the preference and the value of the parameter is the value of the preference set on the portlet to be included

When including content of another portlet keep in mind the limitation that if the parent portlet has a form element (most portlets do) and you are including another portlet inside that form which also has its own form, the included portlet may not function properly because HTML does not allow nested form elements.

Standard JSF Properties

Name	Display Name	Туре	Description	Default
General Properties	3			
id	ID	String	Control ID.	
rendered	Rendered	boolean	Whether to render this control	true
Portlet				
portlet	Portlet	String	Alias of the portlet to include (required)	
layout	Layout	String	Portlet layout to display for legacy WM portlet	

Children

ControlParameter to specify preferences of the included portlet

Standard JSF Facets

None.

Extended CAF Facets

None.

Portlet Parameter

com. we bmethods. caf. faces. portleturl. Portlet Url Parameter

CAF Portlet Controls Component Library

A parameter to be used for com.webmethods.caf.faces.portleturl.PortletUrl ("Portlet URL" on page 487) parent controls.

Name	Display Name	Туре	Description	Default
General Properties				
array	Is Array	boolean	Whether supplied value is a comma-separated array	false

Name	Display Name	Туре	Description	Default
name	Name	String	Name of the portlet prefere	ence to set
value	Value	String	Value binding expression of the value to be set on the portlet	

None.

Standard JSF Facets

None.

Extended CAF Facets

None.

Portlet Simple Link

com. we bmethods. caf. faces. component. portlet. Portlet Simple Link

CAF Portlet Controls Component Library

A link to the other pages and portlets. This control extends the standard "Link" on page 379 control by adding a facet for the "Portlet URL" on page 487 or "Extended Portlet URL" on page 481 controls which will be used as the link URL.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display o	control.	true
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instance to use to convert the control's value to and from a String.		e control's value
value	Value	Object	Link label.		

Name	Display Name	Туре	Description	Choices	Default
Expert Display Prop	perties				
accesskey	Access Key	String	Keyboard shortcu	t letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), R (right-to-left)	TL
lang	Language	String	Language code fo	r content.	
style	CSS Style	String	HTML "style" attr	ibute value.	
styleClass	CSS Class	String	HTML "class" attr	ibute value.	
tabindex	Tab Index	int		sequence, from 1 to n. Zo egative integer removes	
title	Element Title	String	HTML "title" attri tooltip).	bute value (often display	ved as a
Expert Client Side E	Events				
onblur	Blur	script	Control losing foc	us.	
onfocus	Focus	script	Control gaining for	ocus.	
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-cl	icked.	
onkeydown	Key Down	script	Key pressed wher	n control is focused.	
onkeyup	Key Up	script	Key released whe	n control is focused.	
onmousedown	Mouse Down	script	Mouse button pre	ssed when over control.	
onmousemove	Mouse Move	script	Mouse moved wh	en over control.	
onmouseout	Mouse Out	script	Mouse moved out	of control boundaries.	
onmouseover	Mouse Over	script	Mouse moved into	o control boundaries.	
onmouseup	Mouse Up	script	Mouse button rele	eased when over control	
Expert Properties					
charset	Charset	String	Character encodir	ng of linked resource.	
coords	Coordinates	String	Client-side image	map coordinates.	
hreflang	Href Language	String	Language of linke	d resource.	
rel	Relationship	String	Link relationship linked resource.	from the current resource	e to the
rev	Reverse Link	String	Link relationship current resource.	from the linked resource	e to the

Name	Display Name	Туре	Description	Choices	Default
shap	Shape	String	Client-side image map shape.	default, rect, circle, poly	
target	Target	String	Name of frame or window whose content will be replaced with the response from this form.	_blank (new window), _s (current frame), _parent (frame), _top (current win other	parent
type	Content type	String	Content type of link	ked resource.	
Expert Display Prop	perties				
accesskey	Access Key	String	Keyboard shortcut	letter.	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL (right-to-left)	
image	Image	int	Server-side path to button label.	an image to display in the	
lang	Language	String	Language code for	content.	
style	CSS Style	String	HTML "style" attrib	ute value.	
styleClass	CSS Class	String	HTML "class" attrib	ute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attribu	te value (often displayed as a	tooltip).

Any. Children are displayed as link label (after value of "label" property).

javax.faces.Parameter children are added as request parameters to the link.

com.webmethods.caf.faces.portleturl.PortletUrlScript("Portlet Url Script" on page 496) children may be used to hook into client-side javascript events (onClick)

Standard JSF Facets

None

Extended CAF Facets

PortletUrl - to place "Portlet URL" on page 487 or "Extended Portlet URL" on page 481 components there which will be used for the link URL.

Portlet URL

com.webmethods.caf.faces.portleturl.PortletUrl

CAF Portlet Controls Component Library

A JSR 168-compliant portlet URL that can be used in non-webMethods containers. This control allows you to build a portlet URL to the current portlet, change its view mode and window state, and pass parameters using com.webmethods.caf.faces.portleturl.PortletUrlParameter ("Portlet Parameter" on page 483) children controls.

Standard JSF Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	Not used because PortletUrl is not a rendered control		true
Portlet Properties					
secure	Is Secure	boolean	Whether to use https:// protocol for generated portlet links		false
portletMode	Portlet Mode	String	What is target portlet	mode.	view
type	Туре	String	Portlet link type	render, action	render
windowState	Window State	String	What is desired portlet window state	normal, maximized, minimized	normal

Children

com.webmethods.caf.faces.portleturl.PortletUrlParameter ("Portlet Parameter" on page 483) children may be used to pass parameters to the portlet

Standard JSF Facets

None.

Extended CAF Facets

Chapter 21. Script Controls

Control Reference	. 490
Custom Script	. 490
Invoke Script	. 491
Parameter	. 494
Portlet Url Script	. 496
Portlet Url Script Parameter	. 498
Return Value Script	. 500

Control Reference

com.webmethods.caf.faces.script.ControlReference

Base Controls Component Library

A control used to specify multiple controls for the "Invoke Script" on page 491 control to operate with. Control Reference is dropped as a child to Invoke Script. The value of the Control Reference should be set to the ID of the control it references

Standard JSF Properties

Name	Display Name	Туре	Description
Script			
value	Value	String	ID of the control it references

Children

None

Standard JSF Facets

None

Extended CAF Facets

None

Custom Script

com.webmethods.caf.faces.script.CustomScript

Base Controls Component Library

A control that connects client-side javaScript functions to parent control client-side events. To connect to the desired client-side event, the CustomScript control should be dropped as a child into a parent Input or Output control. Children com.webmethods.caf.faces.script.ScriptParameter ("Parameter" on page 494) controls can be used to define parameters to use inside custom script code blocks.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Optional control ID).	

Name	Display Name	Туре	Description	Choices	Default
rendered	Rendered	boolean	Whether this contro	ol is rendered	true
for	For	"Control ID Reference" on page 192		to an event for the specific of the script control).	ed control
Portlet Properties					
event	Event	String	What client event should trigger this script	onclick, onchange, ondblclick, onmousedown, onmouseup, onmousemove, onmouseover, onmouseout, onkeydown, onkeypress, onkeyup	onclick
code	Code	String	Custom javascript o	code	

com.webmethods.caf.faces.script.ScriptParameter ("Parameter" on page 494) children can be used to specify additional parameters used in the custom script code

Standard JSF Facets

None.

Extended CAF Facets

None

Invoke Script

com.webmethods.caf.faces.script.InvokeScript

Base Controls Component Library

A control that connects a standard client-side javascript control-model action to a parent control's (or specified "for" control's) client-side event. The following actions are available:

- raise—actuates the command control specified by the "control" property.
- setDisabled—enables or disables the input control specified by the "control" property, based on the value of the "value" property or "value" facet (true to disable, false to enable).

- **setValue**—sets the value of the control specified by the "control" property to the value of the "value" property or "value" facet.
- **setVisible**—shows or hides the control specified by the "control" property, based on the value of the "value" property or "value" facet (true to show, false to hide).
- toggle—shows the control specified by the "control" property if the control is hidden, hides the control specified by the "control" property if the control is visible.

Other script controls can be nested inside the "value" facet of this script control to provide a value for actions that require a value.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Optional control ID.		
rendered	Rendered	boolean	Whether this control is rendered.		true
for	For	"Control ID Referenc e" on page 192	Connect the script to an (defaults to parent of the	event for the specified co e script control).	ontrol
Script					
event	Event	String	What client event should trigger this script.	onclick, onchange, ondblclick, onmousedown, onmouseup, onmousemove, onmouseover, onmouseout, onkeydown, onkeypress, onkeyup	onclick
action	Action	String	Action to invoke on the control(s) model.	toogle, raise, setValue, setVisible, setDisabled	setValue

Name	Display Name	Туре	Description	Choices	Default
control	Control	String	Binding expression or ID of the control to invoke model action on. If the action is being invoked on multiple controls (for example to show/hide multiple controls on the page) Control Reference controls can be dropped as children to the InvokeScript component to define multiple controls.		
value	Value	String	Value to be passed to control model action. Alternatively value may be taken from another script or the ScriptParamter which are dropped in the "value" facet.		

com.webmethods.caf.faces.script.ControlReference ("Control Reference" on page 490) can be used to specify multiple controls to invoke client side model action.

Standard JSF Facets

Value: Used to specify a value for the action be returned from another script or defined as a ScriptParameter ("Parameter" on page 494). Drop the following controls into value facet:

- com.webmethods.caf.faces.script.ScriptParameter ("Parameter" on page 494)
- com.webmethods.caf.faces.script.CustomScript ("Custom Script" on page 490)
- com.webmethods.caf.faces.script.GetValueScript ("Return Value Script" on page 500)
- com.webmethods.caf.faces.script.PortletUrlScript ("Portlet Url Script" on page 496)

Extended CAF Facets

None

Parameter

com.webmethods.caf.faces.script.ScriptParameter

Base Controls Component Library

A control used to specify parameters for the "Custom Script" on page 490 control. The parameter defines a client-side named variable whose value is set depending on the configured valueType. It can be specified directly, taken from the client control, or returned back from another JavaScript function.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
name	Name	String	Name of the parameter to be used on the client-size portlet URL JavaScript block. Generally this name should match the corresponding portlet preference name		
portlet	Portlet	String	Target portlet identifier. This has to be set only if PortletUrl used for the script block references multiple portlets, otherwise may be left empty (default portlet will be used).		

Name	Display Name	Туре	Description	Choices	Default
valueType	Value Type	String	What is the value type:	function value: value property is left empty; instead a script is dropped in the Parameter value facet to implement a function that returns a value that will be assigned to the client side variable. string: specified value is a JavaScript string. number: specified value is a JavaScript number (either float or integer). boolean: specified value is a JavaScript boolean. array: specified value is a commaseparate array of strings and is defined as array variable on the client side. control reference: specified value is either a control ID or binding expression to the control. On the client side, the parameter is defined as the corresponding control client-side model. control value: specified value is either a control ID or a binding expression to the control. On the client side, the parameter is defined as the value of the corresponding control. function reference: the value property is left empty; instead a script is dropped in the Parameter value facet to implement a named function, which then may be invoked from customer script code. For example if "function reference" parameter name is foo, then child script may be invoked as foo().	string
value	Value	String	Data bindir	ng expression or string value of the param	eter

None.

Standard JSF Facets

 $Value.\ Place\ other\ Script\ in\ this\ facet\ when\ using\ "function\ value"\ or\ "function\ reference"\ value\ Type.$

Extended CAF Facets

None.

Portlet Url Script

com.webmethods.caf.faces.portleturl.PortletUrlScript

CAF Portlet Controls Component Library

A control that navigates to a portlet URL in response to a parent control's (or specified "for" control's) client-side event. Usually this script control is dropped inside other input controls (like Links or Buttons) and is invoked as response to corresponding onclick/onchange action. It can also be used to define a parameter for a com.webmethods.caf.faces.scripts.CustomScript ("Custom Script" on page 490) control. The "context" property controls the behavior of this script:

- link—navigates to a portlet URL as an HTML link.
- form—navigates to a portlet URL as a form submit (may be used to overcome the 2k URL length limitation). Note that the portlet URL must be an "action" type URL instead of the default "render" type.
- href—does not navigate to a portlet URL, but instead returns the portlet URL as a value that can be used from other script components.

com.webmethods.caf.faces.script.PortletUrlScriptParameter ("Parameter" on page 494) child controls can be used to specify parameters for the portlet URL.

Name	Display Name	Туре	Description	Choices	Default
General Properti	es				
id	ID	String	Optional control ID.		
rendered	Rendered	boolean	Whether this control is rendered true		
for	For	"Control ID Reference " on page 192	Connect the script to a to parent of the script of		ied control (defaults

Name	Display Name	Туре	Description	Choices	Default
Portlet Propertie	s				
context	Context	String	What is the context for the generated script block.	link:navigates to the target portlet URL as an HTML link. form: navigates to the target portlet URL as a form submit (may be used to overcome 2k URL length limitation) href: does not navigate to target a portlet URL, but instead returns the value of the portlet URL href. Use the href context to invoke a portlet URL script from other script components	link
event	Event	String	What client event should trigger this script.	onclick, onchange, ondblclick, onmousedown, onmouseup, onmousemove, onmouseover, onmouseout, onkeydown, onkeypress, onkeyup	onclick
formMethod	Form Method	String	Form method when using "form" context. When using "form" context portlet url to submit form at has to be an 'action' type url	GET, POST	POST
formTarget	Form Target	String	Form target when usin _blank, _parent, etc.	ng "form" context. For example: _s	self,

com.webmethods.caf.faces.script.PortletUrlScriptParameter ("Parameter" on page 494) children can be used to specify additional parameters to be set on the portlet URL.

Standard JSF Facets

None.

Extended CAF Facets

PortletUrl facet is used to drop-in com.webmethods.caf.faces.portleturl.PortletUrl ("Portlet URL" on page 487) or com.webmethods.caf.faces.portleturl.ExtendedPortletUrl ("Extended Portlet URL" on page 481) controls which are used by the script action.

Portlet Url Script Parameter

com.webmethods.caf.faces.portleturl.PortletUrlScriptParameter

CAF Portlet Controls Component Library

A control used to specify a portlet parameter for PortletUrlScript ("Portlet Url Script" on page 496). It extends the ScriptParameter ("Parameter" on page 494) control by adding a portlet property to optionally specify what portlet (for multi-portlet URLs) the parameter should be applied to. The parameter defines a client-side value for PortletUrlScript, depending on the valueType set, the value can be specified directly, be taken from the client control, or returned back from another JavaScript function

Name	Display Name	Туре	Description	Choices	Default
General Properties					
name	Name	String	Name of the parameter to be used on the client-size portlet URL JavaScript block. Generally this name should match the corresponding portlet preference name.		
portlet	Portlet	String	URL used for the scrip	r. This has to be set only if t block references multiple t empty (the default portle	e portlets.

Name	Display Name	Туре	Description	Choices	Default
value	Value	String	Data binding expression	on or string value of the paramete	er.
valueType	Value Type	String	What is the value type.	string: specified value is a JavaScript string. number: specified value is a JavaScript number (either float or integer). boolean: specified value is a JavaScript boolean. array: specified value should be a comma-separated array of strings, and is defined as an array variable on the client side. control value: specified value is either a control ID or a binding expression to the control. On the client side, the parameter is defined as the value of the corresponding control. function value: value property is left empty; instead another script is dropped in the Parameter value facet to implement a function which returns a value that will be assigned to the client side variable.	string

None.

Standard JSF Facets

Value. Place another Script in this facet when using the "function value" valueType; the parameter would be defined as the value returned from that child script.

Extended CAF Facets

Return Value Script

com.webmethods.caf.faces.script.GetValueScript

Base Controls Component Library

A control that implements a client-side JavaScript function to return values from the control client-side model. Usually this script is used to define value for:

- com.webmethods.caf.faces.script.ScriptParameter ("Parameter" on page 494)
- com.webmethods.caf.faces.script.InvokeScript ("Invoke Script" on page 491)

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Optional control ID.		
rendered	Rendered	boolean	Whether this control is rendered.		
for	For	"Control ID Reference" on page 192	Connect the script to an (defaults to parent of the	event for the specified co e script control).	ontrol
Script					
event	Event	String	What client event should trigger this script.	onclick, onchange, ondblclick, onmousedown, onmouseup, onmousemove, onmouseover, onmouseout, onkeydown, onkeypress, onkeyup	onclick

Name	Display Name	Туре	Description	Choices	Default
action	Action	String	Action to invoke on the control(s) model.	getValue, isVisible, isDisabled	getValue
control	Control	String	Binding expression or ID of the control to invoke model action on. If the action is being invoked on multiple controls (for example to show/hide multiple controls on the page) ControlReference may be dropped as children to "Invoke Script" on page 491 to define multiple controls.		

None

Standard JSF Facets

None

Extended CAF Facets

None

Chapter 22. Search Controls

Async Search Result Table	. 504
Async Search Result Tree	. 510
Search Bar	. 514
Search Result Table	. 517
Search Result Tree	. 521

Async Search Result Table

com.webmethods.caf.faces.search.AsyncSearchResultTable

CAF Portlet Controls Component Library

A common search result table control. Encapsulates the standard My webMethods Server search result table look-and-feel.

Although this control can be used in any generic JSF application, typically it is used in the Search Results portlet of a Search Bar / Search Result portlet pair. A Search Results portlet can be created via the New Portlet wizard, selecting the Search Results Portlet option on the first page of the wizard. The two portlets are connected via wiring: usually the lastSearchState property of the Search Bar portlet is wired to the queryString property of the Search Results portlet.

The Async Table differs from the standard Table in that the Async Table pages and sorts via asynchronous requests. It has a number of configurable properties (in the Lazy Load group) that can be used to customize its asynchronous behavior. It shares some of these properties in common with the standard "Hideable Controls" on page 194. See the com.webmethods.caf.faces.table.AsyncTable ("Async Table" on page 531).

If you generate a new Search Results portlet with Designer 7.1 or later, you do not have to add any preferences manually; preferences are generated automatically for you. To persist changes to column sizes with a Search Results portlet generated with Designer 7.0.x, it is necessary to add a new preference to the Search Results portlet. Follow these steps:

- 1 Open the /WebContent/WEB-INF/portlet.xml file.
- 2 Using the Configuration page of the Portlet Application Configuration Editor, find your Search Results portlet in the Portlet Application Editor tree, and select its Preferences node.
- 3 Click Add; in the Create Portlet Preference dialog, in the Name field, type columnWidths, and click Next.
- 4 In the Display Name field, type Column Widths.
- 5 In the Type list, choose String.
- 6 In the Scope list, choose Value stored per portlet instance.
- 7 Select the Hidden checkbox and click Finish.

If you use the variant of the Search Results that allows toggling between a table and tree, also add a "treeColumnWidths" preference with the same Type and Scope.

To provide custom content for the export (for example, to provide a set of columns with more machine-friendly formatted data), you can bind this control's "exportProvider" property value to a separate com.webmethods.caf.faces.data.ITableContentProvider instance. A convienence ITableContentProvider implementation designed for CSV export is the com.webmethods.caf.faces.data.export.CSVExportProvider class. It maps the

content of an existing ITableContentProvider (usually the provider which provides the raw data for the table) to a custom set of columns, with each column's content specified by a binding expression.

The following example demonstrates creating a CSVExportProvider in a custom page bean method. (You would bind the property for which this method is a getter, "myExportProvider", to this control's "exportProvider" property value.) It assumes that there is already an existing table content-provider for the raw table data, accessible via the getMyExistingTableContentProvider() method. It creates four columns: "ID", "Title", "Priority", and "Priority Name"; it specifies the content of these columns with binding expressions: "#{row.projectId}_#{row.id}", "#{row.title}", "#{row.priorityValue}", and "#{row.priorityName}" (the name of the row variable, in this case "row", must be specified as the third argument to the CSVExportProvider constructor.) For each row, the content of the "ID" column in the exported table will be generated from the value of the exsting-table row's "projectId" and "id" properties; the content of the "Title" column will be generated from the value of the existing-table row's "title" property; and so on.

```
public ITableContentProvider getMyExportProvider() {
    ITableContentProvider existingProvider = getMyExistingTableContentProvider();

Map keys = new LinkedHashMap();
    keys.put("ID", createValueBinding("#{row.projectId}_#{row.id}"));
    keys.put("Title", createValueBinding("#{row.title}"));
    keys.put("Priority", createValueBinding("#{row.priorityValue}"));
    keys.put("Priority Name", createValueBinding("#{row.priorityName}"));

return new CSVExportProvider(existingProvider, keys, "row");
}
```

Note that since the exported table's column labels are simply the column keys ("ID", "Title", "Priority", and "Priority Name"), it is a good practice to use string resources from a resource bundle instead of hardcoding the labels (the labels are hardcoded in this example for simplicity). Note also, a LinkedHashMap was used in the above example, which ensures that the columns will be exported in the order they were added to the keys map. The exported CSV might look like the following:

```
ID, Title, Priority, Priority Name
higgins_123, Nothing Works, 1, Critical
higgins_ABC, "Name should be ""MWS""", 3, Medium
```

The export processing handles the escaping of column values (in the example output above, the Name should be "MWS" value is escaped as "Name should be ""MWS""").

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of display.	first row in page to	0
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display contr	ol.	true
rows	Rows	int	Number of rows in p signals "all".	age to display. Less thar	n one
Value Properties					
value	Value	ITableContent Provider	Data bound to table.		
var	Row Variable	String	Name of current row row within column c	object (used to reference ontent).	e current
Export Properties					
exportButton Available	Render Export Table Button	boolean	True to show Export	button.	true
showEncoding Dialog	Show Encoding Dialog	boolean	True to show charact	er encoding dialog.	true
Lazy Load Propertie	S				
bufferChunk	Buffer Chunk	int	Chunk size in rows to client-side cache buff buffer via one big cho	C C	0
bufferMin	Buffer Minimum	int	buffer (hidden rows	vs of client-side cache buffered on either side age). Zero signals don't	0
bufferMax	Buffer Maximum	int	Maximum size in rows of client-side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.		0
clientSideCache	Client Side Cache	boolean	True to store rows in control or page is ref	client-side cache until reshed.	false

Name	Display Name	Туре	Description	Choices	Default
flashOnChange	Flash On Change	boolean	True to flash individiual rows that have changed when content finishes loading asynchronously.		true
progressDelay	Progress Delay	int	Milliseconds to wait progress dialog.	before showing	0
progressMsg	Progress Message	String	Message to display in progress dialog.		Defaults to localized message; in English, "Loading "
suppressInputs	Suppress Inputs	Collection	validation/update p	s for which to disable rocessing when handling est (comma separated).	; an
Expert Properties					
firstRequest Param	First Request Param	String	Name of request parameter that specifies table's first row. Used only for "restful" paging.		
rowsRequest Param	Rows Request Param	String	Name of request parameter that specifies number of rows for the table to display. Used only for "restful" paging.		
sortRequest Param	Sort Request Param	String	Name of request par Used only for "restfi	rameter that specifies tab	le's sort inf.
maxOrdinal	Max Ordinal	int		of columns at a time by w than one signals "unlimit	
Expert Display Prop	erties				
bgcolor	Background Color	Color	Table background co	olor.	
border	Border	int	Border width in pixels, for both outer table border and borders between cells.		order and
cellpadding	Cell Padding	String	Padding within cell	content between margin	and text.
cellspacing	Cell Spacing	String	Margin between cell border and cell content.		
columnClasses	Column CSS Classes	Collection	CSS class to use for each column, from left to right (not repeating).		right (not
customRow Class	Custom Row CSS Class	String	Binding expression name to add to the	that specifies a custom C current row.	SS class

Name	Display Name	Туре	Description	Choices	Default
dir	Directionalit y	String	Text direction hint.	LTR (left-to-right), RTL left)	(right-to-
dragAndDrop Columns	Drag and Drop Columns	boolean	True to allow drag-to drag-to-reorder-colu	o-resize-columns and mns functionality.	false
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-footer	
frame	Frame	String	Outer border style.	void (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)	void
headerClass	Header CSS Class	String	CSS class to use for table header.		
lang	Language	String	Language code for content.		
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating. portlet-section-body, portlet-section-body, portlet-section-alternate		ortlet-
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	void
style	CSS Style	String	HTML "style" attribu	ite value.	
styleClass	CSS Class	String	HTML "class" attribu	ıte value.	
summary	Summary	String	Summary text.		
title	Title	String	HTML "title" attribut	te value.	
width	Width	String	Width of table in pix ('100%').	els ('100') or percent	

Name	Display Name	Туре	Description	Choices	Default
Expert Value Proper	ties				
querySupplied	Query Supplied	boolean	True if query supplied (false to display Ready to Search message).	#{empty activePageBean.activeS yString}	SearchQuer
Expert Export Prope	erties				
exclude Components CSV	Exclude Component IDs	String	CSV string of control exclude from the exp	l IDs (of controls within toort.	he table) to
exportCharacter Encoding	Export Encoding	String	Default character encoding.	UnicodeLittle	
exportFileName	Export File Name	String	Name to use for exported file.	export.csv	
exportProvider	Export Provider	ITableContent Provider	The table content to be used for export.		
Expert Client Side E	vents				
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-click	ed.	
onkeydown	Key Down	script	Key pressed when co	ontrol is focused.	
onkeypress	Key Press	script	Key pressed and rele	eased when control is foo	cused.
onkeyup	Key Up	script	Key released when o	control is focused.	
onmousedown	Mouse Down	script	Mouse button pressed when over control.		
onmousemove	Mouse Move	script	Mouse moved when over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundries.		
onmouseover	Mouse Over	script	Mouse moved into control boundries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		

javax.faces.Column ("Basic Column" on page 547)

Facets

None.

Async Search Result Tree

com. we bmethods. caf. faces. search. A sync Search Result Tree

CAF Portlet Controls Component Library

A common search result tree control. Encapsulates the standard My webMethods Server search result tree look-and-feel.

Although this control can be used in any generic JSF application, typically it is used in the Search Results portlet of a Search Bar / Search Result portlet pair. A Search Results portlet can be created via the New Portlet wizard, selecting the Search Results Portlet option on the first page of the wizard. The two portlets are connected via wiring: usually the lastSearchState property of the Search Bar portlet is wired to the queryString property of the Search Results portlet.

The Async Tree has a number of configurable properties (in the Lazy Load group) which can be used to customize its asynchronous behavior. It shares some of these properties in common with the standard "Hideable Controls" on page 194. See the com.webmethods.caf.faces.tree.AsyncTree ("Async Tree" on page 536).

If you generate a new Search Results portlet with Designer 7.1 or later, you do not have to add any preferences manually; preferences are generated automatically for you. To persist changes to column sizes with a Search Results portlet generated with Designer 7.0.x, it is necessary to add a new preference to the Search Results portlet. Follow these steps:

- 1 Open the /WebContent/WEB-INF/portlet.xml file.
- 2 Using the Configuration page of the Portlet Application Configuration Editor, find your Search Results portlet in the Portlet Application Editor tree, and select its Preferences node.
- 3 Click Add; in the Create Portlet Preference dialog, in the Name field, type columnWidths, and click Next.
- 4 In the Display Name field, type Column Widths.
- 5 In the Type list, choose String.
- 6 In the Scope list, choose Value stored per portlet instance.
- 7 Select the Hidden checkbox and click Finish.

If you use the variant of the Search Results that allows toggling between a table and tree, also add a "treeColumnWidths" preference with the same Type and Scope. If you generate a new Search Results portlet, you do not have to add any preferences manually; they will be generated automatically for you.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cor	ntrol.	true
Value Properties					
value	Value	ITableContent Provider	Data bound to table.		
var	Row Variable	String	Name of current ro row within column	ow object (used to referer n content).	nce current
Lazy Load Propertie	es				
initialDepth	Initial Depth	int	Levels of nodes to pull down to the client initially.	0 roots only, 1 roots + children, 2 = roots + children + grandchildren, etc.	0
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog (re-sorting only).		0
progressFlash OnComplete	Flash On Load	boolean	True to flash individiual rows when they finish loading asynchronously.		true
progressMsg	Progress Message	String	Message to display in progress dialog (re-sorting only).		Defaults to localized message; in English, "Loading "
refillDepth	Refill Depth	int	Levels of nodes to pull down to the client when user expands a node whose children aren't on the client.	0 children only, 1 children + grandchildren, 2 = children + grandchildren + great-grandchildren, etc.	0
refreshOnShow	Refresh On Show	boolean	is toggled visible; f	tent every time control false to refresh content itially hidden and then	false

Name	Display Name	Туре	Description	Choices	Default	
suppressInputs	Suppress Inputs	Collection	validation/update	rols for which to disable processing when handlinuest (comma separated).	ng an	
Expert Properties						
sortRequest Param	Sort Request Param	String	Name of request prinf. Used only for	parameter that specifies to "restful" paging.	able's sort	
maxOrdinal	Max Ordinal	int		er of columns at a time by d. Less than one signals "		
Expert Display Prop	erties					
bgcolor	Background Color	Color	Table background	color.		
border	Border	int		Border width in pixels, for both outer table border and borders between cells.		
cellpadding	Cell Padding	String	Padding within ce	ell content between marg	in and text.	
cellspacing	Cell Spacing	String	Margin between c	ell border and cell conter	nt.	
columnClasses	Column CSS Classes	Collection	CSS class to use for repeating).	CSS class to use for each column, from left to right (not repeating).		
customRow Class	Custom Row CSS Class	String	Binding expression name to add to the	n that specifies a custom e current row.	CSS class	
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RT left)	L (right-to-	
dragAndDrop Columns	Drag and Drop Columns	boolean		g-to-resize-columns and olumns functionality.	false	
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-footer		
frame	Frame	String	Outer border style.	void (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)	void	

Name	Display Name	Туре	Description	Choices	Default
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-subheac	ler
lang	Language	String	Language code for	content.	
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	portlet-section-body, po section-alternate	ortlet-
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	void
style	CSS Style	String	HTML "style" attri	bute value.	
styleClass	CSS Class	String	HTML "class" attri	bute value.	
summary	Summary	String	Summary text.		
title	Title	String	HTML "title" attrib	ute value.	
width	Width	String	Width of table in p	ixels ('100') or percent ('1	00%').
Expert Value Proper	ties				
querySupplied	Query Supplied	boolean	True if query supplied (false to display Ready to Search message).	#{empty activePageBean.activeS yString}	earchQuer
Expert Client Side E	vents				
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clie	cked.	
onkeydown	Key Down	script	Key pressed when	control is focused.	
onkeypress	Key Press	script	Key pressed and released when control is focused.		
onkeyup	Key Up	script	Key released when	control is focused.	
onmousedown	Mouse Down	script	Mouse button pres	sed when over control.	

Name	Display Name	Туре	Description	Choices	Default
onmousemove	Mouse Move	script	Mouse moved when over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundries.		
onmouseover	Mouse Over	script	Mouse moved into control boundries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		

javax.faces.Column ("Basic Column" on page 547)

Facets

None.

Search Bar

com.webmethods.caf.faces.search.SearchBar

CAF Portlet Controls Component Library

A common search bar control. Displays standard search Keywords, Advanced, Saved, and Options tabs.

Although this control can be used in any generic JSF application, typically it is used in the Search Bar portlet of a Search Bar / Search Results portlet pair. A Search Bar portlet can be created via the New Portlet wizard, selecting the Search Bar Portlet option on the first page of the wizard. The two portlets are connected via wiring: usually the lastSearchState property of the Search Bar portlet is wired to the queryString property of the Search Results portlet.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to include control in renderered page.	true

Name	Display Name	Туре	Description	Default
Display Properties				
simpleSearch Available	Simple Search Available	boolean	True to display Keyword search tab.	true
advancedSearch Available	Advanced Search Available	boolean	True to display Advanced search tab.	true
saveSearch Available	Save Search Available	boolean	True to display Saved search tab.	true
userSearch Options Available	User Search Options Available	boolean	True to display Options search tab.	true
refineSearch ButtonAvailable	Refine Search Button Available	boolean	True to display Refine search button.	true
useOrCondition Available	Match Any/All Available	boolean	True to display Match Any/All dropdown.	true
maxResultsField Available	Max Results Field Available	boolean	True to display Max Results input field.	true
keywordsField Available	Keywords Field Available	boolean	True to display Keywords input field.	true
clearForm ButtonAvailable	Clear Form Available	boolean	True to display Clear button.	true
Action Properties				
useAsyncGo Button	Async 'Go' Button Enabled	boolean	True to make Go button an Asynchronous Command Button, false to make Go button a basic Command Button.	false
asyncGoRefresh	Async 'Go' Refresh	Control ID Reference	ID of control to refresh when user button.	clicks Go
Expert Properties				
searchQuery Factory	Search Query Factory	ISearchQuery Factory	Custom search builder class.	
searchQuery Factory	Saved Search Provider	ISavedSearch Provider	Custom search options provider cl	ass.
searchQuery Factory	Search Options Manager	ISearchOptionsM anager	Custom search options manager cl	ass.

Name	Display Name	Туре	Description De	fault
Expert Action Proper	rties			
buildQuery StringAction	Go Action	MethodBinding	Method to invoke when user clicks Go	button.
clearForm Action	Clear Form Action	MethodBinding	Method to invoke when user clicks Clebutton.	ear
saveQuery Action	Save Query Action	MethodBinding	Method to invoke when user clicks Sav (search) button.	⁄e
changeActive SavedSearch Action	Change Active Saved Search Action	MethodBinding	Method to invoke when user changes to Saved Search dropdown.	the
deleteSaved Action	Delete Saved Action	MethodBinding	Method to invoke when user clicks De (saved search) button.	lete
updateSaved QueryAction	Update Saved Query Action	MethodBinding	Method to invoke when user clicks Up (saved search) button.	date
persistSearch OptionsAction	Persist Options Action	MethodBinding	Method to invoke when user clicks Sav (options) button.	/e
cancelSearch OptionsAction	Cancel Options Action	MethodBinding	Method to invoke when user clicks Car (options) button.	ncel
Expert Value Propert	ies			
maxResultsMax Value	Maximum 'Max Results' Value	int	Maximum allowed value for 20 'Max Results' option.	00

None.

Facets

Name	Display Name	Description	Allowed Types
refinePanel	Refine Panel	Custom refine/advanced UI.	Any

Search Result Table

com.webmethods.caf.faces.search.SearchResultTable

CAF Portlet Controls Component Library

A common search result table control. Encapsulates the standard My webMethods Server search result table look-and-feel.

Although this control can be used in any generic JSF application, typically it is used in the Search Results portlet of a Search Bar / Search Result portlet pair. A Search Results portlet can be created via the New Portlet wizard, selecting the Search Results Portlet option on the first page of the wizard. The two portlets are connected via wiring: usually the lastSearchState property of the Search Bar portlet is wired to the queryString property of the Search Results portlet.

If you generate a new Search Results portlet with Designer 7.1 or later, you do not have to add any preferences manually; preferences are generated automatically for you. To persist changes to column sizes with a Search Results portlet generated with Designer 7.0.x, it is necessary to add a new preference to the Search Results portlet. Follow these steps:

- 1 Open the /WebContent/WEB-INF/portlet.xml file.
- 2 Using the Configuration page of the Portlet Application Configuration Editor, find your Search Results portlet in the Portlet Application Editor tree, and select its Preferences node.
- 3 Click Add; in the Create Portlet Preference dialog, in the Name field, type columnWidths, and click Next.
- 4 In the Display Name field, type Column Widths.
- 5 In the Type list, choose String.
- 6 In the Scope list, choose Value stored per portlet instance.
- 7 Select the Hidden checkbox and click Finish.

If you use the variant of the Search Results that allows toggling between a table and tree, also add a "treeColumnWidths" preference with the same Type and Scope.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of first row in page to display.		0
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display contr	ol.	true

Name	Display Name	Туре	Description	Choices	Default
rows	Rows	int	Number of rows in pasignals "all".	age to display. Less th	nan one
Value Properties					
value	Value	ITableContent Provider	Data bound to table.		
var	Row Variable	String	Name of current row object (used to reference current row within column content).		
Export Properties					
exportButton Available	Render Export Table Button	boolean	True to show Export	button.	true
showEncoding Dialog	Show Encoding Dialog	boolean	True to show characte	er encoding dialog.	true
Expert Properties					
firstRequest Param	First Request Param	String	Name of request parameter that specifies table's first row. Used only for "restful" paging.		
rowsRequest Param	Rows Request Param	String	Name of request parameter that specifies number of rows for the table to display. Used only for "restful" paging.		
sortRequest Param	Sort Request Param	String	Name of request para inf. Used only for "re		able's sort
maxOrdinal	Max Ordinal	int	Maximum number of table can be sorted. L		
Expert Display Prop	erties				
bgcolor	Background Color	Color	Table background col	or.	
border	Border	int	Border width in pixel and borders between		e border
cellpadding	Cell Padding	String	Padding within cell c	ontent between marg	in and text.
cellspacing	Cell Spacing	String	Margin between cell	border and cell conte	nt.
columnClasses	Column CSS Classes	Collection	CSS class to use for each column, from left to right (not repeating).		
customRow Class	Custom Row CSS Class	String	Binding expression the name to add to the cu		CSS class

Name	Display Name	Туре	Description	Choices Default		
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTL (right-to-left)		
dragAndDrop Columns	Drag and Drop Columns	boolean	True to allow drag-to and drag-to-reorder-functionality.			
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-footer		
frame	Frame	String	Outer border style.	void (no sides), void above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)		
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-subheader		
lang	Language	String	Language code for co	ontent.		
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	portlet-section-body, portlet- section-alternate		
rules	Rules	String	Grid lines.	none, groups void (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)		
style	CSS Style	String	HTML "style" attribu	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attribu	HTML "class" attribute value.		
summary	Summary	String	Summary text.			

Name	Display Name	Туре	Description	Choices	Default	
title	Title	String	HTML "title" attribu	te value.		
width	Width	String	Width of table in pix	Width of table in pixels ('100') or percent ('100%').		
Expert Value Proper	ties					
querySupplied	Query Supplied	boolean	True if query supplied (false to display Ready to Search message).	#{empty activePageBean ueryString}	.activeSearchQ	
Expert Export Prope	rties					
exclude Components CSV	Exclude Component IDs	String	CSV string of contro table) to exclude from	•	vithin the	
exportCharacter Encoding	Export Encoding	String	Default character encoding.	UnicodeLittle		
exportFileName	Export File Name	String	Name to use for exported file.	export.csv		
Expert Client Side E	vents					
onclick	Click	script	Control clicked.			
ondblclick	Double Click	script	Control double-click	ed.		
onkeydown	Key Down	script	Key pressed when co	ontrol is focused.		
onkeypress	Key Press	script	Key pressed and rele	eased when contro	ol is focused.	
onkeyup	Key Up	script	Key released when o	control is focused.		
onmousedown	Mouse Down	script	Mouse button presse	ed when over cont	rol.	
onmousemove	Mouse Move	script	Mouse moved when	over control.		
onmouseout	Mouse Out	script	Mouse moved out of	Mouse moved out of control boundries.		
onmouseover	Mouse Over	script	Mouse moved into c	Mouse moved into control boundries.		
onmouseup	Mouse Up	script	Mouse button releas	ed when over con	trol.	

javax.faces.Column ("Basic Column" on page 547)

Facets

None.

Search Result Tree

com.webmethods.caf.faces.search.SearchResultTree

A common search result tree control. Encapsulates the standard My webMethods Server search result tree look-and-feel.

Although this control can be used in any generic JSF application, typically it is used in the Search Results portlet of a Search Bar / Search Result portlet pair. A Search Results portlet can be created via the New Portlet wizard, selecting the Search Results Portlet option on the first page of the wizard. The two portlets are connected via wiring: usually the lastSearchState property of the Search Bar portlet is wired to the queryString property of the Search Results portlet.

If you generate a new Search Results portlet with Designer 7.1 or later, you do not have to add any preferences manually; preferences are generated automatically for you. To persist changes to column sizes with a Search Results portlet generated with Designer 7.0.x, it is necessary to add a new preference to the Search Results portlet. Follow these steps:

- 1 Open the /WebContent/WEB-INF/portlet.xml file.
- Using the Configuration page of the Portlet Application Configuration Editor, find your Search Results portlet in the Portlet Application Editor tree, and select its Preferences node.
- 3 Click Add; in the Create Portlet Preference dialog, in the Name field, type columnWidths, and click Next.
- 4 In the Display Name field, type Column Widths.
- 5 In the Type list, choose String.
- 6 In the Scope list, choose Value stored per portlet instance.
- 7 Select the Hidden checkbox and click Finish.

If you use the variant of the Search Results that allows toggling between a table and tree, also add a "treeColumnWidths" preference with the same Type and Scope.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display co	ntrol.	true

Name	Display Name	Туре	Description Choices Default		
Value Properties					
value	Value	ITableContent Provider	Data bound to table.		
var	Row Variable	String	Name of current row object (used to reference current row within column content).		
Expert Properties					
sortRequest Param	Sort Request Param	String	Name of request parameter that specifies table's sort inf. Used only for "restful" paging.		
maxOrdinal	Max Ordinal	int	Maximum number of columns at a time by which table can be sorted. Less than one signals "unlimited".		
Expert Display Prop	perties				
bgcolor	Background Color	Color	Table background color.		
border	Border	int	Border width in pixels, for both outer table border and borders between cells.		
cellpadding	Cell Padding	String	Padding within cell content between margin and text.		
cellspacing	Cell Spacing	String	Margin between cell border and cell content.		
columnClasses	Column CSS Classes	Collection	CSS class to use for each column, from left to right (not repeating).		
customRow Class	Custom Row CSS Class	String	Binding expression that specifies a custom CSS class name to add to the current row.		
dir	Directionality	String	Text direction hint. LTR (left-to-right), RTL (right-to-left)		
dragAndDrop Columns	Drag and Drop Columns	boolean	True to allow drag-to-resize-columns and false drag-to-reorder-columns functionality.		
footerClass	Footer CSS Class	String	CSS class to use for portlet-section-footer table footer.		

Name	Display Name	Туре	Description	Choices	Default
frame	Frame	String	Outer border style.	void (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)	void
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-subl	neader
lang	Language	String	Language code for co	ntent.	
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	portlet-section-bod section-alternate	y, portlet-
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	void
style	CSS Style	String	HTML "style" attribut	e value.	
styleClass	CSS Class	String	HTML "class" attribut	e value.	
summary	Summary	String	Summary text.		
title	Title	String	HTML "title" attribute	e value.	
width	Width	String	Width of table in pixe	ls ('100') or percent ('10	00%').

Name	Display Name	Туре	Description	Choices	Default	
Expert Value Prope	rties					
querySupplied	Query Supplied	boolean	True if query supplied (false to display Ready to Search message).	#{empty activePageBean.a QueryString}	activeSearch	
Expert Client Side E	Events					
onclick	Click	script	Control clicked.			
ondblclick	Double Click	script	Control double-clicke	Control double-clicked.		
onkeydown	Key Down	script	Key pressed when co	Key pressed when control is focused.		
onkeypress	Key Press	script	Key pressed and relea	ased when control is	focused.	
onkeyup	Key Up	script	Key released when co	ontrol is focused.		
onmousedown	Mouse Down	script	Mouse button pressec	d when over control		
onmousemove	Mouse Move	script	Mouse moved when	over control.		
onmouseout	Mouse Out	script	Mouse moved out of control boundries.			
onmouseover	Mouse Over	script	Mouse moved into control boundries.			
onmouseup	Mouse Up	script	Mouse button release	d when over contro	l.	

javax.faces.Column ("Basic Column" on page 547)

Facets

None.

Chapter 23. Table Controls

Async Categorized Table	526
Async Table	531
Async Tree	536
Atom Feed Icon	541
Basic Column	547
Calendar	549
Categorized Table	553
Column Sort Link	557
Data Pages	558
Data Prev/Next	559
Data Total	560
Data Total Selected	562
Export Table Button	562
Hidden Column	565
Row Headers Column	566
Scrollbar Column	568
Standard Column	568
Table	570
Tree	575
Tree Toggle	580
Truncating Column	582

Async Categorized Table

com. we bmethods. caf. faces. table. A sync Categorized Table

Base Controls Component Library

A table whose content comes from either a javax.faces.model.DataModel object, or a com.webmethods.caf.faces.data.ITableContentProvider object. Standard list-like containers, such as java.util.List, are wrapped autonomatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset

Each item in the list is mapped to a row in the table. Columns are configured by adding column children. Wiodmth a categorized table, the column with the primary sort (a sort ordinal of 1) is rendered as the category for the group of rows which have the same value for that column. All other columns are rendered once for each row.

To demonstrate how the Categorized Table differs from a regular Table, consider the following regular Table, which is sorted first by Product, then by Quanity:

	Quarterly Sales	i
Product	Quantity	Customer
Hammer	300	General Mills
Hammer	2	General Electric
Saw	12	General Dynamics
Saw	10	General Motors
Screwdriver	70	General Dynamics
Screwdriver	1	General Electric

If we formatted the same data as a Categorized Table, it would look like the following. Rows with the same value for the Product column are displayed as a category of rows, grouped by the Product column value:

Quarterly Sales		
Product	Quantity	Customer
Hammer		
	300	General Mills
	2	General Electric
Saw		
	12	General Dynamics
	10	General Motors
Screwdriver		
	70	General Dynamics
	1	General Electric

The Async Table differs from the standard Table in that the Async Table pages and sorts via asynchronous requests. It has a number of configurable properties in the Lazy Load group that can be used to customize its asynchronous behavior. It shares some of these properties in common with the standard "Hideable Controls" on page 194. See the standard "Async Table" on page 531 for more information.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of first row in page to 0 display.		
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control. true		
rows	Rows	int	Number of rows in page to display. Less than one signals "all".		
Value Properties					
value	Data bound to table.	ITableContent Provider	Number of rows in page to display. Less than one signals "all".		
var	Row Variable	String	Name of current row object (used to reference current row within column content).		

Name	Display Name	Туре	Description Choices	Default	
Lazy Load Propertie	es				
bufferChunk	Buffer Chunk	int	Chunk size in rows to use when filling the client-side cache buffer. Zero signals fill buffer via one big chunk.	0	
bufferMin	Buffer Minimum	int	Minimum size in rows of client-side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.	0	
bufferMax	Buffer Maximum	int	Maximum size in rows of client-side cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.	0	
clientSideCache	Client Side Cache	boolean	True to store rows in client-side cache until control or page is refreshed.	false	
flashOnChange	Flash On Change	boolean	True to flash individiual rows that have changed when content finishes loading asynchronously.	true	
progressDelay	Progress Delay	int	Milliseconds to wait before showing progress dialog.	0	
progressMsg	Progress Message	String	Message to display in progress dialog.	Defaults to localized message; in English, "Loading"	
suppressInputs	Suppress Inputs	Collection	List of ids of controls for which to disable validation/update processing when hand asynchronous request (comma separated	dling an	
Expert Properties					
filter	Filter	String	A filter expression. Used to filter tables be filterable data models.	oound to	
maxOrdinal	Max Ordinal	int	Maximum number of columns at a time by which table can be sorted. Less than one signals "unlimited".		
Expert Display Prop	perties				
bgcolor	Background Color	Color	Table background color.		
border	Border	int	Border width in pixels, for both outer table border and borders between cells.		

Name	Display Name	Туре	Description	Choices	Default
cellpadding	Cell Padding	String	Padding within co	ell content between ma	argin and text.
cellspacing	Cell Spacing	String	Margin between	cell border and cell cor	ntent.
columnClasses	Column CSS Classes	Collection	CSS class to use for repeating).	or each column, from l	eft to right (not
customRow Class	Custom Row CSS Class	String	Binding expression name to add to the	on that specifies a custone current row.	om CSS class
dir	Directionality	String	Text direction hint.	LTR (left-to-right), F left)	TL (right-to-
fixedHeight	Fixed Height	boolean		hould have a fixed the number of rows	false
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-foote	r
frame	Frame	String	Outer border style.	void (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)	void
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-subh	eader
lang	Language	String	Language code fo	or content.	
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	portlet-section-body alternate	, portlet-section-
rowHeight	Row Height	String	Height of each ro (for example 20er	w in pixels (for examp n).	le 20px) or ems

Name	Display Name	Туре	Description	Choices	Default
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	void
style	CSS Style	String	HTML "style" attr	ibute value.	
styleClass	CSS Class	String	HTML "class" attr	ibute value.	
summary	Summary	String	Summary text.		
title	Title	String	HTML "title" attri	bute value.	
width	Width	String	Width of table in	pixels ('100') or percent	('100%').
Expert Client Side I	Events				
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-cl	icked.	
onkeydown	Key Down	script	Key pressed when	n control is focused.	
onkeypress	Key Press	script	Key pressed and	released when control i	s focused.
onkeyup	Key Up	script	Key released whe	n control is focused.	
	Mouse Down	script	Mouse button pre	essed when over contro	1.
onmousemove	Mouse Move	script	Mouse moved wh	nen over control.	
onmouseout	Mouse Out	script	Mouse moved ou	t of control boundaries	•
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button rele	eased when over contro	ol.

javax.faces.Column ("Basic Column" on page 547)

Facets

Name	Display Name	Description	Allowed Types
header	Header	Table header.	Any
footer	Footer	Table footer.	Any
empty	Empty Message	Message displayed in place of content when table is empty.	Any

Async Table

com.webmethods.caf.faces.table.AsyncTable

Base Controls Component Library

A table whose content comes from either a javax.faces.model.DataModel object, or a com.webmethods.caf.faces.data.ITableContentProvider object. Standard list-like containers, such as java.util.List, are wrapped autonomatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset

Each item in the list is mapped to a row in the table. Columns are configured by adding column children. Each column is rendered once for each row in the table.

The Async Table differs from the standard "Table" on page 570 in that the Async Table pages and sorts via asynchronous requests. It has a number of configurable properties in the Lazy Load group that can be used to customize its asynchronous behavior. It shares some of these properties in common with the standard "Hideable Controls" on page 194.

The "Lazy Load" properties unique to the Async Table concern client-side row caching, which is off by default. The "clientSideCache" property toggles this behavior. When true, the table caches rows that it has retrieved from the server until the page is refreshed, or the table (or a parent of the table) is refreshed. Like other controls, the table can be refreshed via client-side script using the refresh() method of its client-side model. The RefreshInterval control can also be used to refresh the table on a timed basis.

The Async Table can also be configured to buffer in its cached rows on either side of the visible page. When the "bufferMin" and "bufferMax" properties are set to positive integers, the table will make sure that there are at least "bufferMin" number of rows are

cached on either side of the visibible page; if not, it will retrieve "bufferMax" number of rows on either side of the visible page to store in the cache. If the "bufferChunk" property is set to zero (0), it will retrieve those rows in one big lump; if it's set to a positive integer, it will use multiple asynchronous requests to retrieve only "bufferChunk" number of rows at a time.

For example, say we have an Async Table with the following property settings

clientSideCache=true bufferMin=10 bufferMax=100 bufferChunk=0

and a user views a page with this Async Table displaying rows 1 - 10 of 1000. When the page is rendered initially, the Async Table will display rows 1 - 10, and buffer rows 11 - 110. If the user pages forward to rows 91 - 100, these rows already will be in the client-side cache, and can be displayed immediately, without an additional request. If the user pages foward to rows 101 - 110, these rows will also be in the client-side cache, and can be displayed immediately; however, the buffer will contain less than 10 rows on the "next" side of the visible page (in fact it will contain zero), so the Async Table will request rows 111 - 210. If the user jumps to rows 991 - 1000, these rows will not be in the client-side cache — and neither will be the previous 10 rows — so the Async Table will request rows 891 - 1000, displaying the last 10 rows in the table, and buffering 100 rows previous to it.

As another example, say we have an Async Table with the following property settings

clientSideCache=true bufferMin=50 bufferMax=50 bufferChunk=10

and a user views a page with this Async Table displaying rows 1 - 10 of 1000. When the page is rendered initially, the Async Table will display rows 1 - 10; as it waits for the user to do something it will make 5 asynchronous requests, each for 10 more rows, buffering 50 rows on the "next" side of the cache. If the user pages forward to rows 11 - 20, these rows will be in the client-side cache, and can be displayed immediately, without an additional request. However, the buffer will contain less than 50 rows on the "next" side of the visible page (it will contain 40 "next" rows), so the Async Table will request rows 61 - 70. If the user jumps to rows 991 - 1000, these rows will not be in the client-side cache, so the Async Table will request rows 991 - 1000, displaying them. It will then make 5 more asynchronous requests, each retrieving 10 more previous rows, bringing the buffer up to 50 on the "previous" side of the cache.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of to display.	f first row in page	0
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	trol.	true
rows	Rows	int	Number of rows in signals "all".	page to display. Les	s than one
Value Properties					
value	Data bound to table.	ITableContent Provider	Number of rows in signals "all".	page to display. Les	s than one
var	Row Variable	String	Name of current ro row within column	w object (used to refe content).	erence current
Lazy Load Propertie	S				
bufferChunk	Buffer Chunk	int	Chunk size in rows the client-side cach signals fill buffer vi	e buffer. Zero	0
bufferMin	Buffer Minimum	int	Minimum size in ro cache buffer (hidde either side of the vi Zero signals don't l	n rows buffered on sible table page).	0
bufferMax	Buffer Maximum	int	Maximum size in rows of client-side 0 cache buffer (hidden rows buffered on either side of the visible table page). Zero signals don't buffer.		0
clientSideCache	Client Side Cache	boolean	True to store rows in until control or page		false
flashOnChange	Flash On Change	boolean	True to flash individiual rows that true have changed when content finishes loading asynchronously.		
progressDelay	Progress Delay	int	Milliseconds to war progress dialog.	it before showing	0

Name	Display Name	Туре	Description	Choices	Default
progressMsg	Progress Message	String	Message to display	in progress dialog.	Defaults to localized message; in English, "Loading"
suppressInputs	Suppress Inputs	Collection	validation/update p	ols for which to disab processing when han est (comma separate	dling an
Expert Properties					
filter	Filter	String	A filter expression. filterable data mod	Used to filter tables els.	bound to
maxOrdinal	Max Ordinal	int		of columns at a time Less than one signal	
Expert Display Prope	erties				
bgcolor	Background Color	Color	Table background o	color.	
border	Border	int	Border width in pix and borders betwee	kels, for both outer ta en cells.	ıble border
cellpadding	Cell Padding	String	Padding within cell	l content between ma	argin and text.
cellspacing	Cell Spacing	String	Margin between ce	ll border and cell cor	ntent.
columnClasses	Column CSS Classes	Collection	CSS class to use for (not repeating).	each column, from l	eft to right
customRowClass	Custom Row CSS Class	String	Binding expression name to add to the	that specifies a custo current row.	om CSS class
dir	Directionality	String	Text direction LTR (left-to-right), RTL (right-to-hint. left)		
fixedHeight	Fixed Height	boolean	True if the table should have a fixed false height, based on the number of rows per page.		false
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-foote	er

Name	Display Name	Туре	Description	Choices	Default	
frame	Frame	String	Outer border style.	void (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)		
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-sub	bheader	
lang	Language	String	Language code for content.			
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	portlet-section-boo section-alternate	dy, portlet-	
rowHeight	Row Height	String	Height of each ro ems (for example	w in pixels (for exam 20em).	ple 20px) or	
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	void	
style	CSS Style	String	HTML "style" attr	ibute value.		
styleClass	CSS Class	String	HTML "class" attr	ibute value.		
summary	Summary	String	Summary text.			
title	Title	String	HTML "title" attri	HTML "title" attribute value.		
width	Width	String	Width of table in	pixels ('100') or perce	ent ('100%').	

Name	Display Name	Туре	Description	Choices	Default
Expert Client Side E	vents				
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-cli	icked.	
onkeydown	Key Down	script	Key pressed when control is focused.		
onkeypress	Key Press	script	Key pressed and released when control is focused.		
onkeyup	Key Up	script	Key released when control is focused.		
onmousedown	Mouse Down	script	Mouse button pre	ssed when over co	ontrol.
onmousemove	Mouse Move	script	Mouse moved wh	en over control.	
onmouseout	Mouse Out	script	Mouse moved out	of control bounda	aries.
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button released when over control.		

javax.faces.Column ("Basic Column" on page 547)

Facets

Name	Display Name	Description	Allowed Types
header	Header	Table header.	Any
footer	Footer	Table footer.	Any
empty	Empty Message	Message displayed in place of content when table is empty.	Any

Async Tree

com.webmethods.caf.faces.tree.AsyncTree

Base Controls Component Library

Tree whose content comes from a com.webmethods.caf.faces.data.tree.ITreeContentProvider object. An ITreeContentProvider presents a flattened view of the tree, iterating over the tree structure in depth-first order. The Async Tree differs from the standard Tree in that the Async Tree initially displays only a small portion of the tree, and uses asynchronous requests to display other portions as a user expands nodes on the client.

The two default ITreeContentProvider implementations described for the standard Tree control will also work fine with the Async Tree. A third default ITreeContentProvider implementation,

com.webmethods.caf.faces.data.tree.object.LazyNodeTreeContentProvider, will work best with large trees.

LazyNodeTreeContentProvider

The LazyNodeTreeContentProvider implementation is designed to access a tree's nodes only as needed, so it is suitable for use with a data source that would normally require many queries to construct the full tree. To use it, you must expose your data as a tree of com.webmethods.caf.faces.data.tree.INode objects. Your INode implementation needn't load children of a given node until the node's getChildren() method is called. This will happen only when a user expands the node in the ui; so for any large tree, only a relatively small number of nodes must load their children.

The LazyNodeTreeContentProvider can be initialized with either a list of root nodes or a single root node. Its initial open state can be configured to a certain depth by using its setOpenToDepth() method (where 0 indicates completely closed, 1 indicates show children, 2 indicates show children and grandchildren, etc.):

```
INode root = new MyNodeImpl("my root info");
ITreeContentProvider tree = new LazyNodeTreeContentProvider(root);
tree.setOpenToDepth(1);
```

Columns

Just like the "Table" on page 570 control, columns can be added to the Tree control in order to display content for each tree row. Table columns (including special columns, like Select Row columns) work in Trees exactly the same as they do in Tables. Additionally, special tree-specific "Tree Toggle" on page 580 column controls allow users to toggle tree rows expanded and collapsed. Usually, each tree should include a Tree Toggle control.

Also just like the Table control, columns can be made sortable by specifing a key by which to sort the tree in the column's "sort" property value. See "Standard Column" on page 568.

Asynchronous Properties

The Async Tree has a number of configurable properties in the Lazy Load group which can be used to customize its asynchronous behavior. It shares some of these properties in common with the standard "Hideable Controls" on page 194.

The "Lazy Load" properties unique to the Async Tree concern how much of the tree to load at a time. The "initialDepth" property determines how many levels of nodes to load on the client initially. A value of "0" indicates roots only; a value of "1" indicates roots and children, a value of "2" indicates roots, children, and grandchildren; etc.

The "refillDepth" property determines how many levels of nodes to load when the user tries to expand a node, and the node's children aren't on the client. A value of "0" indicates children only; a value of "1" indicates children and grandchildren; a value of "2" indicates

children, grandchildren, and great-grandchildren; etc. In most cases it's best to limit this to "0" or "1".

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display co	ontrol.	true
Value Properties					
value	Data bound to table.	ITreeContent Provider	Tree data model.		
var	Row Variable	String	Name of current is row within colum	row object (used to referent	ence current
Lazy Load Properti	es				
initialDepth	Initial Depth	int	Levels of nodes to pull down to the client initially.	0 roots only, 1 roots + children, 2 = roots + children + grandchildren, etc.	0
progressDelay	Progress Delay	int	Milliseconds to w progress dialog (r	rait before showing re-sorting only).	0
progressFlash OnComplete	Flash On Load	boolean	True to flash indiv finish loading asy	vidiual rows when they rnchronously.	true
progressMsg	Progress Message	String	Message to display in progress dialog (re-sorting only).		Defaults to localized message; in English, "Loading"
refillDepth	Refill Depth	int	Levels of nodes to pull down to the client when user expands a node whose children aren't on the client.	0 children only, 1 children + grandchildren, 2 = children + grandchildren + great-grandchildren, etc.	0
refreshOnShow	Refresh On Show	boolean	True to refresh content every time control is toggled visible; false to refresh content only if control is initially hidden and then toggled visible.		false

Name	Display Name	Туре	Description	Choices	Default
suppressInputs	Suppress Inputs	Collection	List of ids of controls for which to disable validation/update processing when handling an asynchronous request (comma separated).		
Expert Properties					
filter	Filter	String	A filter expression filterable data mo	n. Used to filter tables b	ound to
maxOrdinal	Max Ordinal	int		er of columns at a time l ss than one signals "unli	
Expert Display Pro	perties				
bgcolor	Background Color	Color	Table background	l color.	
border	Border	int	Border width in pixels, for both outer table border and borders between cells.		
cellpadding	Cell Padding	String	Padding within cell content between margin and text.		
cellspacing	Cell Spacing	String	Margin between o	cell border and cell cont	ent.
columnClasses	Column CSS Classes	Collection	CSS class to use for repeating).	or each column, from le	ft to right (not
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RT	TL (right-to-left)
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-footer	
frame	Frame	String	Outer border style.	none (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)	
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-subhe	ader
lang	Language	String	Language code fo	or content.	

Name	Display Name	Туре	Description	Choices	Default
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	portlet-section-body, p alternate	ortlet-section-
rowHeight	Row Height	String	Height of each row (for example 20en	w in pixels (for example n).	20px) or ems
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	none
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	HTML "class" attr	ibute value.	
summary	Summary	String	Summary text.		
title	Element Title	String	HTML "title" attri	bute value (often display	red as a tooltip.
width	Width	String	Width of table in 1	pixels ('100') or percent ('	100%').
Expert Client Side E	Events				
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-cl	icked.	
onkeydown	Key Down	script	Key pressed when	n control is focused.	
onkeypress	Key Press	script	Key pressed and a	released when control is	focused.
onkeyup	Key Up	script	Key released whe	n control is focused.	
onmousedown	Mouse Down	script	Mouse button pre	ssed when over control.	
onmousemove	Mouse Move	script	Mouse moved wh	en over control.	
onmouseout	Mouse Out	script	Mouse moved out	t of control boundaries.	
onmouseover	Mouse Over	script	Mouse moved into control boundaries.		
onmouseup	Mouse Up	script	Mouse button rele	eased when over control.	

javax.faces.Column ("Basic Column" on page 547)

Facets

Name	Display Name	Description	Allowed Types
header	Header	Table header.	Any
footer	Footer	Table footer.	Any
empty	Empty Message	Message displayed in place of content when table is empty.	Any

Atom Feed Icon

com.webmethods.caf.faces.export.AtomFeedIcon

Base Controls Component Library

A icon that exports the contents of a specified table to a syndication feed (such as RSS/Atom). This control must be contained within a "Form" on page 268; The Form control's "requireSessionToken" property must be set to false. The control works only in portlets, not standalone servlets.

To provide custom content for the export (for example, to provide custom feed metadata like title or last updated date), you can bind this control's "feedProvider" property value to a custom com.webmethods.caf.faces.data.IContentProvider instance, and/or this control's "entriesProvider" property value to a separate

com.webmethods.caf.faces.data.ITableContentProvider instance. A convenience IContentProvider implementation designed for atom feed metadata is the com.webmethods.caf.faces.data.export.AtomFeedExportProvider class. It maps the properties of an existing IContentProvider (which can be backed by a Java object, a map, and so forth) to a custom set of feed metadata elements and/or attributes, with each element or attribute's content specified via a binding expression. A convenience ITableContentProvider implementation designed for atom entry metadata is the com.webmethods.caf.faces.data.export.AtomEntriesExportProvider class. It maps the content of an existing ITableContentProvider (usually the provider which provides the raw data for the table) to a custom set of entry metadata elements and/or attributes, with each element or attribute's content also specified via a binding expression.

The following example demonstrates the creation of an AtomFeedExportProvider and an AtomEntriesExportProvider in custom page bean methods. (You would bind the properties for which these methods are getters, "myAtomFeedProvider" and "myAtomEntriesProvider", to the AtomFeedIcon's "feedProvider" and "entriesProvider" property values.) The example assumes there is already an existing table content-provider for the raw table data, accessible via the getMyExistingTableContentProvider() method. The getMyAtomFeedProvider() method creates a map of property name, value pairs for the feed metadata, specifing the standard atom "id", "title", and "author" elements, and including a custom "newToday" property (which will be rendered as an attribute, since its value is a simple string value). Although the AtomFeedExportProvider allows for

remapping keys from the feed's original content-provider, in this case the content-provider is just a static map, so there's no need to do so. Therefore the keys are generated with the AtomFeedExportProvider's static createDefaultKeys() method, which simply performs an "identity" mapping, mapping each property to itself. This static method could also be used to get the default set of keys for a dynamic content-provider, and remap just one or two custom properties, leaving the rest of the properties as is.

The getMyAtomEntriesProvider() method maps three standard atom entry elements, "id", "title", and "updated" — plus two custom properties, "priority" and "priorityName" — to row data, using binding expressions (the name of the row variable, in this case "row", must be specified as the third argument to the AtomEntriesExportProvider constructor). For each atom entry, the content of the entry's "id" element will be generated from the value of the table row's "projectId" and "id" properties; the content of the entry's "title" element will be generated from the value of the table row's "title" property; and so on.

```
public IContentProvider getMyAtomFeedProvider() {
    Map feedProperties = new HashMap();
    // standard atom metadata
    feedProperties.put("id", "issues");
    feedProperties.put("title", "Issues");
    feedProperties.put("author", "DevTrack");
    // custom metadata
    feedProperties.put("newToday", getNewToday());
    MapContentProvider feed = new MapContentProvider(feedProperties);
   Map keys = AtomFeedExportProvider.createDefaultKeys(feed, "feed");
   return new AtomFeedExportProvider(feed, keys, "feed"); }
public ITableContentProvider getMyAtomEntriesProvider() {
    ITableContentProvider existingProvider = getMyExistingTableContentProvider();
   Map keys = new HashMap(); // standard atom metadata keys.put("id",
   createValueBinding("#{row.projectId}_#{row.id}")); keys.put("title",
    createValueBinding("#{row.title}")); keys.put("updated",
    createValueBinding("#{row.lastModifiedDate}"));
    // custom metadata
   keys.put("priority", createValueBinding("#{row.priorityValue}"));
   keys.put("priorityName", createValueBinding("#{row.priorityName}"));
   return new AtomEntriesExportProvider(existingProvider, keys, "row"); }
```

The atom feed might be rendered as in the following example. Note that the export processing handles escaping property values (so for example, the Name should be "MWS" value is escaped as Name should be " MWS"):

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xml:base="http://mws:8585/" xml:lang="en-US"</pre>
     xmlns="http://www.w3.org/2005/Atom"
           xmlns:x="http://webmethods.com/caf/faces/data/export/atom/custom"
      x:newToday="2">
          <id>id>issues</id>
          <title>Issues</title>
          <updated>2007-05-22T21:18:40Z</updated>
          <author>
               <name>DevTrack</name>
          </author>
          <generator uri="http://webmethods.com/caf/faces/data/export/atom"</pre>
               version="1.0">webMethods CAF AtomExportBean</generator>
          <entry x:priority="1" x:priorityName="Critical">
               <id>higgins_123</id>
               <title>Nothing Works</title>
               <updated>2007-05-22T21:18:40Z</updated>
          </entry>
          <entry x:priority="3" x:priorityName="Medium">
               <id>higgins_ABC</id>
               <title>Name should be &quot; MWS&quot; </title>
               <updated>2007-05-22T12:22:33Z</updated>
          </entry>
          <entry x:priority="4" x:priorityName="Low">
               <id>getraer_TLC</id>
               <title>I like monkey</title>
               <updated>2007-03-02T01:12:11Z</updated>
          </entry>
</feed>
```

To render custom properties (like the "newToday" feed property and the "priority" and "priorityName" entry properties) as elements, compose their values out of either Map or IContentProvider objects. Use the special "content" key to indicate that a property of the Map or IContentProvider should be rendered as the content of the custom element. In the following example, the feed's "newToday" property is built out of a map, with the map's "content" property set to a string value. Therefore, the feed's "newToday" property will be rendered as an element, with the specified string value as the "newToday" element's content.

Also in the following example, the entries' "priority" property is built out of a map, with the map's "content" property bound to the current table row's "priorityName" property, and the map's "number" property bound to the current table row's "priorityValue" property. Therefore, each entry's "priority" property will be rendered as an element, with the table row's "priorityName" property as the entry's "priority" element's content, and the

table row's "priorityValue" property as the "number" attribute of the entry's "priority" element.

```
public IContentProvider getMyAtomFeedProvider() {
   Map feedProperties = new HashMap();
    // standard atom metadata
   feedProperties.put("id", "issues");
    feedProperties.put("title", "Issues");
    feedProperties.put("author", "DevTrack");
    // custom metadata
   Map newToday = new HashMap();
   newToday.put("content", getNewToday());
    feedProperties.put("newToday", newToday);
    MapContentProvider feed = new MapContentProvider(feedProperties);
   Map keys = AtomFeedExportProvider.createDefaultKeys(feed, "feed");
   return new AtomFeedExportProvider(feed, keys, "feed");
public ITableContentProvider getMyAtomEntriesProvider() {
    ITableContentProvider existingProvider = getMyExistingTableContentProvider();
   Map keys = new HashMap(); // standard atom metadata keys.put("id",
   createValueBinding("#{row.projectId}_#{row.id}")); keys.put("title",
   createValueBinding("#{row.title}")); keys.put("updated",
   createValueBinding("#{row.lastModifiedDate}"));
    // custom metadata
   Map priority = new HashMap();
   priority.put("content", createValueBinding("#{row.priorityName}"));
   priority.put("number", createValueBinding("#{row.priorityValue}"));
   feedProperties.put("priority", priority);
   return new AtomEntriesExportProvider(existingProvider, keys, "row"); }
```

The atom feed might be rendered in the following way:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xml:base="http://mws:8585/" xml:lang="en-US"</pre>
    xmlns="http://www.w3.org/2005/Atom"
        xmlns:x="http://webmethods.com/caf/faces/data/export/atom/custom">
        <x:newToday>2</x:newToday>
        <id>id>issues</id>
        <title>Issues</title>
        <updated>2007-05-22T21:18:40Z</updated>
        <author>
            <name>DevTrack</name>
        </author>
        <generator uri="http://webmethods.com/caf/faces/data/export/atom"</pre>
            version="1.0">webMethods CAF AtomExportBean</generator>
        <entry>
            <x:priority number="1">Critical</x:priority>
            <id>higgins_123</id>
            <title>Nothing Works</title>
            <updated>2007-05-22T21:18:40Z</updated>
        </entry>
        <entry>
            <x:priority number="3">Medium</x:priority>
            <id>higgins_ABC</id>
            <title>Name should be &quot; MWS&quot; </title>
            <updated>2007-05-22T12:22:33Z</updated>
        </entry>
        <entry>
            <x:priority number="4">Low</x:priority>
            <id>getraer_TLC</id>
            <title>I like monkey</title>
            <updated>2007-03-02T01:12:11Z</updated>
        </entry>
</feed>
```

You may also specify a substitute URL for the syndication feed, using this control's "feedUrl" property. If you do so, you will need to manually generate the feed at that location (for example, using a Java servlet).

Using a Feed Reader

When a user attempts to subscribe to a feed generated in this manner, the user's feed reader first must be able to access My webMethods Server. This usually means the user must use a desktop-based feed reader (the user can use a Web-based feed reader only if My webMethods Server is accessible outside of the corporate firewall, or if the Web-based feed reader is hosted within the same firewall as My webMethods Server). The user must also configure the feed with his or her My webMethods Server credentials. The simplest way to do this is for the user to add "username" and "password" parameters to the feed

URL. For example, if the atom feed URL was the following (an oversimplified feed URL for demonstration purposes):

http://mws:8585/meta/default/wm_xt_fabricfolder/0000005202?export=atom

the user can add his or her username and password to the url as in the following:

 $http://mws:8585/meta/default/wm_xt_fabricfolder/0000005202?export=atom \&username=alice\&password=alicepassword$

Some feed readers also allow authentication credentials to be specified directly in the UI. To enable the feed reader to authenticate with those credentials, a system administrator must first set the authentication scheme of the page the feed is on to use the "basic" authentication scheme. See the *My webMethods Server Administrator's Guide* for details on setting a page's auth scheme.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control id.	
rendered	Rendered	boolean	True to display control.	true
Export Properties				
for	For	Control ID Reference	Table ID. If contained within a table, defaults to that table.	
Expert Export Properties				
exportCharacter Encoding	Export Encoding	String	Default character encoding.	UnicodeLittle
exportFileName	Export File Name	String	Name to use for exported file.	export.csv
exportForce Download	Export Force Download	boolean	True to force "Save As" dialog (instead of displaying content in browser).	false
entriesProvider	Entries Provider	ITableContent Provider	Custom table content provider to use for feed entries.	
feedProvider	Feed Provider	IContentProvi der	Custom content provider to use for feed metadata.	
feedUrl	Feed URL	String	URL to use for feed (instead of generating content from table or entriesProvider and feedProvider).	

Any. Children are displayed as the icon label (after the icon).

javax.faces.Parameter children are added as request parameters to the link.

com.webmethods.caf.faces.portleturl.PortletUrlScript ("Portlet Url Script" on page 496) children may be used to hook into client-side javascript events (onClick).

Facets

None.

Basic Column

javax.faces.Column

CAF JSF Base Controls Component Library

Determines the content of a data table's column. A standard JavaServer Faces control. The content of the column will be rendered multiple times, one for each row displayed by the table.

The column control also optionally specifies the sorting behavior for the column. See "Standard Column" on page 568 for further explanation.

Standard JSF Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control.		true

Extended CAF Properties

Name	Display Name	Туре	Description	Choices	Default
Display Properties					
align	Alignment	String	Horizontal alignment of content.	left, center, right	
group	Group	String	Display title of column group (for Add/Remove Columns dialog). Null or "" is the Default group.		lumns
label	Label	String	Display title of column (for Add/Remove Columns dialog). This will not affect the column header. Column header content comes from the "header" facet.		

Name	Display Name	Туре	Description Choices	Default
nowrap	No Wrap	boolean	True if column should be as wide as widest line in fal the column content.	
optional	Optional	boolean	True if column displayed only after chosen via Add/Remove Columns dialog.	false
valign	Vertical Alignment	String	Vertical alignment of top, middle, bottom content.	
width	Width	String	Width of column in pixels ('100') or percent ('100%)	o').
Sorting				
ascending	Ascending	boolean	Default ascending value. True to sort column from true A-Z, 1-10, Jan-Dec, etc.	
ordinal	Ordinal	int	Default ordinal value. Primacy of this column in sorting the table; 0=none (not sorted on column), 1=primary, 2=secondary, etc.	
sort	Sort	Object	Key by which to sort column.	
Expert Sorting				
allowToggle	Allow Toggle	boolean	True to allow user to toggle between ascending and descending.	true

Any.

Standard JSF Facets

Name	Display Name	Description	Allowed Types
header	Header	Column header.	Any
footer	Footer	Column footer.	Any

Extended CAF Facets

None.

Calendar

com.webmethods.caf.faces.table.CalendarTable

Base Controls Component Library

The Calendar control is a specialized form of table. Instead of displaying cells arranged in rows and columns, it displays cells as days within a calendar layout. The control's primary utility is to display calendar events within this layout. The calendar supports only asynchronous actions.

The form of the calendar is controlled by the calendar Display Units and Rows properties. Display units determine how the days are arranged, either as a full calendar month, a single week, or a single day. The Rows property determines how many of this calendar unit are to be displayed at one time. For example, specifying units of "days" and setting rows to 3 will display a three-day horizontal span of days. If a paging control is attached to the calendar, it moves forward or back three days at a time.

All tables display a range of items based upon a zero-based index. For the calendar to have a zero index, it must have a Start Date. If no Start Date is specified, a default value is calculated based upon the current Display Units. For example, if the Display Units property is set to "week," the default start date is set to the week prior to the current week. The user can never page backward to before the calendar Start Date. The End Date property specifies the last date visible by the calendar, but may be left empty indicating an open-ended range.

The events to be displayed within the Calendar are provided by binding the Calendar Provider property to an ICalendarEventProvider, which exposes a List of objects that implement the ICalendarEvent interface. This can be an actual ICalendarEventProvider implementation, a List containing ICalendarEvent objects, or an array of ICalendarEvents.

Events are placed into two categories when displayed by the Calendar control:

- Multi-day events (events that last longer than one day or span over midnight into the next day)
- Sub-day events.

The calendar draws these event types differently. Multi-day events span horizontally and take up a single row while sub-day events stack vertically within a day cell.

The contents of the calendar are displayed by means of the Header and Event facets. The Header facet is a set of controls that draws the calendar header, and usually contains a set of text controls that display the range of dates currently displayed. The event facet contains a set of controls that draw each event within the calendar, and allow you to specify exactly how you want the event to be drawn.

The facet controls can bind to any expression, however the calendar exposes three scoped variables that represent the current item being drawn. The properties exposed by these variables are visible in the Bindings View:

- -Unit—A CalendarUnit object exposing information about the currently displayed unit (month, week, day). this variable is useful when drawing the calendar header.
- -Day An ICalendarDay object exposing information about the current day being drawn. The variable contains a single Date.
- Event—The ICalendarEvent object currently being drawn. Although this can be implemented by any object, the interface exposes the event Start/Stop date, event type, header, and body. This information is primarily used within the Event facet controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of the first row be displayed. This property is mu exclusive with the Initial Start Da property.	ıtually	0
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display control.		true
rows	Rows	int	Number of calendar units in page displayed. The default is 1, and salways be a positive number.		1
Value Properties					
events	Calendar Events	ICalendarEvent Provider	Provides the ICalendarEvents to Calendar. Values of type ICalend be automatically wrapped as ICalendarEventProvider.	1 2	
value	Calendar Provider	ICalendar Provider	The calendar provider that handles the grouping of days into appropriate units, exposing each Day in an ICalendarDay object. If this property is left blank, a DefaultCalendarProvider is used automatically. It is unlikely that any other provider will be necessary here.		ay in an lank, a lly. It is
var	Calendar Unit Variable	String	Name of the current calendar unit object used to reference the curre drawn).	•	

Name	Display Name	Туре	Description	Choices	Default
dayVar	Day Variable	String	Name of current calendar Day (A object used to reference the curre drawn).		
eventVar	Event Variable	String	Name of current calendar event (object used to reference the curredrawn).	•	
Display Properties					
units	Display Units	String	The type of units to display in the calendar.	Week, Month, Year	
startDate	Start Date	Date	The Start Date of the range of dates displayed by the calendar. If the property is left blank, a reasonable default value is calculated. The user can never page backward to before this date.		nable
endDate	End Date	Date	The End Date of the date range displayed by the calendar. An optional property. If the property is left blank, the user can page forward indefinitely.		ty is left
initialDate	Initial Display Date	Date	The Date initially displayed by the calendar control. When provided, this value is used to calculate the value of the Index property.		
allDayEvents	All Day Events	Date	A flag to treat all events as all day or multi-day events. This property may provider a cleaner looking display when dealing with a mix of multi-day and sub-day events.		looking
headerCssClass	Header CSS Class	String	The CSS class to use for the calen	dar header.	
weekdayHeader CssClass	Weekday Header CSS Class	String	The CSS class to use for the Day column headers.		ders.
dayCssClass	Day CSS Class	String	The CSS class to use for the Day	cell.	
eventCssClass	Event CSS Class	String	The CSS class to use for the Event.		
timeslotCss Class	Timeslot Label CSS Class	String	The CSS class to use for the time-slot label. Only applicable if the Display Timeslot Label Column property is true.		•
startHour	Start Hour	int	The starting hour to display with For example, a value of 8 would time-slot display to begin at 8:00	limit the	

Name	Display Name	Туре	Description	Choices	Default
endHour	End Hour	int	The ending hour to display with example, a value of 18 would lin display to end at 6:00 p.m.		
timeIncrement	Time Increment	int	The number of minutes per time default value is 30, indicating a t for each half hour.		30
timeslotLabel Format	Timeslot Label Format	String	The format used when displayin column. Defaults to the user's prapplicable when the Display Timproperty is set to true.	eference. O	nly
displayAll Intervals	Display All Time Intervals	String	A flag to control the display of time-slot rows within a day. When true, a row is drawn for each time slot regardless of whether it contains an event, and will look similar to the Microsoft Outlook calendar. When false, any time-slot row that does not contain an event is not drawn. This mode conserves space when drawing a row of calendar days, and looks similar to the previous calendar control.		me slot nd will ar. contain space

None

Facets

Name	Display Name	Description	Allowed Types
header	Header	Control used to draw the Calendar header.	Any
event	Event	Control used to draw each event.	Any

Categorized Table

com. we bmethods. caf. faces. table. Categorized Table

Base Controls Component Library

A table whose content comes from either a javax.faces.model.DataModel object, or a com.webmethods.caf.faces.data.ITableContentProvider object. Standard list-like containers, such as java.util.List, are wrapped autonomatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset

Each item in the list is mapped to a row in the table. Columns are configured by adding column children. With a categorized table, the column with the primary sort (a sort ordinal of 1) is rendered as the category for the group of rows which have the same value for that column. All other columns are rendered once for each row.

To demonstrate how the Categorized Table differs from a regular Table, consider the following regular Table, which is sorted first by Product, then by Quantity:

	Quarterly Sales		
Product	Quantity	Customer	
Hammer	300	General Mills	
Hammer	2	General Electric	
Saw	12	General Dynamics	
Saw	10	General Motors	
Screwdriver	70	General Dynamics	
Screwdriver	1	General Electric	

If we formatted the same data as a Categorized Table, it would look like the following. Rows with the same value for the Product column are displayed as a category of rows, grouped by the Product column value:

Quarterly Sales		
Product	Quantity	Customer
Hammer		
	300	General Mills
	2	General Electric
Saw		
	12	General Dynamics
	10	General Motors
Screwdriver		
	70	General Dynamics
	1	General Electric

Name	Display Name	Туре	Description	Choices	Default
General Properties					
first	First	int	Zero-based index of display.	first row in page to	0
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cont	rol.	true
rows	Rows	int	Number of rows in j signals "all".	page to display. Less	than one
Value Properties					
value	Data bound to table.	ITableContentPr ovider	Number of rows in page 15 signals "all".	page to display. Less	than one
var	Row Variable	String	Name of current row object (used to reference current row within column content).		

Name	Display Name	Туре	Description	Choices	Default	
Expert Properties						
filter	Filter	String	-	A filter expression. Used to filter tables bound to filterable data models.		
maxOrdinal	Max Ordinal	int		Maximum number of columns at a time by which table can be sorted. Less than one signals "unlimited".		
Expert Display Prop	perties					
bgcolor Background Color Table background color. Color						
border	Border	int	Border width in particular and borders between	ixels, for both outer to een cells.	able border	
cellpadding	Cell Padding	String	Padding within ce text.	ll content between m	argin and	
cellspacing	Cell Spacing	String	Margin between c	ell border and cell co	ntent.	
columnClasses	Column CSS Classes	Collection	CSS class to use fo (not repeating).	CSS class to use for each column, from left to right (not repeating).		
customRow Class	Custom Row CSS Class	String		Binding expression that specifies a custom CSS class name to add to the current row.		
dir	Directionality	String	Text direction hint.	LTR (left-to-right) to-left)	, RTL (right-	
fixedHeight	Fixed Height	boolean	True if the table sheight, based on the per page.	nould have a fixed the number of rows	false	
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-foo	oter	
frame	Frame	String	Outer border style.	void (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)		

Name	Display Name	Туре	Description	Choices	Default	
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-sul	oheader	
lang	Language	String	Language code for	Language code for content.		
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	for each column, section-alternate from top to		
rowHeight	Row Height	String	Height of each row ems (for example 2	in pixels (for examp 0em).	ole 20px) or	
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	void	
style	CSS Style	String	HTML "style" attrik	oute value.		
styleClass	CSS Class	String	HTML "class" attrib	oute value.		
summary	Summary	String	Summary text.			
title	Element Title	String	HTML "title" attrib tooltip.	ute value (often disp	played as a	
width	Width	String	Width of table in p	ixels ('100') or percer	nt ('100%').	
Expert Client Side	Events					
onclick	Click	script	Control clicked.			
ondblclick	Double Click	script	Control double-clic	ked.		
onkeydown	Key Down	script	Key pressed when	control is focused.		
onkeypress	Key Press	script	Key pressed and re	leased when control	is focused.	
onkeyup	Key Up	script	Key released when	control is focused.		
onmousedown	Mouse Down	script	Mouse button pres	sed when over contr	ol.	
onmousemove	Mouse Move	script	Mouse moved whe	n over control.		
onmouseout	Mouse Out	script	Mouse moved out	of control boundarie	es.	

Name	Display Name	Туре	Description Choices Default
onmouseover	Mouse Over	script	Mouse moved into control boundaries.
onmouseup	Mouse Up	script	Mouse button released when over control.

javax.faces.Column ("Standard Column" on page 568)

Facets

Name	Display Name	Description	Allowed Types
header	Header	Table header.	Any
footer	Footer	Table footer.	Any
empty	Empty Message	Message displayed in place of content when table is empty.	Any

Column Sort Link

com.webmethods.caf.faces.table.ColumnSortLink

Base Controls Component Library

A control that draws a link to sort a table by specified column. This functionality is incorporated automatically by the header in a "Standard Column" on page 568 header.

If you drop this control into a column's header or footer, you need to do nothing else to enable "form" sorting, other than to set the column's "sort" property to the appropriate sort key; clicking the link maintains the current JSF form state. If you place this control elsewhere on a page, you must specify the control's "for" property. This control must be contained within a Command "Form" on page 268 to do "form" sorting.

To enable "restful" sorting, you must set this target table's "sortParamName" property to the name of the request parameter from which the target table derives its "sort" property. See "Table" on page 570.

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	Table to which this applies. If unspecified, pages the nearest ancestor table.	

Name	Display Name	Туре	Description	Default
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
label	Label	String	Control label (rendered before children); displayed as the sort link label	

Any. Children are used as the link's label.

Facets

None.

Data Pages

com.webmethods.caf.faces.table.DataPages

Base Controls Component Library

A control that draws 1 ... 12 13 14 15 16 17 18 19 ... 100 links that allow a user to jump several pages forward or backward.

If you drop this control into a table's header or footer, you need to do nothing else to enable "form" paging (clicking a link maintains the current JSF form state). If you place this control elsewhere on a page, you must specify the control's "for" property. This control must be contained within a Command "Form" on page 268 to do "form" paging.

To enable "restful" paging, you must set this target table's "firstParamName" property to the name of the request param from which the target table derives its "first" property. See "Table" on page 570.

Name	Display Name	Туре	Description	Default
General Propertion	es			
for	For	String	Table to which this applies. If unsp pages the nearest ancestor table.	ecified,
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description	Default
Display Propertie	es			
maxPages	Max Pages	int	Maximum number of page links to display. If table has more than this number of pages, the control will draw an ellipsis () in place of some of the distant links.	10
showAlways	Show Always	boolean	True to display control even if there's only one page.	false

None.

Facets

None.

Data Prev/Next

com.webmethods.caf.faces.table.PrevNext

Base Controls Component Library

A control that draws **«Previous and Next»** links that page a table one page backward or forward.

If you drop this control into a table's header or footer, you need to do nothing else to enable "form" paging (clicking **«Previous or Next»** maintains the current JSF form state). If you place this control elsewhere on a page, you must specify the control's "for" property. This control must be contained within a Command "Form" on page 268 to do "form" paging.

To enable "restful" paging, you must set this target table's "firstParamName" property to the name of the request param from which the target table derives its "first" property. See "Table" on page 570.

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	Table to which this applies. If unspecified, pages the nearest antable.	cestor

Name	Display Name	Туре	Description	Default
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
expandToEdges	Expand To Edges	boolean	True to force «Previous text to left edge and Next» text to right edge of container.	false
nextText	Next Text	String	Custom Next» label.	
prevText	Prev Text	String	Custom «Previous label.	
showAlways	Show Always	boolean	Show «Previous and Next» even if user can't page to the previous or next page.	false

Any. Children are placed between «Previous and Next» labels.

Facets

None.

Data Total

com. we bmethods. caf. faces. table. Data Total

Base Controls Component Library

A control that displays the row indicies of the current page of the table, as well as the total number of rows in the table.

If you drop this control into a tables's header or footer, you need to do nothing else. If you place this control elsewhere on a page, you must specify the control's "for" property.

Name	Display Name	Туре	Description	Default
General Properties				
changePageSize	Change Page Size	String	True to allow user to change the number of rows per page.	false

Name	Display Name	Туре	Description	Default
for	For	String	Table to which this applie unspecified, pages the near ancestor table.	
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
standardMsg	Standard Message	String	Standard message to disp Defaults to a localized me in en-US "'{0}' - '{1}' of '{2}" 10 of 100).	essage;
emptyMsg	Empty Message	String	Displays when table is empty. Defaults to a localized message; in en-US "Empty".	
atEndMsg	At End Message	String	Displays when table is at the last row; in en-US "'{0}' of '{2}" (ex: 100 of 100).	
beyondEndMsg	Beyond End Message	String	Displays when table is beyond the last row. Defaults to a localized message; in en-US "Beyond End of Data".	
unknownTotalMsg	Unknown Total Message	String	Displays when the total nof rows in the table is unk Defaults to a localized me in en-US "'{0}' - '{1}"" (ex: 1	known. essage;

None.

Facets

None.

Data Total Selected

com.webmethods.caf.faces.table.DataTotalSelected

Base Controls Component Library

A table that displays the number of selected rows in the table.

If you drop this control into a table's header or footer, you need to do nothing else. If you place this control elsewhere on a page, you must specify the control's "for" property.

Properties

Name	Display Name	Туре	Description	Default
General Properti	es			
for	For	String	Table to which this applies. If un pages the nearest ancestor table.	specified,
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Propertie	es			
standardMsg	Standard Message	String	Standard message to display. Defaults to a localized message; in en-US "'{0}' selected' (ex: 10 selected).	

Children

None.

Facets

None.

Export Table Button

com. we bmethods. caf. faces. export. Export Table Button

Base Controls Component Library

A button that exports the contents of a specified table to a CSV file. Must be contained within a Command "Form" on page 268.

To provide custom content for the export (for example, to provide a set of columns with more machine-friendly formatted data), you can bind this control's "exportProvider" property value to a separate com.webmethods.caf.faces.data.ITableContentProvider instance. A convenience ITableContentProvider implementation designed for CSV export

is the com.webmethods.caf.faces.data.export.CSVExportProvider class. It maps the content of an existing ITableContentProvider (usually the provider which provides the raw data for the table) to a custom set of columns, with each column's content specified by a binding expression.

The following example demonstrates creating a CSVExportProvider in a custom page bean method. (You would bind the property for which this method is a getter, "myExportProvider", to this control's "exportProvider" property value.) It assumes that there is already an existing table content-provider for the raw table data, accessible via the getMyExistingTableContentProvider() method. It creates four columns: "ID", "Title", "Priority", and "Priority Name"; it specifies the content of these columns with binding expressions: "#{row.projectId}_#{row.id}", "#{row.title}", "#{row.priorityValue}", and "#{row.priorityName}" (the name of the row variable, in this case "row", must be specified as the third argument to the CSVExportProvider constructor.) For each row, the content of the "ID" column in the exported table will be generated from the value of the existing-table row's "projectId" and "id" properties; the content of the "Title" column will be generated from the value of the existing-table row's "title" property; and so on.

```
public ITableContentProvider getMyExportProvider() {
    ITableContentProvider existingProvider = getMyExistingTableContentProvider();

Map keys = new LinkedHashMap();
    keys.put("ID", createValueBinding("#{row.projectId}_#{row.id}"));
    keys.put("Title", createValueBinding("#{row.title}"));
    keys.put("Priority", createValueBinding("#{row.priorityValue}"));
    keys.put("Priority Name", createValueBinding("#{row.priorityName}"));

return new CSVExportProvider(existingProvider, keys, "row");
}
```

Note that since the exported table's column labels are simply the column keys ("ID", "Title", "Priority", and "Priority Name"), it is a good practice to use string resources from a resource bundle instead of hardcoding the labels (the labels are hardcoded in this example for simplicity). Note also, a LinkedHashMap was used in the above example, which ensures that the columns will be exported in the order they were added to the keys map. The exported CSV might look like the following:

```
ID, Title, Priority, Priority Name
higgins_123, Nothing Works, 1, Critical
higgins_ABC, "Name should be "MWS"", 3, Medium
```

The export processing handles the escaping of column values (in the example output above, the Name should be "MWS" value is escaped as "Name should be ""MWS""").

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Export Properties				
for	For	Control ID Reference	Table ID. If contained defaults to that table.	within a table,
showEncoding Dialog	Show Encoding Dialog	boolean	True to show character encoding dialog.	true
Expert Export Prope	erties			
exclude Components CSV	Exclude Component IDs	String	CSV string of control l within the table) to ex- export.	
exportCharacter Encoding	Export Encoding	String	Default character encoding.	UnicodeLittle.
exportFileName	Export File Name	String	Name to use for exported file.	export.csv

Children

None.

Facets

None.

Hidden Column

com.webmethods.caf.faces.table.HiddenColumn

Base Controls Component Library

A control that adds a hidden column to a table. The content of this hidden column can be accessed via client-side script.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Children

None.

Facets

None.

Newsfeed

com. we bmethods. caf. faces. table. specialized. News feed

Base Controls Component Library

A control that displays the entries of a syndicated feed (RSS or Atom).

Display Name	Туре	Description	Choices	Default
ID	String	Control id.		
Rendered	boolean	True to display control	•	true
URL	String	Feed url.		
	ID Rendered	ID String Rendered boolean	ID String Control id. Rendered boolean True to display control	ID String Control id. Rendered boolean True to display control.

Name	Display Name	Туре	Description	Choices	Default
Display Propertion	es				
layout	Layout	String	Location of entries table relative to entry details pane.	table-top, table- bottom, table-left, table-right	table-top
rows	Rows	int	Number of rows per per	Number of rows per page of the table.	
size	Size	String	Height of entry details ems (10em).	Height of entry details pane in pixels (10px) or ems (10em).	
Expert Display P	roperties				
style	CSS Style	String	HTML "style" attribute	e value.	
styleClass	CSS Class	String	HTML "class" attribute value.		
title	Title	String	HTML "title" attribute value (often displayed as a ToolTip).		

None.

Facets

None.

Row Headers Column

com. we bmethods. caf. faces. table. Row Headers Column

Base Controls Component Library

A table column that displays as row-wise headers. See "Standard Column" on page 568 control for more information about using columns.

Name	Display Name	Туре	Description	Choices	Default
General Proper	ties				
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display control.		true
Display Proper	ties				
align	Alignment	String	Horizontal alignment of content.	left, center, rig	ht

Name	Display Name	Туре	Description	Choices	Default
group	Group	String	Display title of the column group (for an Columns dialog) is the Default group.	n Add/Remove	Null or ""
label	Label	String	Display title of the column (for an Add dialog). This label does not affect the coheader content comes from the "header	olumn header. Co	
nowrap	No Wrap	boolean	True if the column should be as wide as in the column content.	the widest line	false
optional	Optional	boolean	True if the column is displayed only aft an Add/Remove Columns dialog.	er chosen via	false
valign	Vertical Alignment	String	Vertical alignment of content.	top, middle, bo	ottom
width	Width	String	Width of column in pixels (100px) or pe	ercent (100%).	
Sorting					
ascending	Ascending	boolean	Default ascending value. True to sort co to Z, 1 to 10, Jan to Dec, and so forth.	olumn from A	true
ordinal	Ordinal	int	Default ordinal value. Primacy of this c sorting the table; 0=none (not sorted on 1=primary, 2=secondary, and so forth.		0
sort	Sort	Object	Key by which to sort column.		
Expert Sorting					
allowToggle	Allow Toggle	boolean	True to allow the user to toggle between and descending order.	n ascending	true

Any.

Facets

Name	Display Name	Description	Allowed Types
header	Header	Column header; incorporates Column Sort Link functionality.	Any

Scrollbar Column

com.webmethods.caf.faces.table.ScrollbarPager

Base Controls Component Library

A table column that displays a scroll bar which pages the table. While this control will work with any type of table, it will be frustrating to use unless paired with an "Async Table" on page 531 that has its "clientSideCache" property set to "true".

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Children

None.

Facets

None.

Standard Column

com.webmethods.caf.faces.table.html.HtmlTableColumn

Base Controls Component Library

A control that determines the content of a data table's column. The content of the column will be rendered multiple times, one for each row displayed by the table.

The Standard Column control is different from the "Basic Column" on page 547 control in that it does not have a footer facet, and its header automatically incorporates the functionality of the "Column Sort Link" on page 557 control.

The column control also optionally specifies the sorting behavior for the column. Note that to enable sorting, both the table and each sortable column must be configured with an "id" property, and each sortable column must have a "sort" property.

The "sort" property specifies a key by which to sort the column. Usually this is a ValueBinding expression, bound to the current row object. For example, to sort on the "name" property of the current row, we'd use an expression "#{row.name}" (assuming the table's row variable -- configured via the "var" property -- was simply "row"). To sort on the zeroth index of a row object (when the row object is an array or list), we'd use an

expression "#{row0}". Any property whose values implement the java.lang.Comparable interface can be used for sorting. Note that the data model implementation provided to the table actually does the sorting, and some data models may use keys other than ValueBinding expressions.

The "ascending" and "ordinal" properties specify the default sorting state of the column. For example, say we have a table with a "type" column and a "name" column. By default we want to sort the table first by type, with larger type numbers first, and then by name, alphabetically:

```
TYPE NAME
8000 Phase-a-tron
8000 The Destruction Maker
7000 Laser, Blue
7000 Laser, Green
7000 Laser, Red
0400 Advil
0400 Anvil
```

We would set the "ascending" property of the "type" column to false, and the "ordinal" property of the "type" column to 1. We'd set the "ascending" property of the "name" column to true, and the "ordinal" property of the "name" column to 2.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control.		true
Display Properties					
align	Alignment	String	Horizontal alignment of content.	left, center, right	
group	Group	String	Display title of column group (fo dialog). Null or "" is the Default		Columns
label	Label	String	Display title of column (for Add, This will not affect the column had content comes from the "header"	eader. Column h	0,
nowrap	No Wrap	boolean	True if column should be as wide in the column content.	e as widest line	false
optional	Optional	boolean	True if column displayed only af Add/Remove Columns dialog.	ter chosen via	false

Name	Display Name	Туре	Description	Choices	Default
valign	Vertical Alignment	String	Vertical alignment of content.	top, middle, bottom	
width	Width	String	Width of column in pixels ('100p	ox') or percent ('1	00%').
Sorting					
ascending	Ascending	boolean	Default ascending value. True to sort column from A-Z, 1-10, Jan-Dec, etc.		true
ordinal	Ordinal	int	Default ordinal value. Primacy of this column in sorting the table; 0=none (not sorted on column), 1=primary, 2=secondary, etc.		0
sort	Sort	Object	Key by which to sort column.		
Expert Sorting					
allowToggle	Allow Toggle	boolean	True to allow user to toggle betwand descending.	veen ascending	true

Any.

Facets

Name	Display Name	Description	Allowed Types
header	Header	Column header; incorporates Column Sort Link functionality.	Any

Table

javax.faces.HtmlDataTable

CAF JSF Base Controls Component Library

A table whose content comes from either a javax.faces.model.DataModel object, or a com.webmethods.caf.faces.data.ITableContentProvider object. A standard JavaServer Faces control. Standard list-like containers, such as java.util.List, are wrapped autonomatically as an ITableContentProvider. The following containers are wrapped automatically:

- java.lang.Object array
- java.util.List
- javax.sql.rowset.CachedRowset

Each item in the list is mapped to a row in the table. Columns are configured by adding "Basic Column" on page 547 children.

To enable "restful" paging, you must set the tables's "firstParamName" property to the name of the request param from which the table derives its "first" property. For example, if the table binds the "myFirst" request parameter (where an example url might be '/myTable?myFirst=10') to its "first" property, using a value binding expression "#{param.myFirst}", you would set the value of the table's "firstParamName" property to "myFirst". Note that in order to bind the value of a request parameter to the table's "first" property, you must manually convert the request parameter value to a primitive int (see the following getMyFirst() method). The table does not need to be contained within a Command "Form" on page 268 to do "restful" paging.

To enable "restful" page sizing, you must set the tables's "rowsParamName" property to the name of the request param from which the table derives its "rows" property. For example, if the table binds the "myRows" request parameter (where an example url might be '/myTable?myRows=10') to its "rows" property, you would set the value of the table's "rowsParamName" property to "myRows". Note that in order to bind the value of a request parameter to the table's "rows" property, you must manually convert the request parameter value to a primitive int (see the following getMyRows() method). The table does not need to be contained within a Command "Form" on page 268 to do "restful" page sizing.

To enable "restful" sorting, you must set the tables's "sortParamName" property to the name of the request param from which the table derives its "sort" property. For example, if the table binds the "mySort" request parameter (where an example url might be 'myTable?mySort=name-,description+') to the "ordinals" and "ascending" properties of its contained columns, you would set the value of the table's "sortParamName" property to "mySort". Note that in order to bind the value of a request parameter to the table's "sort" property, you must manually convert the request parameter value to an ISortInfo object (see the following getMySort() method). The table does not need to be contained within a Command Form to do "restful" paging.

```
// Convert "myFirst" request parameter to a primitive int.
public int getMyFirst() {
   FacesContext context = FacesContext.getCurrentInstance();
   String s = (String) context.getExternalContext().getRequestParameterMap().get("myFirst");
   if (s != null)
   return Integer.parseInt(s);
   return 0;
// Convert "myRows" request parameter to a primitive int.
public int getMyRows() {
   FacesContext context = FacesContext.getCurrentInstance();
   String s = (String) context.getExternalContext().getRequestParameterMap().get("myRows");
   if (s != null)
       return Integer.parseInt(s);
   return 10;
// Convert "mySort" request parameter to structured ISortInfo data.
public ISortInfo getMySort() {
    FacesContext context = FacesContext.getCurrentInstance();
    String s = (String) context.getExternalContext().getRequestParameterMap().get("mySort");
    return (s != null ? new SortInfo(s) : new SortInfo());
}
```

Standard JSF Properties

Name	Display Name	Туре	Description	Choices	Default
General Propert	ies				
first	First	int	Zero-based index of first row in page to 0 display.		0
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control. tru		true
rows	Rows	int	Number of rows in page to display. Less than one signals "all".		
Value Properties	S				
value	Data bound to table.	ITableContent Provider	Number of rows signals "all".	s in page to display. Less tha	n one
var	Row Variable	String	Name of current row object (used to reference current row within column content).		
Expert Display F	Properties				
bgcolor	Background Color	Color	Table backgrour	nd color.	

Name	Display Name	Туре	Description	Choices	Default
border	Border	int	Border width in pixels, for both outer table border and borders between cells.		
cellpadding	Cell Padding	String	Padding within cell co	ntent between margin an	d text.
cellspacing	Cell Spacing	String	Margin between cell be	order and cell content.	
columnClasses	Column CSS Classes	Collection	CSS class to use for eac repeating).	ch column, from left to ri	ght (not
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTI to-left)	L (right-
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-footer	
frame	Frame	String	Outer border style.	void (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)	void
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-subhead	der
lang	Language	String	Language code for cor	ntent.	
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	portlet-section-body, p section-alternate	ortlet-
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	void
style	CSS Style	String	HTML "style" attribute	e value.	
styleClass	CSS Class	String	HTML "class" attribute	e value.	

Name	Display Name	Туре	Description	Choices	Default
summary	Summary	String	Summary text.		
title	Title	String	HTML "title" attribute	value.	
width	Width	String	Width of table in pixel	s ('100') or percent ('100%	(o').
Expert Client Side	Events				
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-clicked	l.	
onkeydown	Key Down	script	Key pressed when con	trol is focused.	
onkeypress	Key Press	script	Key pressed and releas	sed when control is focu	sed.
onkeyup	Key Up	script	Key released when cor	ntrol is focused.	
onmousedown	Mouse Down	script	Mouse button pressed	when over control.	
onmousemove	Mouse Move	script	Mouse moved when or	ver control.	
onmouseout	Mouse Out	script	Mouse moved out of co	ontrol boundaries.	
onmouseover	Mouse Over	script	Mouse moved into cor	trol boundaries.	
onmouseup	Mouse Up	script	Mouse button released	when over control.	

Extended CAF Properties

Name	Display Name	Туре	Description Default
Expert Properties			
firstRequestParam	First Request Param	String	Name of request parameter that specifies table's first row. Used only for "restful" paging.
rowsRequestParam	Rows Request Param	String	Name of request parameter that specifies number of rows for the table to display. Used only for "restful" paging.
sortRequestParam	Sort Request Param	String	Name of request parameter that specifies table's sort inf. Used only for "restful" paging.
filterRequestParam	Filter Request Param	String	Name of request parameter that specifies table's filter expression. Used only for "restful" paging.
filter	Filter	String	A filter expression. Used to filter tables bound to filterable data models.
maxOrdinal	Max Ordinal	int	Maximum number of columns at a time by which table can be sorted. Less than one signals "unlimited".

Name	Display Name	Туре	Description	Default
Expert Display Properti	es			
fixedHeight	Fixed Height	boolean	True if the table should have a fixed height, based on the number of rows per page.	false
rowHeight	Row Height	String	Height of each row in pixels (for example 20px) or ems (for example 20em).	

javax.faces.Column ("Basic Column" on page 547)

Standard JSF Facets

Name	Display Name	Description	Allowed Types
header	Header	Table header.	Any
footer	Footer	Table footer.	Any

Extended CAF Facets

Name	Display Name	Description	Allowed Types
empty	Empty Message	Message displayed in place of content when table is empty.	Any

Tree

com.webmethods.caf.faces.tree.Tree

Base Controls Component Library

A tree whose content comes from a com.webmethods.caf.faces.data.tree.ITreeContentProvider object. An ITreeContentProvider presents a flattened view of the tree, iterating over the tree structure in depth-first order. Although a developer can choose to implement a custom ITreeContentProvider, the standard My webMethods library provides two default implementations that can wrap most types of tree data:

- com.webmethods.caf.faces.data.tree.object.NodeTreeContentProvider
- com.webmethods.caf.faces.data.tree.object.ListTreeContentProvider.

NodeTreeContentProvider

The NodeTreeContentProvider implementation works best with tree data that is already in the form of a tree object model. A common example of such data is an XML document loaded into a W3C DOM tree. A developer needs only to provide the NodeTreeContentProvider with a data model that implements the com.webmethods.caf.faces.data.tree.INode interface, and the NodeTreeContentProvider handles the rest.

The com.webmethods.caf.faces.data.tree.object.XMLDOMNode class is a concrete INode implementation that wraps a W3C DOM tree object model. It can be instantiated with the DOM node that serves as the root of the tree, and an XPath string that specifies a unique ID for each node. For example, with the following XML, we'd use the uri attribute of each node for a unique ID:

We can use the XMLUtil.loadDocumentFromString() static utility method to load an XML document into a W3C DOM tree. We can create an XMLDOMNode tree model using the document's root element ('Root Folder') as the tree model root, and using the uri attribute as each node's unique ID. With the INode tree model in hand, we can instantiate a NodeTreeContentProvider. And this ITreeContentProvider is what the tree control uses (configured with the control's "value" property):

```
org.w3c.dom.Document doc = com.webmethods.caf.common.XMLUtil.loadDocumentFromString(xml);
INode root = new XMLDOMNode(doc.getDocumentElement(), "@uri");
ITreeContentProvider tree = new NodeTreeContentProvider(root);
```

ListTreeContentProvider

The other ITreeContentProvider implementation, ListTreeContentProvider, works best with tree data in the form of a list of rows, where each row has a property that identifies either the "parent" of the row or the "children" of the row. Tree data stored in a relational database usually takes this form. A developer needs to provide the ListTreeContentProvider with the list of rows, a ValueBinding expression that specifies a unique ID for each row, and either a ValueBinding expression that specifies the parent ID for each row, or a ValueBinding expression that specifies a list of child IDs for each row.

In the following example, the row data is simply a list of String arrays. The first item in each String array is the row's parent ID (specified via a ValueBinding expression

"#{row[0]}"). Rows where the parent ID is null are displayed as roots. The second item in each String array is the row's unique ID (specified via a ValueBinding expression "#{row[1]}"). So the first row, "alice" has two children, "albert" and "agnes":

```
java.util.List list = java.util.Arrays.asList(new Object[] {
   new Object[] {null, "alice", "apple", "astros"},
   new Object[] {null, "bob", "banana", "brewers"},
   new Object[] {null, "carla", "cantaloupe", "cubs"},
   new Object[] {null, "dirk", "dingleberries", "dodgers"},
   new Object[] {null, "edna", "eggplant", "expos"},
   new Object[] {null, "frank", "fruitcake", "phillies"},
   new Object[] {"alice", "albert", "pineapple", "texans"},
   new Object[] {"alice", "agnes", "cranapple", "stars"},
   new Object[] {"bob", "bertha", "banan-o-rama", "packers"},
   new Object[] {"bob", "bradley", "banana split", "badgers"},
   new Object[] {"frank", "felicity", "fruit fly", "eagles"},
   new Object[] {"frank", "fredrick", "fruitful", "flyers"},
   new Object[] {"frank", "fanny", "fruition", "76ers"},
});
ITreeContentProvider tree = new ListTreeContentProvider(list, "#{row[1]}", "#{row[0]}", null);
```

Columns

Just like the "Table" on page 570 control, columns can be added to the Tree control to display content for each tree row. Table columns (including special columns, like Select Row columns) work in Trees exactly the same as they do in Tables. Additionally, special tree-specific "Tree Toggle" on page 580 column controls allow users to toggle tree rows expanded and collapsed. Usually, each tree should include a Tree Toggle control.

Also just like the Table control, columns can be made sortable by specifying a key by which to sort the tree in the column's "sort" property value. See the "Standard Column" on page 568 control.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cont	rol.	true
Value Properties					
value	Data bound to table.	ITreeContent Provider	Tree data model.		
var	Row Variable	String	Name of current row within column	v object (used to referen content).	ce current

Name	Display Name	Туре	Description	Choices	Default	
Expert Properties						
filter	Filter	String	-	A filter expression. Used to filter tables bound to filterable data models.		
filterRequest Param	Filter Request Param	String		Name of request parameter that specifies table's filter expression. Used only for "restful" paging.		
sortRequest Param	Sort Request Param	String		Name of request parameter that specifies table's sort information. Used only for "restful" paging.		
maxOrdinal	Max Ordinal	int		of columns at a time by w		
Expert Display Prop	perties					
bgcolor	Background Color	Color	Table background co	olor.		
border	Border	int	Border width in pixels, for both outer table border and borders between cells.			
cellpadding	Cell Padding	String	Padding within cell content between margin and text.			
cellspacing	Cell Spacing	String	Margin between cell	Margin between cell border and cell content.		
columnClasses	Column CSS Classes	Collection	CSS class to use for erepeating).	CSS class to use for each column, from left to right (not repeating).		
dir	Directionality	String	Text direction hint.	LTR (left-to-right), RTI left)	L (right-to-	
footerClass	Footer CSS Class	String	CSS class to use for table footer.	portlet-section-footer		
frame	Frame	String	Outer border style.	none (no sides), above (top side only), below (bottom side only), hsides (top and bottom sides), vsides (right and left sides), lhs (left side only), rhs (right side only), box (all sides), border (all sides)	none	
headerClass	Header CSS Class	String	CSS class to use for table header.	portlet-section-subhea	der	
lang	Language	String	Language code for c	ontent.		

Name	Display Name	Туре	Description	Choices	Default
rowClasses	Row CSS Classes	Collection	CSS class to use for each column, from top to bottom, repeating.	portlet-section-body, posection-alternate	ortlet-
rowHeight	Row Height	String	Height of each row i (for example 20em).	n pixels (for example 20p	ox) or ems
rules	Rules	String	Grid lines.	none, groups (horizontal lines between header rows and footer rows), rows (horizontal lines only), cols (vertical lines only), all (horizontal and vertical lines)	none
style	CSS Style	String	HTML "style" attribu	ıte value.	
styleClass	CSS Class	String	HTML "class" attribu	ıte value.	
summary	Summary	String	Summary text.		
title	Title	String	HTML "title" attribu	te value.	
width	Width	String	Width of table in pix	xels ('100') or percent ('100	0%').
Expert Client Side E	vents				
onclick	Click	script	Control clicked.		
ondblclick	Double Click	script	Control double-click	sed.	
onkeydown	Key Down	script	Key pressed when co	ontrol is focused.	
onkeypress	Key Press	script	Key pressed and rele	eased when control is foc	rused.
onkeyup	Key Up	script	Key released when o	control is focused.	
onmousedown	Mouse Down	script	Mouse button presse	ed when over control.	
onmousemove	Mouse Move	script	Mouse moved when	over control.	
onmouseout	Mouse Out	script	Mouse moved out of	f control boundaries.	
onmouseover	Mouse Over	script	Mouse moved into c	ontrol boundaries.	
onmouseup	Mouse Up	script	Mouse button releas	ed when over control.	

javax.faces.Column ("Basic Column" on page 547)

Facets

Name	Display Name	Description	Allowed Types
header	Header	Table header.	Any
footer	Footer	Table footer.	Any
empty	Empty Message	Message displayed in place of content when table is empty.	Any.

Tree Toggle

com.webmethods.caf.faces.tree.TreeControl

Base Controls Component Library

A special tree column that allows the user to toggle tree rows expanded and collapsed. It also allows content; the content of the column will be rendered multiple times, one for each row displayed by the tree. The control's header facet automatically incorporates the functionality of the "Column Sort Link" on page 557 control.

The control also optionally specifies the sorting behavior for the column. See the "Standard Column" on page 568 control for further explanation.

Name	Display Name	Туре	Description	Default
General Properties	S			
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties	3			
group	Group	String	Display title of column group (for Add/Remove Columns dialog, etc.). Null or "" is the Default group.	
nowrap	No Wrap	boolean	True if column should be as wide as widest line in the column content.	false
width	Width	String	Width of column in pixels ('100 percent ('100%').)px') or

Name	Display Name	Туре	Description	Default
Sorting				
ascending	Ascending	boolean	Default ascending value. True to sort column from A-Z, 1-10, Jan-Dec, etc.	true
ordinal	Ordinal	int	Default ordinal value. Primacy of this column in sorting the table; 0=none (not sorted on column), 1=primary, 2=secondary, etc.	0
sort	Sort	Object	Key by which to sort column.	
Expert Sorting				
allowToggle	Allow Toggle	boolean	True to allow user to toggle between ascending and descending.	true
showSortLink	Show Sort Link	boolean	True to wrap (display) a header with ColumnSortLink; false to disable ColumnSortLink.	true

Any.

Facets

Name	Display Name	Description	Allowed Types
header	Header	Column header; incorporates Column Sort Link functionality.	Any

Truncating Column

com. we bmethods. caf. faces. table. Truncating Column

Base Controls Component Library

A data table column that automatically limits its content to a single line and to the specified width of the column. Like a standard column, the content of the column will be rendered multiple times, one for each row displayed by the table. See the "Standard Column" on page 568 control for more information about using columns.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control.		true
Display Properties					
align	Alignment	String	Horizontal alignment of content.	left, center, right	
group	Group	String	Display title of column dialog). Null or "" is the	group (for Add/Remove Default group.	Columns
label	Label	String	Display title of column (for Add/Remove Columns dialog). This will not affect the column header. Column header content comes from the "header" facet.		
nowrap	No Wrap	boolean	True if column should b in the column content.	e as wide as widest line	false
optional	Optional	boolean	True if column displaye Add/Remove Columns		false
valign	Vertical Alignment	String	Vertical alignment of content.	top, middle, bottom	
width	Width	String	Width of column in pixe	els ('100px') or percent ('	100%').
Sorting					
ascending	Ascending	boolean	Default ascending value from A-Z, 1-10, Jan-Dec		true
ordinal	Ordinal	int	Default ordinal value. P in sorting the table; 0=ne column), 1=primary, 2=s	one (not sorted on	0
sort	Sort	Object	Key by which to sort co	lumn.	

to toggle between ascending	true

Name	Display Name	Description	Allowed Types
header	Header	Column header; incorporates Column Sort Link functionality.	Any

Chapter 24. Table Row Controls

Add Row Button	586
Add Row Icon	587
Add Row Link	589
Move Row Down Button	590
Move Row Down Icon	591
Move Row Down Link	593
Move Row Up Button	594
Move Row Up Icon	595
Move Row Up Link	596
Remove Row Button	597
Remove Row Icon	599
Remove Row Link	500
Select All Rows Checkbox	502
Select All Visible Rows Checkbox	503
Select Row Checkbox	504
Select Row Column	505
Select Row Link	506
Select Row On Click	507
Select Row Quad-State Checkbox	508
Select Row Tri-State Checkbox	509

Add Row Button

com.webmethods.caf.faces.table.AddRowButton

Base Controls Component Library

A button that adds a row to a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IUpdateableTableContentProvider. The row is not added until the enclosing form is posted to the server, and it passes validation. The behavior of this control is complex; see "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	ID of table to which containing table.	n to add rows. If unspecified, uses	6
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	trol.	true
row	Row	String	Index or ID of row before which to insert the new row, or a special keyword (see choices).	_first (insert before first row), _last (append after last row), _selection (insert before row selection), _this (insert before this row, if button is contained by a row), _next (insert after this row, if button is contained by a row), other (index or ID of row before which to insert)	_last
addLastRowOnly	Add Last Row Only	boolean		for the last row only; in a Table n add a row only if the last row ted.	False
Display Properties					
disabled	Disabled	boolean	True to display as "	grayed-out" and un-editable.	false
label	Label	String	Label text (displaye	ed before contents of children).	

Name	Display Name	Туре	Description	Choices	Default
width	Width	String	Pre-defined button width.	button2 (~2 characters wide), button3, button4, button5, button6, button7, button8, button9, button10, button12, button14, button16, button18, button20 (~20 characters wide)	
Expert Display Prope	rties				
accesskey	Access Key	String	Keyboard shortcut	letter.	
style	CSS Style	String	HTML "style" attril	oute value.	
styleClass	CSS Class	String	HTML "class" attril	oute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Title	String	HTML "title" attrib	ute value (often displayed as a to	oltip).

Any. Children are displayed as button label (after value of "label" property).

Facets

None.

Add Row Icon

com.webmethods.caf.faces.table.AddRowlcon

Base Controls Component Library

An icon that adds a row to a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IUpdateableTableContentProvider. The row is not added until the enclosing form is posted to the server, and it passes validation. The behavior of this control is complex; see "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	ID of table to which containing table.	h to add rows. If unspecified, use	es

Name	Display Name	Туре	Description	Choices	Default
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display cor	ntrol.	true
row	Row	String	Index or ID of row before which to insert the new row, or a special keyword (see choices).	_first (insert before first row), _last (append after last row), _selection (insert before row selection), _this (insert before this row, if link is contained by a row), _next (insert after this row, if link is contained by a row), other (index or ID of row before which to insert)	_last
addLastRowOnly	Add Last Row Only	boolean	True to enable Add for the last row only; in a Table control, the user can add a row only if the last row of the table is selected.		False
Display Properties					
disabled	Disabled	boolean	True to display as '	'grayed-out" and un-editable.	false
url	URL	String	Image icon URL. If used.	unspecified, the default Add Ro	w icon is
urlMousedown	Mousedown URL	String	URL of optional im	nage to display when mouse click	s icon.
urlMouseover	Mouseover URL	String	URL of optional im icon.	nage to display when mouse hove	ers over
Expert Display Proper	rties				
accesskey	Access Key	String	Keyboard shortcut	letter.	
style	CSS Style	String	HTML "style" attril	bute value.	
styleClass	CSS Class	String	HTML "class" attril	bute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		
title	Element Title	String	HTML "title" attrib	ute value (often displayed as a to	oltip.

None.

Facets

Add Row Link

com.webmethods.caf.faces.table.AddRowLink

Base Controls Component Library

A link that adds a row to a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IUpdateableTableContentProvider. The row is not added until the enclosing form is posted to the server, and it passes validation. The behavior of this control is complex; see "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Choices	Default
General Propert	ies				
for	For	String	ID of table to which to table.	add rows. If unspecified, uses co	ntaining
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control		true
row	Row	String	Index or ID of row before which to insert the new row, or a special keyword (see choices).	_first (insert before first row), _last (append after last row), _selection (insert before row selection), _this (insert before this row, if link is contained by a row), _next (insert after this row, if link is contained by a row), other (index or ID of row before which to insert)	_last
addLastRow Only	Add Last Row Only	boolean		the last row only; in a Table dd a row only if the last row of	False
Display Properties					
disabled	Disabled	boolean	True to display as "gra	yed-out" and un-editable.	false
label	Label	String	Label text (displayed b	pefore contents of children).	
Expert Display F	Properties				
accesskey	Access Key	String	Keyboard shortcut lett	er.	
lang	Language	String	Language code for con	tent.	
style	CSS Style	String	HTML "style" attribute	e value.	

Name	Display Name	Туре	Description	Choices	Default
styleClass	CSS Class	String	HTML "class" attribute	e value.	
tabindex	Tab Index	int	Order in tab (key) sequence, negative int		Zero adds item to tab

Any. Children are displayed as link label (after value of "label" property).

Facets

None.

Move Row Down Button

com.webmethods.caf.faces.table.MoveRowDownButton

Base Controls Component Library

A button that moves the selected row(s) down one row in a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IReorderableTableContentProvider. The row is not moved until the enclosing form is posted to the server, and it passes validation. See "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Choices	Default			
General Properties								
for	For	String	ID of table for which to containing table.	ID of table for which to move rows. If unspecified, uses containing table.				
id	ID	String	Control ID.					
rendered	Rendered	boolean	True to display contro	1.	true			
Display Propertion	es							
disabled	Disabled	boolean	True to display as "gra	yed-out" and un-editable.	false			
label	Label	String	Label text (displayed b	pefore contents of children).				
width	Width	String	Pre-defined button width. button2 (~2 characters wide), button3, button4, button5, button6, button7, button8, button9, button10, button12, button14, button16, button18, button20 (~20 characters wide)		on7, tton12,			

Name	Display Name	Туре	Description	Choices	Default		
Expert Display Properties							
accesskey	Access Key	String	Keyboard shortc	ut letter.			
lang	Language	String	Language code f	Language code for content.			
style	CSS Style	String	HTML "style" att	ribute value.			
styleClass	CSS Class	String	HTML "class" att	ribute value.			
tabindex	Tab Index	int		y) sequence, from 1 to n. Zerve integer removes item.	ro adds item to tab		

Any. Children are displayed as button label (after value of "label" property).

Facets

None.

Move Row Down Icon

com.webmethods.caf.faces.table.MoveRowDownIcon

Base Controls Component Library

An icon that moves the selected row(s) down one row in a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IReorderableTableContentProvider. The row is not moved until the enclosing form is posted to the server, and it passes validation. See "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	ID of table for which to move rows. If unspecific containing table.	ed, uses
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description	Default
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" and un-editable.	false
url	URL	String	Image icon URL. If unspecified, the default Move is used.	Row icon
urlMousedown	Mousedown URL	String	URL of optional image to display when mouse cl	icks icon.
urlMouseover	Mouseover URL	String	URL of optional image to display when mouse ho icon.	overs over
Expert Display Prop	perties			
accesskey	Access Key	String	Keyboard shortcut letter.	
lang	Language	String	Language code for content.	
style	CSS Style	String	HTML "style" attribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero add tab sequence, negative integer removes item.	ds item to

None.

Facets

Move Row Down Link

com.webmethods.caf.faces.table.MoveRowDownLink

Base Controls Component Library

A link that moves the selected row(s) down one row in a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IReorderableTableContentProvider. The row is not moved until the enclosing form is posted to the server, and it passes validation. See "Table Row Tools" on page 214 for more information.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	ID of table for which to move rows. If unspecified containing table.	d, uses
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" and un-editable.	false
label	Label	String	Label text (displayed before contents of children)).
Expert Display Prop	perties			
accesskey	Access Key	String	Keyboard shortcut letter.	
lang	Language	String	Language code for content.	
style	CSS Style	String	HTML "style" attribute value.	
styleClass	CSS Class	String	HTML "class" attribute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero ad tab sequence, negative integer removes item.	ds item to

Children

Any. Children are displayed as link label (after value of "label" property).

Facets

Move Row Up Button

com. we bmethods. caf. faces. table. Move Row Up Button

Base Controls Component Library

A button that moves the selected row(s) up one row in a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IReorderableTableContentProvider. The row is not moved until the enclosing form is posted to the server, and it passes validation. See "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Choices	Default	
General Properties						
for	For	String	ID of table for which containing table.	ID of table for which to move rows. If unspecified, uses containing table.		
id	ID	String	Control ID.			
rendered	Rendered	boolean	True to display contr	ol.	true	
Display Properties						
disabled	Disabled	boolean	True to display as "g	rayed-out" and un-editable.	false	
label	Label	String	Label text (displayed	before contents of children).		
width	Width	String	Pre-defined button width. button2 (~2 characters wide), button3, button4, button5, button6, button7, button8, button9, button10, button12, button14, button16, button18, button20 (~20 characters wide)			
Expert Display Prop	perties					
accesskey	Access Key	String	Keyboard shortcut le	etter.		
lang	Language	String	Language code for co	ontent.		
style	CSS Style	String	HTML "style" attribute value.			
styleClass	CSS Class	String	HTML "class" attribute value.			
tabindex	Tab Index	int		quence, from 1 to n. Zero adds ve integer removes item.	item to	

Any. Children are displayed as button label (after value of "label" property).

Facets

None.

Move Row Up Icon

com.webmethods.caf.faces.table.MoveRowUplcon

Base Controls Component Library

Icon that moves the selected row(s) up one row in a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IReorderableTableContentProvider. The row is not moved until the enclosing form is posted to the server, and it passes validation. See "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	ID of table for which to move rows. If unspecified containing table.	l, uses
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" and un-editable.	false
url	URL	String	Image icon URL. If unspecified, the default Move is used.	Row icon
urlMousedown	Mousedown URL	String	URL of optional image to display when mouse cli	icks icon.
urlMouseover	Mouseover URL	String	URL of optional image to display when mouse ho icon.	overs over
Expert Display Prop	erties			
accesskey	Access Key	String	Keyboard shortcut letter.	
lang	Language	String	Language code for content.	
style	CSS Style	String	HTML "style" attribute value.	

Name	Display Name	Туре	Description	Default
styleClass	CSS Class	String	HTML "class" attribute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero at tab sequence, negative integer removes item.	adds item to
	Children			
	None.			
	Facets			
	None.			

Move Row Up Link

com. we bmethods. caf. faces. table. Move Row Up Link

Base Controls Component Library

A link that moves the selected row(s) up one row in a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IReorderableTableContentProvider. The row is not moved until the enclosing form is posted to the server, and it passes validation. See "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Default
General Properties				
for	For	String	ID of table for which to move rows. If unspecified, uses containing table.	
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" and un-editable.	false
label	Label	String	Label text (displayed before contents of children)	•
Expert Display Prop	erties			
accesskey	Access Key	String	Keyboard shortcut letter.	
lang	Language	String	Language code for content.	
style	CSS Style	String	HTML "style" attribute value.	

Name	Display Name	Туре	Description	Default
styleClass	CSS Class	String	HTML "class" attribute value.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero ad tab sequence, negative integer removes item.	lds item to

Any. Children are displayed as link label (after value of "label" property).

Facets

None.

Remove Row Button

com.webmethods.caf.faces.table.RemoveRowButton

Base Controls Component Library

A button that removes a row from a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IUpdateableTableContentProvider. The row is not removed until the enclosing form is posted to the server, and it passes validation. The behavior of this control is complex; see "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	ID of table from wh containing table.	ich to remove rows. If unspecifie	ed, uses
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	trol.	true
row	Row	String	Index or ID of row to remove, or a special keyword (see choices).	_first (remove first row), _last (remove last row), _selection (remove row selection), _this (remove this row, if button is contained by a row), _next (remove next row, if button is contained by a row), other (index or ID of row to remove)	_last

Name	Display Name	Туре	Description	Choices	Default
removeLastRow	Remove Last Row	boolean	is left; in a Table o	Remove even when only one row ontrol, the user can remove a row ly row in the table.	true
Display Properties					
disabled	Disabled	boolean	True to display as	grayed-out" and un-editable.	false
label	Label	String	Label text (displa	yed before contents of children).	
width	Width	String	Pre-defined button width.	button2 (~2 characters wide), button4, button5, button6, button9, button10, bubutton14, button16, button18, b(~20 characters wide)	ton7, tton12,
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shortcu	ıt letter.	
lang	Language	String	Language code fo	or content.	
style	CSS Style	String	HTML "style" att	ribute value.	
styleClass	CSS Class	String	HTML "class" att	ribute value.	
tabindex	Tab Index	int	` 2) sequence, from 1 to n. Zero adds ative integer removes item.	item to

Any. Children are displayed as button label (after value of "label" property).

Facets

Remove Row Icon

com.webmethods.caf.faces.table.RemoveRowlcon

Base Controls Component Library

An icon that removes a row from a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IUpdateableTableContentProvider. The row is not removed until the enclosing form is posted to the server, and it passes validation. The behavior of this control is complex; see "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	ID of table from whi containing table.	ch to remove rows. If unspecif	ied, uses
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display conti	rol.	true
row	Row	String	Index or ID of row to remove, or a special keyword (see choices).	_first (remove first row), _last (remove last row), _selection (remove row selection), _this (remove this row, if button is contained by a row), _next (remove next row, if button is contained by a row), other (index or ID of row to remove)	_last
removeLastRow	Remove Last Row	boolean	is left; in a Table con	nove even when only one row trol, the user can remove a only row in the table.	true
Display Properties					
disabled	Disabled	boolean	True to display as "g	rayed-out" and un-editable.	false
url	URL	String	Image icon URL. If u icon is used.	inspecified, the default Remov	e Row
urlMousedown	Mousedown URL	String	URL of optional ima	ge to display when mouse clic	ks icon.
urlMouseover	Mouseover URL	String	URL of optional ima icon.	ge to display when mouse hov	ers over

Name	Display Name	Туре	Description	Choices	Default
Expert Display Pr	roperties				
accesskey	Access Key	String	Keyboard shorte	cut letter.	
lang	Language	String	Language code f	for content.	
style	CSS Style	String	HTML "style" at	tribute value.	
styleClass	CSS Class	String	HTML "class" at	tribute value.	
tabindex	Tab Index	int		y) sequence, from 1 to n. gative integer removes i	

None.

Facets

None.

Remove Row Link

com.webmethods.caf.faces.table.RemoveRowLink

Base Controls Component Library

A link that removes a row from a table or list. The table or list must be bound to a com.webmethods.caf.faces.data.IUpdateableTableContentProvider. The row is not removed until the enclosing form is posted to the server, and it passes validation. The behavior of this control is complex; see "Table Row Tools" on page 214 for more information.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	String	ID of table from which containing table.	h to remove rows. If unspecified	, uses
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display contro	ıl.	true

Name	Display Name	Туре	Description	Choices	Default
row	Row	String	Index or ID of row to remove, or a special keyword (see choices).	_first (remove first row), _last (remove last row), _selection (remove row selection), _this (remove this row, if button is contained by a row), _next (remove next row, if button is contained by a row), other (index or ID of row to remove)	_last
removeLastRow	Remove Last Row	boolean		ove even when only one row is l, the user can remove a row ow in the table.	true
Display Properties					
disabled	Disabled	boolean	True to display as "gr	ayed-out" and un-editable.	false
label	Label	String	Label text (displayed	before contents of children).	
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shortcut le	tter.	
lang	Language	String	Language code for co	ntent.	
style	CSS Style	String	HTML "style" attribu	te value.	
styleClass	CSS Class	String	HTML "class" attribu	te value.	
tabindex	Tab Index	int	Order in tab (key) sec sequence, negative in	quence, from 1 to n. Zero adds iter teger removes item.	m to tab

Any. Children are displayed as link label (after value of "label" property).

Facets

Select All Rows Checkbox

com.webmethods.caf.faces.table.SelectAllRowsCheckbox

Base Controls Component Library

A checkbox that selects all (or de-selects all) rows in an entire table. For rows to be selectable, the table's content provider must implement the com.webmethods.caf.faces.data.ISelectableTableContentProvider interface. This interface allows rows to be selected either in 'selected' or 'unselected' mode. In 'selected' mode, specified rows are selected explicitly. In 'unselected' mode, all rows are implicitly selected, and only specified rows are unselected.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Expert Display Prop	erties			
accesskey	Access Key	String	Keyboard shortcut letter.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero at tab sequence, negative integer removes item.	dds item to

Children

None.

Facets

Select All Visible Rows Checkbox

com.webmethods.caf.faces.table.SelectAllVisibleRowsCheckbox

Base Controls Component Library

A checkbox that selects all (or de-selects all) rows in the current, visible page of the table. For rows to be selectable, the table's content provider must implement the com.webmethods.caf.faces.data.ISelectableTableContentProvider interface. This interface allows rows to be selected either in 'selected' or 'unselected' mode. In 'selected' mode, specified rows are selected explicitly. In 'unselected' mode, all rows are implicitly selected, and only specified rows are unselected.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Expert Display Prop	erties			
accesskey	Access Key	String	Keyboard shortcut letter.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero act tab sequence, negative integer removes item.	lds item to

Children

None.

Facets

Select Row Checkbox

com.webmethods.caf.faces.table.SelectRowIndividualCheckbox

Base Controls Component Library

A checkbox (or radio-button) which selects the containing row in a table. For rows to be selectable, the table's content provider must implement the com.webmethods.caf.faces.data.ISelectableTableContentProvider interface. This interface allows rows to be selected either in 'selected' or 'unselected' mode. In 'selected' mode, specified rows are selected explicitly. In 'unselected' mode, all rows are implicitly selected, and only specified rows are unselected.

This control displays checkboxes when the 'multiple' property is true, and radio-buttons when the 'multiple' property is false. The checkboxes/radio-buttons can be disabled on a per-row basis by supplying a binding expression for the 'disabled' property.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
multiple	Multiple	boolean	True to allow multiple rows to be selected.	true
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" (use a binding expression to enable/disable on a per-row basis).	false
Expert Display Prop	erties			
accesskey	Access Key	String	Keyboard shortcut letter.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item tab sequence, negative integer removes item.	

Children

None.

Facets

Select Row Column

com.webmethods.caf.faces.table.SelectRowCheckbox

Base Controls Component Library

A column of checkboxes (or radio-buttons), using the "Select Row Checkbox" on page 604 and "Select All Rows Checkbox" on page 602 controls. See "Table Row Tools" on page 214.

Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control id.		
multiple	Multiple	boolean	True to allow multiple rows to b	e selected.	true
rendered	Rendered	boolean	True to display control.	true	
Display Properties					
disabled	Disabled	boolean	True to display as grayed-out (uexpression to enable or disable obasis).		false
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shortcut letter.		
tabindex	Tab Index	int	Order in tab (key) sequence, fro to a tab sequence, a negative int		
valign	Vertical Alignment	String	Vertical alignment of content.	top, middle, bottom	
	Children				
	None				

None.

Facets

Select Row Link

com.webmethods.caf.faces.table.SelectRowLink

Base Controls Component Library

A link that toggles the selection state of the containing row in a table. For rows to be selectable, the table's content provider must implement the com.webmethods.caf.faces.data.ISelectableTableContentProvider interface. This interface allows rows to be selected either in 'selected' or 'unselected' mode. In 'selected' mode, specified rows are selected explicitly. In 'unselected' mode, all rows are implicitly selected, and only specified rows are unselected.

Links can be disabled on a per-row basis by supplying a binding expression for the 'disabled' property.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
multiple	Multiple	boolean	True to allow multiple rows to be selected.	true
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" (use a binding expression to enable/disable on a per-row basis).	false
Value Properties				
label	Label	String	Label text (displayed before contents of children	ı).
Expert Display Prop	perties			
accesskey	Access Key	String	Keyboard shortcut letter.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.	

Children

Any. Children are displayed as link label (after value of "label" property).

Facets

Select Row On Click

com.webmethods.caf.faces.table.SelectRowOnClick

Base Controls Component Library

A control that, when placed inside a table column or list, makes each row in the table or list selectable by clicking the row. For rows to be selectable, the table's content provider must implement the com.webmethods.caf.faces.data.ISelectableTableContentProvider interface. This interface allows rows to be selected either in 'selected' or 'unselected' mode. In 'selected' mode, specified rows are selected explicitly. In 'unselected' mode, all rows are implicitly selected, and only specified rows are unselected.

Rows can be made unselectable on a per-row basis by supplying a binding expression for the 'disabled' property.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
multiple	Multiple	boolean	True to allow multiple rows to be selected.	true
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to make row unselectable (use a binding expression to enable/disable on a per-row basis).	false
Expert Display Prop	erties			
accesskey	Access Key	String	Keyboard shortcut letter.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero act tab sequence, negative integer removes item.	lds item to
	Children			

None

Facets

Select Row Quad-State Checkbox

com.webmethods.caf.faces.tree.SelectRowQuadStateCheckbox

Base Controls Component Library

A checkbox with four states, selecting the containing row in a tree. For rows to be selectable, the tree's content provider must implement the com.webmethods.caf.faces.data.ISelectableTableContentProvider interface. This interface allows rows to be selected either in 'selected' or 'unselected' mode. In 'selected' mode, specified rows are selected explicitly. In 'unselected' mode, all rows are implicitly selected, and only specified rows are unselected. Additionally, the tree's content provider may implement the com.webmethods.caf.faces.data.tree.ISelectableTreeContentProvider interface, which allows the content provider to optimize mixed state calculations.

The four checkbox states are off, on, mixed-off, and mixed-on. Changing the selection state of descendant rows toggles only the mixed state of a row — for example, even when all descendants of a selected row are deselected, the selected row will remain selected. In other words, individual rows can be selected independent of their descendants' selection state. See the following table for a full description of checkbox behavior. Checkboxes can be disabled on a per-row basis by supplying a binding expression for the 'disabled' property.

Initial State		User Action	
	Deselect All Descendants	Deselect Some, Select Some Descendants	Select All Descendants
off	off	off-mixed	off-mixed
on	on-mixed	on-mixed	on
off-mixed	off	off-mixed	off-mixed
on-mixed	on-mixed	on-mixed	on

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Name	Display Name	Туре	Description	Default			
Display Properties							
disabled	Disabled	boolean	True to display as "grayed- out" (use a binding expression to enable/disable on a per-row basis).	false			
Expert Display Prop	erties						
accesskey	Access Key	String	Keyboard shortcut letter.				
tabindex	Tab Index	int	Order in tab (key) sequence, f Zero adds item to tab sequen negative integer removes iter	ce,			

None.

Facets

None.

Select Row Tri-State Checkbox

com.webmethods.caf.faces.tree.SelectRowTriStateCheckbox

Base Controls Component Library

A checkbox with three states, selecting the containing row in a tree. For rows to be selectable, the tree's content provider must implement the com.webmethods.caf.faces.data.ISelectableTableContentProvider interface. This interface allows rows to be selected either in 'selected' or 'unselected' mode. In 'selected' mode, specified rows are selected explicitly. In 'unselected' mode, all rows are implicitly selected, and only specified rows are unselected. Additionally, the tree's content provider may implement the com.webmethods.caf.faces.data.tree.ISelectableTreeContentProvider interface, which allows the content provider to optimize mixed state calculations.

The three checkbox states are off, on, and mixed-off. When a user deselects all descendants of a given row, the row's selection state switches to off; when a user selects some, but not all descendants of a given row, the row's selection state switches to mixed-off; and when a user selects all descendants of a given row, the row's selection state switches to on (see the following table). Checkboxes can be disabled on a per-row basis by supplying a binding expression for the 'disabled' property.

Initial State		User Action		
	Deselect All Descendants	Deselect Some, Select Some Descendants	Select All Descendants	
off	off	off-mixed	on	
on	off	off-mixed	on	
off-mixed	off	off-mixed	on	

Properties

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed- out" (use a binding expression to enable/disable on a per-row basis).	false
Expert Display Prop	erties			
accesskey	Access Key	String	Keyboard shortcut letter.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.	

Children

None.

Facets

Chapter 25. Toggle Controls

Initiate Toggle67	12
One Way Toggle Button	12
One Way Toggle Checkbox	14
One Way Toggle Link	15
Toggle Button	16
Toggle Dropdown	18
Toggle Link	20
Toggle Radio Button Group	21
Toggle Tabs	23

Initiate Toggle

com.webmethods.caf.faces.toggle.InitiateToggle

Base Controls Component Library

A control that executes the toggle behavior for a toggle control, when the toggle control is positioned after one the controls it toggles. See "Toggle Controls" on page 197 for more information.

Properties

Name	Display Name	Туре	Description	Default
General Properties				
for	For	Control ID Reference	ID of toggle control.	
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true

Children

None.

Facets

None.

One Way Toggle Button

com. we bmethods. caf. faces. toggle. One Way Toggle Button

Base Controls Component Library

A button that toggles the client-side visibility of a single "hideable" control, or the server-side "rendered" property of a single non-hideable control. See "Hideable Controls" on page 194 for an explanation of "hideable" controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	Control ID Reference		e. (May also reference a h portlet, or a raw HTML o	

Name	Display Name	Туре	Description	Choices	Default
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control.		true
way	Way	String	Way to toggle.	toggle (toggle between visible and hidden), show (toggle visible only), hide (toggle hidden only)	toggle
Display Properties					
disabled	Disabled	boolean	True to display as "grayed-out" and un- editable.		false
width	Width	String	Pre-defined button width. button2 (~2 characters wide), button3, button4, button5, button6, button7, button8, button9, button10, button12, button14, button16, button18, button20 (~20 characters wide)		on5, on8, ton12, tton18,
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instance to u and from a String.	se to convert the contro	l's value to
value	Value	Object	Button label.		
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shortcut lette	er.	
tabindex	Tab Index	int	Order in tab (key) sequence, negative	ence, from 1 to n. Zero a integer removes item.	dds item to

Any. Children are displayed as button label (after the content of the "value" property).

Facets

One Way Toggle Checkbox

com.webmethods.caf.faces.toggle.OneWayToggleCheckbox

Base Controls Component Library

A checkbox that toggles the client-side visibility of a single "hideable" control, or the server-side "rendered" property of a single non-hideable control. See "Hideable Controls" on page 194 for an explanation of "hideable" controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
checkedWhen Visible	Checked When Visible	boolean	True if checked who false if checked targ		true
for	For	Control ID Reference		gle. (May also referen an external portlet, or	
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	trol.	true
Display Properties					
disabled	Disabled	boolean	True to display as "geditable.	grayed-out" and un-	false
width	Width	String	Pre-defined input width.	input2 (~2 character input4, input6, inpu input10, input20, in input40, input50 (~5 characters wide)	ıt8, put30,
Value Properties					
converter	Converter	javax.faces. convert.Converter	Converter instance value to and from a	to use to convert the c String.	control's
value	Value	Object	Checkbox label.		
Expert Display Prop	erties				
accesskey	Access Key	String	Keyboard shortcut	letter.	
tabindex	Tab Index	int	Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.		

Any. Children are displayed as checkbox label (after the content of the "value" property).

Facets

None.

One Way Toggle Link

com.webmethods.caf.faces.toggle.OneWayToggleLink

Base Controls Component Library

A link that toggles the client-side visibility of a single "hideable" control, or the server-side "rendered" property of a single non-hideable control. See "Hideable Controls" on page 194 for an explanation of "hideable" controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
for	For	Control ID Reference	ID of control to toggle. hideable control in an e HTML element.)		
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control		true
way	Way	String	Way to toggle.	toggle (toggle between visible and hidden), show (toggle visible only), hide (toggle hidden only)	toggle
Display Properties					
disabled	Disabled	boolean	True to display as "grayeditable.	yed-out" and un-	false
Value Properties					
converter	Converter	javax.faces. convert.Converter	Converter instance to use to convert the control's value to and from a String.		
value	Value	Object	Link label.		

Name	Display Name	Туре	Description	Choices	Default			
Expert Display Properties								
accesskey	Access Key	String	Keyboard shortc	Keyboard shortcut letter.				
tabindex	Tab Index	int		Order in tab (key) sequence, from 1 to n. Zero adds item to tab sequence, negative integer removes item.				

Any. Children are displayed as link label (after the content of the "value" property).

Facets

None.

Toggle Button

com.webmethods.caf.faces.toggle.ToggleButton

Base Controls Component Library

A button that toggles the client-side visibility of a group of "hideable" controls, or the server-side "rendered" property of a group of non-hideable controls. Only the selected control from this group (specified by the "value" property of this control) is visible. Selection options are specified via javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children, with each value specifying a control ID. Selecting an option whose value does not specify a control ID hides all controls in the group.

For example, if the toggle control has three options, "control-one", "control-two", and "control-three", and the toggle control's "value" property is empty, then "control-one", "control-two", and "control-three" will all be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible, and "control-two" and "control-three" will be hidden.

This control can also be used to toggle a single control between visible and hidden. For example, if the toggle control has two options, "control-one" and "", and the toggle control's "value" property is empty, then "control-one" will be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible.

See "Toggle Controls" on page 197 for further explanation of "toggle" controls. See "Hideable Controls" on page 194 for an explanation of "hideable" controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control		true
Display Properties					
style	CSS Style	String	HTML "style" attribute	value.	
styleClass	CSS Class	String	HTML "class" attribute	value.	
disabled	Disabled	boolean	True to display as "graed editable.	yed-out" and un-	false
width	Width	String	Pre-defined button width.	button2 (~2 characters button3, button4, butto button6, button7, butto button9, button10, but button14, button16, but button20 (~20 character	on5, on8, ton12, itton18,
Value Properties					
converter	Converter	javax.faces. convert. Converter	Converter instance to uto and from a String.	ise to convert the contro	l's value
immediate	Immediate	boolean	True to process ValueC Apply Request Values in the Process Validation	phase; false to process	false
other	Other	boolean	True to allow users to e the specified values.	enter values other than	false
required	Required	boolean	True to display control	as "required".	false
validator	Validator	javax.faces. validator. Validator	Validator instance to u submitted value.	se to validate the control	l's
value	Value	Object	Current selection value	2.	
valueChange Listener	Value Change Listener	javax.faces.el. MethodBinding	Value change callback.		

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Toggle Dropdown

com.webmethods.caf.faces.toggle.ToggleMenu

Base Controls Component Library

A dropdown control that toggles the client-side visibility of a group of "hideable" controls, or the server-side "rendered" property of a group of non-hideable controls. Only the selected control from this group (specified by the "value" property of this control) is visible. Selection options are specified via javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children, with each value specifying a control ID. Selecting an option whose value does not specify a control ID hides all controls in the group.

For example, if the toggle control has three options, "control-one", "control-two", and "control-three", and the toggle control's "value" property is empty, then "control-one", "control-two", and "control-three" will all be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible, and "control-two" and "control-three" will be hidden.

This control can also be used to toggle a single control between visible and hidden. For example, if the toggle control has two options, "control-one" and "", and the toggle control's "value" property is empty, then "control-one" will be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible.

See "Toggle Controls" on page 197 for further explanation of "toggle" controls. See "Hideable Controls" on page 194 for an explanation of "hideable" controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display co	ontrol.	true

Name	Display Name	Туре	Description	Choices	Default
Display Properties					
disabled	Disabled	boolean	True to display as "gr editable.	ayed-out" and un-	false
width	Width	String	Pre-defined input width.	input2 (~2 characte input4, input6, inp input10, input20, i input40, input50 (~ characters wide)	ut8, nput30,
Value Properties					
converter	Converter	javax.faces. convert.Converter	Converter instance to use to convert the control's value to and from a String.		
immediate	Immediate	boolean	True to process ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		
other	Other	boolean	True to allow users to than the specified val		false
required	Required	boolean	True to display contro	ol as "required".	false
validator	Validator	javax.faces. validator. Validator	Validator instance to submitted value.	use to validate the co	ontrol's
value	Value	Object	Current selection val	ue.	
valueChange Listener	Value Change Listener	javax.faces.el. MethodBinding	Value change callback	k.	

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

Toggle Link

com.webmethods.caf.faces.toggle.ToggleLink

Base Controls Component Library

A link that toggles the client-side visibility of a group of "hideable" controls, or the server-side "rendered" property of a group of non-hideable controls. Only the selected control from this group (specified by the "value" property of this control) is visible. Selection options are specified via javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children, with each value specifying a control ID. Selecting an option whose value does not specify a control ID hides all controls in the group.

For example, if the toggle control has three options, "control-one", "control-two", and "control-three", and the toggle control's "value" property is empty, then "control-one", "control-two", and "control-three" will all be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible, and "control-two" and "control-three" will be hidden.

This control can also be used to toggle a single control between visible and hidden. For example, if the toggle control has two options, "control-one" and "", and the toggle control's "value" property is empty, then "control-one" will be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible.

See "Toggle Controls" on page 197 for further explanation of "toggle" controls. See "Hideable Controls" on page 194 for an explanation of "hideable" controls.

Name	Display Name	Туре	Description	Default
General Properties				
id	ID	String	Control ID.	
rendered	Rendered	boolean	True to display control.	true
Display Properties				
disabled	Disabled	boolean	True to display as "grayed-out" and un- editable.	false
Value Properties				
converter	Converter	javax.faces. convert.Conver ter	Converter instance to use to convert the control's value to and from a String.	

Name	Display Name	Туре	Description	Default
immediate	Immediate	boolean	True to process ValueChangeListener s in the Apply Request Values phase; false to process in the Process Validations phase.	false
other	Other	boolean	True to allow users to enter values other than the specified values.	false
required	Required	boolean	True to display control as "required".	false
validator	Validator	javax.faces. validator. Validator	Validator instance to use to validate the control's submitted value.	
value	Value	Object	Current selection value.	
valueChange Listener	Value Change Listener	javax.faces.el. MethodBinding	Value change callback.	

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Toggle Radio Button Group

com.webmethods.caf.faces.toggle.ToggleRadio

Base Controls Component Library

A radio button group that toggles the client-side visibility of a group of "hideable" controls, or the server-side "rendered" property of a group of non-hideable controls. Only the selected control from this group (specified by the "value" property of this control) is visible. Selection options are specified via javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children, with each value specifying a control ID. Selecting an option whose value does not specify a control ID hides all controls in the group.

For example, if the toggle control has three options, "control-one", "control-two", and "control-three", and the toggle control's "value" property is empty, then "control-one", "control-two", and "control-three" will all be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible, and "control-two" and "control-three" will be hidden.

This control can also be used to toggle a single control between visible and hidden. For example, if the toggle control has two options, "control-one" and "", and the toggle control's "value" property is empty, then "control-one" will be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible.

See "Toggle Controls" on page 197 for further explanation of "toggle" controls. See "Hideable Controls" on page 194 for an explanation of "hideable" controls.

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display control		true
Display Properties					
disabled	Disabled	boolean	True to display as "grayed-out" and uneditable.		false
width	Width	String	Pre-defined input width. input2 (~2 character input4, input6, inp input10, input20, i input40, input50 (characters wide)		ut8, nput30,
Value Properties					
converter	Converter	javax.faces. convert.Converter	Converter instance to uvalue to and from a Str		ntrol's
immediate	Immediate	boolean	True to process ValueC the Apply Request Val process in the Process	ues phase; false to	false
other	Other	boolean	True to allow users to enter values other than the specified values.		false
required	Required	boolean	True to display control	as "required".	false
validator	Validator	javax.faces. validator. Validator	Validator instance to u submitted value.	se to validate the cor	itrol's

Name	Display Name	Туре	Description	Choices	Default
value	Value	Object	Current selection value	2.	
valueChange Listener	Value Change Listener	javax.faces.el. MethodBinding	Value change callback.		

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None.

Toggle Tabs

com.webmethods.caf.faces.toggle.ToggleTabs

Base Controls Component Library

Tabs that toggle the client-side visibility of a group of "hideable" controls, or the server-side "rendered" property of a group of non-hideable controls. Only the selected control from this group (specified by the "value" property of this control) is visible. Selection options are specified via javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326) children, with each value specifying a control ID. Selecting an option whose value does not specify a control ID hides all controls in the group.

For example, if the toggle control has three options, "control-one", "control-two", and "control-three", and the toggle control's "value" property is empty, then "control-one", "control-two", and "control-three" will all be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible, and "control-two" and "control-three" will be hidden.

This control can also be used to toggle a single control between visible and hidden. For example, if the toggle control has two options, "control-one" and "", and the toggle control's "value" property is empty, then "control-one" will be hidden. If the toggle's control "value" property is "control-one", then "control-one" will be visible.

See "Toggle Controls" on page 197 for further explanation of "toggle" controls. See "Hideable Controls" on page 194 for an explanation of "hideable" controls.

Properties

Name	Display Name	Туре	Description	Choices	Default
General Properties					
id	ID	String	Control ID.		
rendered	Rendered	boolean	True to display con	trol.	true
Display Properties					
disabled	Disabled	boolean	True to display as "geditable.	grayed-out" and un-	false
layout	Layout	String	Tabs layout.	top, left, bottom, right	top
Value Properties					
converter	Converter	javax.faces. convert.Converter	Converter instance to use to convert the control's value to and from a String.		
immediate	Immediate	boolean	True to process ValueChangeListeners in the Apply Request Values phase; false to process in the Process Validations phase.		false
other	Other	boolean	True to allow users than the specified v	to enter values other ralues.	false
required	Required	boolean	True to display con	trol as "required".	false
validator	Validator	javax.faces. validator. Validator	Validator instance t submitted value.	o use to validate the con	trol's
value	Value	Object	Current selection va	alue.	
valueChange Listener	Value Change Listener	javax.faces.el. MethodBinding	Value change callba	nck.	

javax.faces.SelectItem ("Option" on page 324) and javax.faces.SelectItems ("Option Group" on page 326)

Facets

None

Chapter 26. Webapp Controls

Breadcrumbs	626
Popup Menus	627
Static Menus	629
Toggle Menus	631

Breadcrumbs

com.webmethods.caf.faces.nav.Breadcrumbs

Base Controls Component Library

A control that displays Web application breadcrumbs navigation as a list of links.

Properties

Name	Display Name	Туре	Description	Default		
General Properti	es					
id	ID	String	Control id.			
rendered	Rendered	boolean	True to display control.	true		
Expert Display P	Expert Display Properties					
customRow Class	Custom Row CSS Class	String	Binding expression that specifies a custom CSS class name to add to the current row.			
rowClasses	Row CSS Classes	Collection	CSS class to use for each row, from top to bottom, repeating.			
selectedRow Class	Selected Row CSS Class	String	CSS class to use for the selected row.			
style	CSS Style	String	HTML "style" attribute value.			
styleClass	CSS Class	String	CSS class to use for the list.			
title	Title	String	HTML "title" attribute value.			
Expert Value Pro	Expert Value Properties					
appNavBean	App Nav Bean	BaseAppNav Bean	The AppNavBean object to use as the data-model for this control. Defaults to the current page bean's AppNavBean.			

Children

None.

Facets

Popup Menus

com.webmethods.caf.faces.nav.PopupMenus

Base Controls Component Library

A control that displays Web application top-level navigation as a list of links. Clicking on or hovering over one of the top-level links displays the children of that page as a popupmenu of second-level links.

Name	Display Name	Туре	Description	Choices	Default		
General Propertie	es						
id	ID	String	Control id.				
rendered	Rendered	boolean	True to display control		true		
Display Propertie	Display Properties						
emptyMsg	emptyMsg Empty String Message to display on the second level if the selected top-level page has no children.						
hover	Hover	boolean	the mouse hovers over to toggle the second-le	e to toggle the second-level display when false mouse hovers over a top-level item; false oggle the second-level display only if the use clicks a top-level item.			
linkToplevel	Link Top- Level	boolean	True to display links to false to display their na		false		
marker	Marker	String	String to place next to (for "vertical" or "horizontal" types) or between (for "horizontal-separated" type) items. Does not apply to "ol" or "ul" types.	decimal (decimal row other (any raw HTM			
layout	Layout	String	Layout style.	vertical, horizontal (marker aligned next to item), horizontal- separated (marker centered between items)	vertical		

Name	Display Name	Туре	Description	Choices	Default
Expert Display Pro	perties				
customRow Class	Custom Row CSS Class	String	Binding expression that name to add to the cur	•	SS class
rowClasses	Row CSS Classes	Collection	CSS class to use for the top-level rows, from top to bottom, repeating.		
selectedRow Class	Selected Row CSS Class	String	CSS class to use for the selected top-level row.		
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	CSS class to use for the top-level list.		
title	Title	String	HTML "title" attribute value.		
sublistClass	Sublist CSS Class	String	CSS class to use for the second-level list.		
sublistow Classes	Sublist Row CSS Classes	Collection	CSS class to use for the bottom, repeating.	e second-level rows, fro	om top to
sublistSelected RowClass	Sublist Selected Row CSS Class	String	CSS class to use for the selected second-level row.		
Expert Value Prop	erties				
appNavBean	App Nav Bean	BaseAppNav Bean	The AppNavBean obje control. Defaults to the AppNavBean.		odel for this

None.

Facets

Static Menus

com.webmethods.caf.faces.nav.StaticMenus

Base Controls Component Library

A control that displays Web application top-level navigation (and optionally second- and third-level pages) as a list of links. If this control's "depth" property is set to 1, only the top-level links are displayed. If this control's "depth" property is set to 2, the top-level links are displayed, and if the current page is one of the top-level pages or a descendent of one of the top-level pages, the children of that top-level page are displayed as the second-level links. If this control's "depth" property is set to 3, the top-level and second-level links are displayed the same way as a depth of 2, and if the current page is one of the second-level pages or a descendent of one of the second-level pages, the children of that second-level page are displayed as the third-level links.

Name	Display Name	Туре	Description	Choices	Default
General Proper	rties				
depth	Depth	int	Depth of navigation lists.	1: Top-level, 2: Top-level and second-level, 3: Top-level, second-level, and third-level	1
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display control.		true
Display Proper	ties				
marker	Marker	String	String to place next to (for "vertical" or "horizontal" types) or between (for "horizontal-separated" type) items. Does not apply to "ol" or "ul" types.	decimal (decimal row other (any raw HTM	
layout	Layout	String	Layout style.	vertical, horizontal (marker aligned next to item), horizontal- separated (marker centered between items)	vertical

Name	Display Name	Туре	Description	Choices	Default	
Expert Display Pro	perties					
customRow Class	Custom Row CSS Class	String		Binding expression that specifies a custom CSS class name to add to the current row.		
rowClasses	Row CSS Classes	Collection	CSS class to use for th bottom, repeating.	CSS class to use for the top-level rows, from top to bottom, repeating.		
selectedRow Class	Selected Row CSS Class	String	CSS class to use for th	e selected top-level ro	OW.	
style	CSS Style	String	HTML "style" attribut	e value.		
styleClass	CSS Class	String	CSS class to use for th	e top-level list.		
title	Title	String	HTML "title" attribute	value.		
sublistClass	Sublist CSS Class	String	CSS class to use for the second-level list.			
sublistow Classes	Sublist Row CSS Classes	Collection	CSS class to use for the second-level rows, from top to bottom, repeating.			
sublistSelected RowClass	Sublist Selected Row CSS Class	String	CSS class to use for the selected second-level row.			
subsublist Class	Sub-Sublist CSS Class	String	CSS class to use for th	e second-level list.		
subsublistow Classes	Sub-Sublist Row CSS Classes	Collection	CSS class to use for th bottom, repeating.	e second-level rows,	from top to	
subsublist SelectedRow Class	Sub-Sublist Selected Row CSS Class	String	CSS class to use for th	e selected second-lev	el row.	
Expert Value Prope	erties					
appNavBean	App Nav Bean	BaseAppNav Bean	The AppNavBean objecthis control. Defaults AppNavBean.			

None.

Facets

Toggle Menus

com. we bmethods. caf. faces. nav. Toggle Menus

Base Controls Component Library

A control that displays Web application top-level navigation as a list of links. Clicking or hovering over one of the top-level links displays the children of that page as a list of second-level links.

Name	Display Name	Туре	Description	Choices	Default
General Propertie	S				
id	ID	String	Control id.		
rendered	Rendered	boolean	True to display control		true
Display Properties	S				
emptyMsg	Empty Message	String	Message to display in top-level page has no c	the second-level if the sell children.	lected
hover	Hover	boolean	True to toggle the second the mouse hovers over to toggle the second-le mouse clicks a top-lever	a top-level item; false vel display only if the	false
linkToplevel	Link Top-Level	boolean	True to display links to false to display their na		false
marker	Marker	String	String to place next to (for "vertical" or "horizontal" types) or between (for "horizontal-separated" type) each item. Does not apply to "ol" or "ul" types.	decimal (decimal row number), other (any raw HTML string)	
Expert Display Pr	operties				
customRow Class	Custom Row CSS Class	String	Binding expression that name to add to the cur	nt specifies a custom CSS rent row.	class
rowClasses	Row CSS Classes	Collection	CSS class to use for the top-level rows, from top to bottom, repeating.		
selectedRow Class	Selected Row CSS Class	String	CSS class to use for the	selected top-level row.	

Name	Display Name	Туре	Description	Choices	Default
style	CSS Style	String	HTML "style" attribute value.		
styleClass	CSS Class	String	CSS class to use for the top-level list.		
title	Title	String	HTML "title" attribute value.		
sublistClass	Sublist CSS Class	String	CSS class to use for the second-level list.		
sublistow Classes	Sublist Row CSS Classes	Collection	CSS class to use for the second-level rows, from top to bottom, repeating.		m top to
sublistSelected RowClass	Sublist Selected Row CSS Class	String	CSS class to use for the selected second-level row.		row.
Expert Value Prop	erties				
appNavBean	App Nav Bean	BaseAppNav Bean	The AppNavBean object to use as the data-model for this control. Defaults to the current page bean's AppNavBean.		

None.

Facets

Part III. Converters Reference

Converter Concepts
Converters
Currency Converters
Formatted Converters

webMethods Designer - Composite Application Framework Help Version 7.1.1

Chapter 27. Converter Concepts

Using Converters	. 630
Creating a Custom Converter	. 630

Using Converters

Converters provide a way to format objects as strings so they can be displayed properly. You can apply converters to either input or output controls. The Converter node of the Palette view contains both standard JSF converters and those developed by Software AG. Some converters, such as the "Date and Time Converter" on page 643, have properties that you can modify on the Properties view.



To add a converter to the design canvas

- 1 In design canvas, locate the input or output control that needs to have a converter.
- 2 On the Palette view expand the Converter node to find the converter you need.
- 3 Drag the converter until you can drop it on the control.
 - You can only drop onto a control whose boundary is highlighted when you drag the converter over it. If the **o** overlay is visible on the cursor, you cannot drop the converter.
- 4 In the design canvas, click the converter.
 If the converter has any editable properties, they are visible on the Properties view.

For descriptions of properties in a specific converter, see the reference page for that converter.

Creating a Custom Converter

If you need a converter that is not available from among those provided in the Palette view and are familiar with writing Java code, you can create your own converter. Designer provides some initial code to get you started. A custom converter is valid only for the portlet in which it is created. If you plan to use a particular converter in many portlets, you should create your own library and load it through the project.



To create a custom converter

- 1 In the design canvas, select the input or output control for which you need to create a custom converter.
- 2 In the General tab of the Properties view, type a unique ID in the ID field and click away from the view.
- In the design canvas, right-click the control and then click Event Handlers > Convert Handler.

Designer creates some Java code in the managed bean and opens a Java editor to the location of the code. An example of converter code created for an input control is shown here:

```
public javax.faces.convert.Converter getInputID_converter()
  return new javax.faces.convert.Converter()
      /**
      * Convert the input Object into a string.
      public String getAsString( javax.faces.context.FacesContext context,
javax.faces.component.UIComponent component, java.lang.Object value)
         // TODO: Convert the Object to a String here.
        return value.toString();
      /**
      * Convert the input String into an Object.
     public Object getAsObject( javax.faces.context.FacesContext context,
javax.faces.component.UIComponent component, java.lang.String value)
         // TODO: Convert the String to an Object here.
        return "New Object: "+ value;
   };
}
```

After the TODO comments, place the Java code needed to make the needed conversions.

Chapter 28. Converters

Array Converter
Big Decimal Converter
Big Integer Converter
Boolean Converter
Byte Converter
Character Converter
Collection Converter
Date and Time Converter
Double Converter
Float Converter
Integer Converter
Iterator Converter
Long Converter
Map Converter
Number Converter
Short Converter
String Array Converter
WmPortal IURI Converter649

Array Converter

com. we bmethods. caf. faces. convert. Array Converter

A converter for an array of generic objects.

Properties

Name	Display Name	Туре	Description	Choices	Default
Converter Properties					
convertType	Convert To Type	String	Converts a CSV string into an Array of the selected type. The elements in the string must be parsable by the selected type and separated by commas (CSV).	boolean, byte, double, float, int, long, short, String	String
customErrorMsg	Custom Error Message	String	A custom message for ir conversion errors.	nvalid object types and	d
showCAFErrorMsg	Append CAF Error Message	boolean	Include the CAF error m custom error message.	nessage with the	false
Children					
N	one.				
Fa	cets				

Big Decimal Converter

javax.faces.convert.BigDecimalConverter

A converter implementation for java.math.BigDecimal values. A standard JSF converter.

JSF Properties

None.

None.

Children

None.

Facets

Big Integer Converter

javax.faces.convert.BigIntegerConverter

A converter implementation for java.math.BigInteger values. A standard JSF converter.

JSF Properties

None.

Children

None.

Facets

None.

Boolean Converter

javax.faces.convert.BooleanConverter

A converter implementation for java.lang.Boolean (and boolean primitive) values. A standard JSF converter.

JSF Properties

None.

Children

None.

Facets

None.

Byte Converter

javax.faces.convert.ByteConverter

Converter implementation for java.lang.Byte (and byte primitive) values. A standard JSF converter.

JSF Properties

None.

Children

None.

Facets

Character Converter

javax.faces.convert.CharacterConverter

Converter implementation for java.lang.Character (and char primitive) values. A standard JSF converter.

JSF Properties

None.

Children

None.

Facets

None.

Collection Converter

com.webmethods.caf.faces.convert.CollectionConverter

A converter for java.util.Collection.

Converter Properties convertType	Convert To Type				
convertType	Convert To Type				
	Convert To Type	String	Converts a CSV string into a collection of the selected type. The elements in the string must be parsable by the selected type and separated by commas (CSV).	boolean, byte, double, float, int, long, short, String	String
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.		
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error custom error message.	message with the	false

None.

Facets

None.

Date and Time Converter

javax.faces.convert.DateTimeConverter

A converter implementation for java.util.Date values. If a pattern has been specified, its syntax must conform to the rules specified by the java.text.SimpleDateFormat. Such a pattern is used to parse, and the type, dateStyle, and timeStyle properties are ignored. A standard JSF converter.

JSF Properties

Name	Display Name	Туре	Description	Choices	Default	
Converter Properties						
dateStyle	Date Style	String	The style to be used to format or parse dates.	default, full, long, medium, short	default	
locale	Locale	String	The Locale to be used when parsing or formatting dates and times.			
timeStyle	Time Style	String	The style to be used to format or parse times.	default, full, long, medium, short	default	
timeZone	Time Zone	String	The time zone used to interpret the time value.			
pattern	Pattern	String	The format pattern to be used when formatting and parsing dates and times.			
type	Туре	String	The type of value to be formatted or parsed.	both, date, time	date	

	Children
	None.
	Facets
	None.
Double Co	nverter
	javax.faces.convert.DoubleConverter
	A converter implementation for java.lang.Double (and double primitive) values. A standard JSF converter.
	Children
	None.
	Facets
	None.
Float Conv	verter
	javax.faces.convert.FloatConverter
	A converter implementation for java.lang.Float (and float primitive) values. A standard JSF converter.
	Children
	None.
	Facets

Integer Converter

javax.faces.convert.IntegerConverter

A converter implementation for java.lang.Integer (and int primitive) values. A standard JSF converter.

Children

None.

Facets

None.

Iterator Converter

com.webmethods.caf.faces.convert.IteratorConverter

A converter for java.util.Iterator.

Name	Display Name	Туре	Description	Choices	Default
Converter Properties					
convertType	Convert To Type	String	Converts a CSV string into an iterator containing the selected type. The elements in the string must be parsable by the selected type and separated by commas (CSV).	boolean, byte, double, float, int, long, short, String	String
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.		
showCAFErrorMsg	Append CAF Error Message	boolean	Include the CAF error custom error message.	message with the	false

None.

Facets

None.

Long Converter

javax.faces.convert.LongConverter

A converter implementation for java.lang.Long (and long primitive) values. A standard JSF converter.

Children

None.

Facets

None.

Map Converter

com. we bmethods. caf. faces. convert. Map Converter

A converter for java.util.Map. Converts a Map to a CSV string of key, value pairs.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.	
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	false

Children

None.

Facets

Number Converter

javax.faces.convert.NumberConverter

A converter implementation for java.lang.Number values. If a pattern has been specified, its syntax must conform to the rules specified by the Java class java.text.DecimalFormat. Such a pattern is used to parse, and the type property is ignored.

JSF Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
currencyCode	Currency Code	String	The three-letter ISO 4217 currency code used to define the type of currency. If not set, the value used is based on the formatting Locale.	
currencySymbol	Currency Symbol	String	The currency symbol to use.	
groupingUsed	Grouping Enforced	String	Include grouping separators if necessary.	true
integerOnly	Display as an Integer	String	Only the integer portion of the given value should be returned from the converter's method: getAsObject.	false
locale	Locale	String	The Locale to be used when parsing numbers.	
maxFractionDigits	Fraction Maximum Length	String	The maximum number of digits the converter method getAsString should render in the fraction portion of the result.	
maxIntegerDigits	Integer Maximum Length	String	The maximum number of digits the converter method getAsString should render in the integer portion of the result.	
minFractionDigits	Fraction Minimum Length	String	The minimum number of digits the converter method getAsString should render in the fraction portion of the result.	
minIntegerDigits	Integer Minimum Length	String	The minimum number of digits the converter method getAsString should render in the integer portion of the result.	

Name	Display Name	Туре	Description	Default
pattern	Pattern	String	The format pattern to be used when formatting and parsing numbers. Valid values are those supported by java.text.DecimalFormat. An invalid value cause a ConverterException.	
type	Number Type	String	The number type to be used when formatting and parsing numbers. An invalid value causes a ConverterException.	number

None.

Facets

None.

Short Converter

javax.faces.convert.ShortConverter

A converter implementation for java.lang.Short (and short primitive) values. A standard JSF converter.

Children

None.

Facets

String Array Converter

com. we bmethods. caf. faces. convert. String Array Converter

A converter for a Java String array. Converts a String Array to a CSV string and back.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
customErrorMsg	Custom Error Message	String	A custom message fo object types and converrors.	
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	false

Children

None.

Facets

None.

WmPortal IURI Converter

com.webmethods.caf.faces.converter.PortalURI

A converter that can be used for a text input field where the user is typing in a My webMethods Server thingID or alias. When the form is posted, the supplied string is automatically converted into the equivalent My webMethods Server IURI object.



Note: This converter is not portable to third-party portals. You can use it only in portlets published to instances of My webMethods Server.

A IURI is the interface used to represent an object that references a particular My webMethods Server resource. All objects in the system can be looked up my means of an IURI. For more information on IURIs, see the JavaDoc in the My webMethods Server installation at:

webMethods_install_dir/MWS/doc/javaapi/wm-caf-server/com/webmethods/portal/system/IURI.html

Properties	
None.	
Children	
None.	

Facets
None.

Chapter 29. Currency Converters

Currency Big Decimal Converter	652
Currency Big Integer Converter	653
Currency Double Converter	654
Currency Float Converter	655
Currency Integer Converter	656
Currency Long Converter	657
Currency Number Converter	658
Currency Short Converter	659

Currency Big Decimal Converter

com. we bmethods. caf. faces. convert. Currency Big Decimal Converter

A converter implementation for java.math.BigDecimal values, taking into account localized currency formatting.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.	
region	Locale	String	The Locale to use for d the value.	letermining
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.	
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	false
currencyCode	Currency Code	String	The formatting for a currency. Use an ISO 4217 three-letter code.	

Children

None.

Facets

Currency Big Integer Converter

com. we bmethods. caf. faces. convert. Currency BigInteger Converter

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.	
region	Locale	String	The Locale to use for detection the value.	termining
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.	
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	false
currencyCode	Currency Code	String	The formatting for a currency. Use an ISO 4217 three-letter code.	

Children

None.

Facets

Currency Double Converter

com. we bmethods. caf. faces. convert. Currency Double Converter

A converter implementation for java.lang.Double (and double primitive) values, taking into account localized currency formatting.

Properties

Name	Display Name	Туре	Description Default	
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.	
region	Locale	String	The Locale to use for determining the value.	
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.	
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF false error message with the custom error message.	
currencyCode	Currency Code	String	The formatting for a currency. Use an ISO 4217 three-letter code.	

Children

None.

Facets

Currency Float Converter

com. we bmethods. caf. faces. convert. Currency Float Converter

A converter implementation for java.lang.Float (and float primitive) values, taking into account localized currency formatting.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.	
region	Locale	String	The Locale to use for d the value.	etermining
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.	
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	false
currencyCode	Currency Code	String	The formatting for a cu Use an ISO 4217 three-	•

Children

None.

Facets

Currency Integer Converter

com. we bmethods. caf. faces. convert. Currency Integer Converter

A converter implementation for java.lang.Integer (and int primitive) values, taking into account localized currency formatting.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.	
region	Locale	String	The Locale to use for determining the value	•
customErrorMsg	Custom Error Message	String	A custom message for invalid object types a conversion errors.	nd
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	false
currencyCode	Currency Code	String	The formatting for a currency. Use an ISO4 2 letter code.	217 three-

Children

None.

Facets

Currency Long Converter

com. we bmethods. caf. faces. convert. Currency Long Converter

A converter implementation for java.lang.Long (and long primitive) values, taking into account localized currency formatting.

Properties

Name	Display Name	Туре	Description Default	t
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.	
region	Locale	String	The Locale to use for determining the value.	
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.	
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	
currencyCode	Currency Code	String	The formatting for a currency. Use an ISO 4217 thre letter code.	ee-

Children

None.

Facets

Currency Number Converter

com. we bmethods. caf. faces. convert. Currency Number Converter

A converter implementation for java.lang.Number values, taking into account localized currency formatting.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conform syntax rules of the Java class java.text.DecimalFormat.	s to the
region	Locale	String	The Locale to use for determining the valu	ıe.
customErrorMsg	Custom Error Message	String	A custom message for invalid object types conversion errors.	and
showCAFErrorMsg	Append CAF Error Message	boolean	Include the CAF error message with the custom error message.	false
currencyCode	Currency Code	String	The formatting for a currency. Use an ISO letter code.	4217 three-

Children

None.

Facets

Currency Short Converter

com. we bmethods. caf. faces. convert. Currency Short Converter

A converter implementation for java.lang.Short (and short primitive) values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description	Choices	Default
Converter Properties					
pattern	Pattern	String	Custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.		
region	Locale	String	The Locale to use for determining the value.		
customErrorMsg	Custom Error Message	String	A custom message for invalid conversion errors.	object types	and
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.		false
currencyCode	Currency Code	String	The formatting for a currency. letter code.	Use an ISO	4217 three-

Children

None.

Facets

Chapter 30. Formatted Converters

Formatted Big Decimal Converter	662
Formatted Big Integer Converter	662
Formatted Boolean Converter	663
Formatted Date Time Converter	664
Formatted Double Converter	665
Formatted Float Converter	665
Formatted Integer Converter	666
Formatted Long Converter	667
Formatted Number Converter	667
Formatted Short Converter	668

Formatted Big Decimal Converter

com. we bmethods. caf. faces. convert. For matted Big Decimal Converter

A converter implementation for java.math.BigDecimal values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description Default
Converter Properties			
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.
region	Locale	String	The Locale to use for determining the value.
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the false custom error message.

Children

None.

Facets

None.

Formatted Big Integer Converter

com.webmethods.caf.faces.convert.FormattedBigIntegerConverter

A converter implementation for java.math.BigInteger values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conform syntax rules of the Java class java.text.DecimalFormat.	ns to the

Name	Display Name	Туре	Description	Default
region	Locale	String	The Locale to use for determining the value	ie.
customErrorMsg	Custom Error Message	String	A custom message for invalid object types conversion errors.	and
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	false

Children

None.

Facets

None.

Formatted Boolean Converter

com.webmethods.caf.faces.convert.FormattedBooleanConverter

A converter implementation for java.lang.Boolean (and boolean primitive) values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conform syntax rules of the Java class java.text.DecimalFormat.	s to the
region	Locale	String	The Locale to use for determining the valu	ie.
customErrorMsg	Custom Error Message	String	A custom message for invalid object types conversion errors.	and
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	false

Children

None.

Facets

Formatted Date Time Converter

com. we bmethods. caf. faces. convert. For matted Date Time Converter

A converter implementation for java.util.Date values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description	Choices	Default
Converter Properties					
pattern	Date Pattern	String	The input format pattern for pa for the minimum and maximu		ng to a date
useTime	Time Displayed	boolean	Includes or excludes time in the	ne validation	check
timePattern	Time Pattern	String	The input format pattern for parsing a String to time for the minimum and maximum values.		
timeZone	Time Zone	String	The time zone used to interpret the time value.		ılue.
timePrecision	Time Precision	String	Time precision of the displayed value. When left blank (the default) or given a negative number (-1), the precision is taken from the Time Pattern property.	0 - minute 1 - second 2 - millise	s
region	Locale	String	The Locale to use for determin	ing the valu	e.
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.		and
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message custom error message.	ge with the	false

Children

None.

Facets

Formatted Double Converter

com. we bmethods. caf. faces. convert. Formatted Double Converter

A converter implementation for java.lang.Double (and double primitive) values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description Default	
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.	ie
region	Locale	String	The Locale to use for determining the value.	
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.	
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with false the custom error message.	

Children

None.

Facets

None.

Formatted Float Converter

com. we bmethods. caf. faces. convert. Formatted Float Converter

A converter implementation for java.lang.Float (and float primitive) values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conform syntax rules of the Java class java.text.DecimalFormat.	is to the

Name	Display Name	Туре	Description Default
region	Locale	String	The Locale to use for determining the value.
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the false custom error message.

Children

None.

Facets

None.

Formatted Integer Converter

com. we bmethods. caf. faces. convert. For matted Integer Converter

A converter implementation for java.lang.Integer (and int primitive) values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description Default
Converter Properties			
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.
region	Locale	String	The Locale to use for determining the value.
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the false custom error message.

Children

None.

Facets

Formatted Long Converter

com. we bmethods. caf. faces. convert. For matted Long Converter

A converter implementation for java.lang.Long (and long primitive) values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description Default
Converter Properties			
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.
region	Locale	String	The Locale to use for determining the value.
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the false custom error message.

Children

None.

Facets

None.

Formatted Number Converter

com.webmethods.caf.faces.convert.Formatted Number Converter

A converter implementation for java.lang.Number values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description	Default
Converter Properties				
pattern	Pattern	String	A custom formatting pattern that conform syntax rules of the Java class java.text.DecimalFormat.	ns to the

Name	Display Name	Туре	Description De	efault
region	Locale	String	The Locale to use for determining the value.	
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.	d
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.	alse

Children

None.

Facets

None.

Formatted Short Converter

com. we bmethods. caf. faces. convert. For matted Short Converter

A converter implementation for java.lang.Short (and short primitive) values, taking into account localized number formatting.

Properties

Name	Display Name	Туре	Description Default
Converter Properties			
pattern	Pattern	String	A custom formatting pattern that conforms to the syntax rules of the Java class java.text.DecimalFormat.
region	Locale	String	The Locale to use for determining the value.
customErrorMsg	Custom Error Message	String	A custom message for invalid object types and conversion errors.
showCAFErrorMsg	Append CAF Error Message	boolean	Includes the CAF error message with the custom error message.

Children

None.

Facets

Part IV. Validators Reference

Validator Concepts	. 671
Validators	. 675
Formatted Validators	. 683

webMethods Designer - Composite Application Framework Help Version 7.1.1

Chapter 31. Validator Concepts

Using Validators	672
Creating a Custom Validator	672

Using Validators

Validators provide a way to ensure that users enter correct values. You can apply validators only to input controls. The Validator portion of the Palette view contains both standard JSF validators and those developed by Software AG. Some validators, such as the "Credit Card Validator" on page 676, have properties that you can modify on the Properties view.



To add a validator to the design canvas

- 1 In design canvas, locate the input control that needs to have a validator.
- 2 On the Palette view expand the Validator node to find the validator you need.
- 3 Drag the validator until you can drop it on the control.
 - You can only drop onto a control whose boundary is highlighted when you drag the validator over it. If the **o** overlay is visible on the cursor, you cannot drop the validator.
- 4 In the design canvas, click the validator.
 If the validator has any editable properties, they are visible on the Properties view.

For descriptions of properties in a specific validator, see the reference page for that validator.

Creating a Custom Validator

If you need a validator that is not available from among those provided in the Palette view and are familiar with writing Java code, you can create your own validator. Designer provides some initial code to get you started. A custom validator is valid only for the portlet in which it is created. If you plan to use a particular validator in many portlets, you should create your own library and load it through the project.



To create a custom validator

- 1 In the design canvas, select the input or output control for which you need to create a custom validator.
- 2 In the General tab of the Properties view, type a unique ID in the ID field and click away from the view.
- In the design canvas, right-click the control and then click Event Handlers > Validate Handler.

Designer creates some Java code in the managed bean and opens a Java editor to the location of the code. An example of validator code created for an input control is shown here:

```
/**
  * Validator for the control with id='inputID'
  */
public void inputID_validator(javax.faces.context.FacesContext context,
  javax.faces.component.UIComponent component, java.lang.Object value) {
    // TODO: Check the value parameter here, and if not valid, do something like this:
    // throw new ValidatorException(new FacesMessage("Not a valid value!"));
}
```

4 After the TODO comments, place the Java code needed to perform the needed validation.

Chapter 32. Validators

Credit Card Validator	676
Double Range Validator	677
Email Validator	677
Equal Validator	678
Exact RegExp Validator	678
Length Validator	679
Long Range Validator	680
One Of Validator	681
RegExpr Validator	681

Credit Card Validator

com. we bmethods. caf. faces. validator. Credit Card Validator

Checks that the input String of numbers matches one of the card number styles for Visa, MasterCard, AMEX, or Discover.

Properties

Name	Display Name	Туре	Description
Converter Properties			
typeVisa	Visa	Boolean	Account number format.
typeAmex	Amex	Boolean	Account number format.
typeMasterCard	MasterCard	Boolean	Account number format.
typeDiscover	Discover	Boolean	Account number format.

Children

None.

Facets

Double Range Validator

javax.faces.validator.DoubleRangeValidator

A comparison of the value against the specified minimum and maximum settings. If the current value can not be converted to Double, a ValidatorException will be thrown. A standard JSF validator.

JSF Properties

Name	Display Name	Туре	Description			
Converter Properties						
minimum	Minimum	String	If the number value is smaller than the specified minimum, throw a ValidatorException containing a MINIMUM_MESSAGE_ID message. Leave this property blank to exclude the validation check for minimum size.			
maximum	Maximum	String	If the number value is larger than the specified maximum, throw a ValidatorException containing a MAXIMUM_MESSAGE_ID message. Leave this property blank to exclude the validation check for maximum size.			
Children						
None.						
Facets						
None.						

Email Validator

com.webmethods.caf.faces.validator.EmailValidator

Checks that the input string matches the BNF (Backus-Naur Form) for an e-mail address.

Properties

None.

Children

Facets

None.

Equal Validator

com.webmethods.caf.faces.validator.EqualValidator

Matches the value of another component.

Properties

Name	Display Name	Туре	Description
Converter Prope	erties		
for	Component ID	String	The Control ID reference to search for.

Children

None.

Facets

None.

Exact RegExp Validator

com. we bmethods. caf. faces. validator. Exact Reg Exp Validator

Checks that the Regular Expression matches the (sub-)string exactly (matches the entire string). Compare with "RegExpr Validator" on page 681.

Properties

Name	Display Name	Туре	Description		
Converter Prope	Converter Properties				
pattern	Regular Expression Pattern	String	The Regular Expression String to match against.		

Children	1
None.	

Facets
None.

Length Validator

javax.faces.validator.LengthValidator

A comparison of the String length against the specified minimum and maximum settings. Validation failure will result in a ValidatorException. A standard JSF validator.

JSF Properties

Name	Display Name	Туре	Description		
Converter Properties					
minimum	Minimum	String	If the String length is smaller than the specified minimum, throw a ValidatorException containing a MINIMUM_MESSAGE_ID message. Leave this property blank to exclude the validation check for minimum size.		
maximum	Maximum	String	If the String length is larger than the specified maximum, throw a ValidatorException containing a MAXIMUM_MESSAGE_ID message. Leave this property blank to exclude the validation check for maximum size.		

Children

None.

Facets

Long Range Validator

javax.faces.validator.LongRangeValidator

A comparison of the value against the specified minimum and maximum settings. If the current value is a floating point number, or a String that can not be converted to a long, a ValidatorException will be thrown. A standard JSF validator.

JSF Properties

Name	Display Name	Туре	Description
Converter Prope	erties		
minimum	Minimum	String	If the number value is smaller than the specified minimum, throw a ValidatorException containing a MINIMUM_MESSAGE_ID message. Leave this property blank to exclude the validation check for minimum size.
maximum	Maximum	String	If the number value is larger than the specified maximum, throw a ValidatorException containing a MAXIMUM_MESSAGE_ID message. Leave this property blank to exclude the validation check for maximum size.

Children

None.

Facets

One Of Validator

com.webmethods.caf.faces.validator.OneOfValidator

Checks that the value is equal to one of the items in a collection.

Properties

Name	Display Name	Туре	Description
Converter Prop	oerties		
pattern	Search Value	String	The String value to match in the collection.

Children

None.

Facets

None.

RegExpr Validator

com. we bmethods. caf. faces. validator. RegExprValidator

Checks that the Regular Expression matches the portion of the (sub-)string represented by String. Compare with "Exact RegExp Validator" on page 678.

Properties

Name	Display Name	Туре	Description
Converter Properties			
pattern	Regular Expression Pattern	String	The Regular Expression String to match against.

Children

None.

Facets

Chapter 33. Formatted Validators

Formatted Date Range Validator	. 684
Formatted Double Range Validator	. 685
Formatted Long Range Validator	. 686

Formatted Date Range Validator

com. we bmethods. caf. faces. validator. For matted Date Range Validator.

Checks the value of the corresponding component against specified minimum and maximum values, taking into account localized date formatting.

Properties

Name	Display Name	Туре	Description			
Converter Properties						
minimum	Minimum	String	If the date value is earlier than the specified minimum, throw a ValidatorException containing a MINIMUM_MESSAGE_ID message.			
maximum	Maximum	String	If the date value is later than the specified maximum, throw a ValidatorException containing a MAXIMUM_MESSAGE_ID message.			
region	Locale	String	The Locale to use for determining the value.			
datePattern	Date Pattern	String	The input format pattern for parsing a String to a date for the minimum and maximum values.			
pattern	Time Pattern	String	The input format pattern for parsing a String to time for the minimum and maximum values.			
useTime	Use Time	boolean	Set to true to include or exclude time in the validation check.			
timeZone	Time Zone	String	The Time Zone used to interpret the time value.			

Children

None.

Facets

Formatted Double Range Validator

com. we bmethods. caf. faces. validator. For matted Double Range Validator. For matt

Checks the value of the corresponding component against specified minimum and maximum values, taking into account localized number formatting.

If the current component value cannot be converted to a double, a ValidatorException is thrown.

Properties

Name	Display Name	Туре	Description
Converter Prope	erties		
minimum	Minimum	String	If the number value is smaller than the specified minimum, throw a ValidatorException containing a MINIMUM_MESSAGE_ID message.
maximum	Maximum	String	If the number value is larger than the specified maximum, throw a ValidatorException containing a MAXIMUM_MESSAGE_ID message.
region	Locale	String	The Locale to use for determining the value.

	re	

None.

Facets

None.

Formatted Long Range Validator

com. we bmethods. caf. faces. validator. For matted Long Range Validator.

Checks the value of the corresponding component against specified minimum and maximum values, taking into account localized number formatting. If the current component can not be converted to long, a ValidatorException is thrown.

Properties

Name	Display Name	Туре	Description
Converter Prope	erties		
minimum	Minimum	String	If the number value is smaller than the specified minimum, throw a ValidatorException containing a MINIMUM_MESSAGE_ID message.
maximum	Maximum	String	If the number value is larger than the specified maximum, throw a ValidatorException containing a MAXIMUM_MESSAGE_ID message.
region	Locale	String	The Locale to use for determining the value.

Children

None.

Facets

None.

Index

A	Async Search Result Table Control 504
absolute references, Control ID Reference 192	Async Search Result Tree control 510
Access Control Panel control 450	Async Table control 531
access control sample 160	Async Tabs control 403
Add Row Button control 586	Async Tree control 536
Add Row Icon control 587	Atom Feed Icon control 541
Add Row Link control 589	Attachments Panel control 451
adding	authentication credentials
control to a view 33, 73	IS Browser view 118
home link to a Web application 49	MWS Admin view 129
new property to a page bean 109	authentication information for Integration Server at run time
portlet to a My webMethods Server page 77	120
portlet to portlet application 70	authentication information for Integration Server, modifying
reference to another page 112	119
view file to portlet 72	Autocomplete Text Input control 290
view file to Web application 32	
adding and removing rows, table row tools 214	В
Ajax defined 83	Basic Column control 547
Applet control 356	Behavior controls
application initialization parameters, Web applications 50	Custom Drag 236
Application Navigation Configuration Editor 47	Custom Drop 237
application scope	Drag to Move Rows 239
scope	Drag to Reorder Columns 243
application 83	Drag to Resize 244
application server	Drag to Resize Columns 245
deploying component libraries on 62	Raise On Change 246
publishing to 34	Synchronize Values 247
application server, installing 54	Big Decimal converter 640
Array converter 640	Big Integer converter 641
Async Categorized Table control 526	binding data to a control 148
Async Command Button control 250	Bindings view 106
Async Command Icon control 252	adding a new property to a page bean 109
Async Command Interval control 255	adding a reference 112
Async Command Link control 256	creating a new method 107
Async Hidden Command control 259	customizing initial values 110, 111
Async Iterator control 414	moving data objects to 140
Async List control 398	showing all managed beans 111
Async Listbox control 401	Block Edge Panel control 452
	- J

Block Panel control 453 Custom Drop 237 Bookmarklet, CAF Logger 218 Custom Script 490 Boolean converter 641 Data Pages 558 Breadcrumbs control 626 Data Prev/Next 559 breadcrumbs, hiding on the home page of a Web application Data Total 560 49 Data Total Selected 562 Button control 357 Date Input 299 Byte converter 641 Date Range Input 301 Disableable Panel 454 Drag To Move Rows 239 C Drag To Reorder Columns 243 **CAF Base Controls library** Drag To Resize 244 Access Control Panel 450 Drag To Resize Columns 245 Add Row Button 586 Dynamic Map Marker List 426 Add Row Icon 587 Else 417 Add Row Link 589 Export Table Button 562 Applet 356 Extended Select-Many Listbox 308 Async Categorized Table 526 Extended Select-One Listbox 310 Async Command Button 250 File Input 312, 314 Async Command Icon 252 Flash 363 Async Command Interval 255 Formatted Message 365 Async Command Link 256 Formatted Messages 366 Async Hidden Command 259 Formatted Text 366 Async Iterator 414 Frame 368 Async List 398 Google Map 431 Async Listbox 401 Google Map Key 435 Async Table 531 Header 369 Async Tabs 403 Hidden Column 565 Async Tree 536 Hidden Command 271 Atom Feed Icon 541 Hideable Panel 457 Attachments Panel 451 Horizontal Rule 370 Autocomplete Text Input 290 HTML Input 318 Block Edge Panel 452 Icon 371 Block Panel 453 If 418 Breadcrumbs 626 Import View 418 Button 357 In Place Text Input 320 Calendar 549 Include HTML 375 Categorized Table 553 Include Resource Bundle 376 Column Sort Link 557 Include Script 377 Combobox 297 Include Stylesheet 378 Command Icon 263 Initiate Toggle 612 Control Reference 490 Inline Hideable Panel 460 Custom Drag 236 Invoke Map Script 435

Invoke Script 491 Iterator 421 Listbox 406

Load Resource Bundle 423

Map Marker 438 Map Marker Group 443 Map Navigation Control 444

Modal Dialog 276
Modeless Dialog 278
Move Row Down Button 590
Move Row Down Icon 591
Move Row Down Link 593
Move Row Up Button 594
Move Row Up Icon 595
Move Row Up Link 596

Newsfeed 565

One Way Toggle Button 612
One Way Toggle Checkbox 614
One Way Toggle Link 615

Overlay Panel 463
Page Group 465
Parameter 494
Popup Menus 627
Popup Panel 466
Progress Dialog 286
Property Group 467
Property Line 469

Property Sub-Group 469
Raise On Change 246
Refresh Button 385
Refresh Icon 386
Refresh Interval 387
Refresh Link 388

Remove Row Button 597 Remove Row Icon 599 Remove Row Link 600

Return Map Coords Script 445 Return Map Value Script 446 Return Value Script 500

Row Headers Column 566 Script Block 389 Scrollbar Column 568

Scrolling Panel 471

Select All Rows Checkbox 602

Select All Visible Rows Checkbox 603

Select Row Checkbox 604 Select Row Column 605 Select Row Link 606 Select Row On Click 607

Select Row Quad-State Checkbox 608 Select Row Tri-State Checkbox 609

Select-One Button 338
Select-One Link 340
Select-One Output Text 389
Select-One Tabs 345
Servlet Command Form 272

Simple List 408

Simple Schedule Input 347 Specific Formatted Messages 390

Specific Messages 391

Stack Cell 473
Stack Panel 472
Standard Column 568
Static Menus 629
Static Row 474
Submit Group 475
Swapbox 349

Synchronize Values 247

Tabs 411

Titlebar Tabs Wrapper 476 Toggle Button 616

Toggle Dropdown 618 Toggle Link 620 Toggle Menus 631

Toggle Radio Button Group 621

Toggle Tabs 623
Tooltip 476
Tree 575
Tree Toggle 580
Truncated Text 394
Truncating Column 582
CAF Base Controlslibrary

Custom Element 362

CAF JSF Base Controls library

Basic Column 547 Checkbox 292

Checkbox Group 294	CAF.Draggable class 223
Command Button 261	CAF.Droppable class 225
Command Link 265	CAF.Input.Model class 204
Content Parameter 416	CAF.Link.Model class 203
Control Label 360	CAF.Model class 202
Control Parameter 416	CAF.Output.Model class 202
Dropdown 305	CAF.Progress class 221
form 268	CAF.Request class 222
Grid Panel 455	CAF.Select.Model class 205, 228
Hidden Input 317	CAF.Table.Model class 207
Image 374	CAF.Table.Row.Model class 207
Inline Panel 462	CAF.Tooltip class 222
Link 379	CAF.Tree.Model class 209
Message 381	CAF.Updater class 223
Messages 383	Calendar control 549
Multi-Line Text Input 322	Categorized Table control 553
Naming Container 424	change control type on the design canvas 102
Option 324	change labels on the design canvas 103
Option Group 326	Character converter 642
Radio Button 328	Checkbox control 292
Radio Button Group 330	Checkbox Group control 294
Secret Input 333	class
Select-Many Listbox 335	CAF.Checkbox.Model 205
Select-One Listbox 342	CAF.Command.Model 203
Table 570	CAF.Dialog 220
Text 392	CAF.Draggable 223
Text Input 351	CAF.Droppable 225
CAF Logger 218	CAF.Input.Model 204
Bookmarklet 218	CAF.Link.Model 203
Firebug, and 219	CAF.Model 202
CAF Portlet Controls library	CAF.Output.Model 202
Async Search Result Table 504	CAF.Progress 221
Async Search Result Tree 510	CAF.Request 222
Portlet Parameter 483	CAF.Select.Model 205, 228
Portlet Simple Link 484	CAF.Table.Model 207
Portlet URL 487	CAF.Table.Ros.Model 207
Portlet URL Script 496	CAF.Tooltip 222
Portlet URL Script Parameter 498	CAF.Tree.Model 209
Search Bar 514	CAF.Updater 223
Search Result Table 517	client-side functionality, hideable controls 194
CAF.Checkbox.Model class 205	client-side libraries 217
CAF.Command.Model class 203	CAF Logger 218
CAF.Dialog class 220	csv 217

string object 218	Async Hidden Command 259
client-side model 201	Async Iterator 414
bglobal variables 211	Async List 398
listeners 209	Async Listbox 401
model references 211	Async Table 531
paging 208	Async Tabs 403
script placement 211	Async Tree 536
template row 208	Atom Feed Icon 541
Collection converter 642	Attachments Panel 451
Column Sort Link control 557	Autocomplete Text Input 290
Combobox control 297	Block Edge Panel 452
Command Button control 261	Block Panel 453
Command controls	Breadcrunbs 626
Async Command Button 250	Button 357
Async Command Icon 252	Calendar 549
Async Command Interval 255	Categorized Table 553
Async Command Link 256	Column Sort Link 557
Async Hidden Command 259	Combobox 297
Command Button 261	Command Icon 263
Command Icon 263	Control Reference 490
Command Link 265	Custom Drag 236
Form 268	Custom Drop 237
Hidden Command 271	Custom Element 362
Servlet Command 272	Custom Script 490
Command Icon control 263	Data Pages 558
Command Link control 265	Data Prev/Next 559
common effect options, Scriptaculous Effects 200	Data Total 560
component libraries	Data Total Selected 562
adding 38	Date Input 299
deploying 39	Date Range Input 301
removing 39	Disableable Panel 454
component library	Drag To Move Rows 239
CAF Base Controls	Drag To Reorder Columns 243
Access Control Panel 450	Drag To Resize 244
Add Row Button 586	Drag To Resize Columns 245
Add Row Icon 587	Dynamic Map Marker List 426
Add Row Link 589	Else 417
Applet 356	Export Table Button 562
Async Categorized Table 526	Extended Select-Many Listbox 308
Async Command Button 250	Extended Select-One Listbox 310
Async Command Icon 252	File Input 312, 314
Async Command Interval 255	Flash 363
Async Command Link 256	Formatted Message 365

Formatted Messages 366 Formatted Text 366

Frame 368 Google Map 431 Google Map Key 435

Header 369 Hidden Column 565 Hidden Command 271 Hideable Panel 457 Horizontal Rule 370 HTML Input 318

Icon 371 If 418

Import View 418 In Place Text Input 320 Include HTML 375

Include Resource Bundle 376

Include Script 377 Include Stylesheet 378 Initiate Toggle 612 Inline Hideable Panel 460 Invoke Map Script 435 Invoke Script 491 Iterator 421

Listbox 406

Load Resource Bundle 423

Map Marker 438 Map Marker Group 443 Map Navigation Control 444

Modal Dialog 276 Modeless Dialog 278

Move Row Down Button 590 Move Row Down Icon 591 Move Row Down Link 593 Move Row Up Button 594 Move Row Up Icon 595 Move Row Up Link 596

Newsfeed 565

One Way Toggle Button 612
One Way Toggle Checkbox 614
One Way Toggle Link 615

Overlay Panel 463 Page Group 465 Parameter 494
Popup Menus 627
Popup Panel 466
Progress Dialog 286
Property Group 467
Property Line 469
Property Sub-Group 469
Raise On Change 246
Refresh Button 385
Refresh Icon 386
Refresh Interval 387
Refresh Link 388

Retresh Link 388
Remove Row Button 597
Remove Row Icon 599
Remove Row Link 600

Return Map Coords Script 445 Return Map Value Script 446 Return Value Script 500 Row Headers Column 566

Script Block 389 Scrollbar Column 568 Scrolling Panel 471

Select All Rows Checkbox 602 Select All Visible Rows Checkbox 603

Select Row Checkbox 604 Select Row Column 605 Select Row Link 606 Select Row On Click 607

Select Row Quad-State Checkbox 608 Select Row Tri-State Checkbox 609

Select-One Button 338 Select-One Link 340 Select-One Output Text 389 Select-One Tabs 345 Servlet Command Fom 272

Simple List 408

Simple Schedule Input 347 Specific Formatted Messages 390

Specific Messages 391 Stack Cell 473 Stack Panel 472 Standard Column 568 Static Menus 629

Static Row 474	Select-One Listbox 342
Submit Group 475	Table 570
Swapbox 349	Text 392
Sychronize Values 247	Text Input 351
Tabs 411	CAF Portlet Controls
Titlebar Tabs Wrapper 476	Async Search Result Table 504
Toggle Button 616	Async Search Result Tree 510
Toggle Dropdown 618	Portlet Parameter 483
Toggle Link 620	Portlet Simple Link 484
Toggle Menus 631	Portlet URL 487
Toggle Radio Button Group 621	Portlet URL Script 496
Toggle Tabs 623	Portlet URL Script Parameter 498
Tooltip 476	Search Bar 514
Tree 575	Search Bar Result Table 517
Tree Toggle 580	controls listed by 181
Truncated Text 394	deploying on an application server 62
Truncating Column 582	MWS Control
CAF JSF Base Controls	Extended Portlet Parameter 480
Basic Column 547	Extended Portlet URL 481
Checkbox 292	Portal Resource Picker Dialog 280
Checkbox Group 294	Portlet Include 482
Command Button 261	Principal Picker Dialog 283
Command Link 265	configuring
Content Parameter 416	JBoss 57
Control Label 360	Tomcat 54
Control Parameter 416	connecting
Dropdown 305	to My webMethods Server 74
Form 268	connecting to
Grid Panel 455	Integration Server in IS Browser view 116
Hidden Input 317	My webMethods Server on MWS Admin view 127
Image 374	connecting to a database 113
Inline Panel 462	Content Parameter control 416
Link 379	content parameters and page templates 45
Message 381	control
Messages 383	adding to a view 33, 73
Multi-Line Text Input 322	Control ID Reference 192
Naming Container 424	absolute references 192
Option 324	absolute versus relative 193
Option Group 326	external portlet references 193
Radio Button 328	literal HTML references 193
Radio Button Group 330	nearest control algorithm 193
Secret Input 333	relative references 192
Select-Many Listbox 335	Control Label control 360

Control Parameter control 416	Formatted Float 665
control parameters and page templates 46	Formatted Integer 666
Control Reference control 490	Formatted Long 667
control source order	Formatted Number 667
toggle controls 197	Formatted Short 668
controls	Integer 645
change control type 102	Iterator 645
change labels 103	Long 646
edit CSS values 102	Map 646
listed by component library 181	Number 647
matching 101	Short 648
minimizing 101	String Array 649
moving 101	WmPortallURI 649
resizing 101	core controls sample 161
select a parent or child control 103	core providers sample 161
show a view from 103	creating
conventions used in this document 17	custom converters 636
Converters	custom validators 672
adding to a control 147	method on Bindings view 107
Array 640	portlet application 68
Big Decimal 640	portlet application on the Solutions view 150
Big Integer 641	portlet on the Solutions view 150
Boolean 641	portlet view file on the Solutions view 152
Byte 641	security roles 94
Character 642	Web application
Collection 642	for JBoss 27
Currency Big Decimal 652	for multiple server types 29
Currency Big Integer 653	for My webMethods Server 30
Currency Double 654	for Tomcat 26
Currency Float 655	Web application on the Solutions view 153
Currency Integer 656	Web application view file on the Solutions view 154
Currency Long 657	Web service connector 121
Currency Number 658	creating a database connector 114
Currency Short 659	Credit Card Validator 676
custom, creating 636	CSS
Date and Time 643	for individual control 41
Double 644	view file 42
Float 644	Web applications 41
Formatted Big Decimal 662	CSS, editing 102
Formatted Big Integer 662	csv, client-side libraries 217
Formatted Boolean 663	Currency Big Decimal converter 652
Formatted Date Time 664	Currency Big Integer converter 653
Formatted Double 665	Currency Double converter 654

Currency Float converter 655	Dialog controls
Currency Integer converter 656	Modal Dialog 276
Currency Long converter 657	Modeless Dialog 278
Currency Number converter 658	Portal Resource Picker Dialog 280
Currency Short converter 659	Principal Picker Dialog 283
custom converters, creating 636	Process Dialog 286
Custom Drag control 236	Disableable Panel control 454
Custom Drop control 237	displaying the My webMethods perspective 66
Custom Element control 362	documentation
Custom Script control 490	additional 17
custom validators, creating 672	conventions used 17
customizing initial values for a page 110, 111	feedback 17
	using effectively 17
D	Double converter 644
Data Export Selected control 562	Double Range Validator 677
Data Pages control 558	Drag to Move Rows control 239
Data Prev/Next control 559	Drag to Reorder Columns control 243
Data Source Explorer view	Drag to Resize Columns control 245
connecting to a database 113	Drag to Resize control 244
creating a database connector 114	Dynamic Map Marker List control 426
Data Total control 560	
database connector, creating 114	E
database, connecting to 113	edit CSS values on the design canvas 102
Date and Time converter 643	effect transitions, Scriptaculous Effects 201
Date Input control 299, 305	Else control 417
Date Range Input control 301	Email Validator 677
debugging from the design canvas 106	environment variables, Web applications 51
deleting	Exact RegExp Validator 678
Web service connector 122	Export Table Button control 562
deploying	Extended Portlet Parameter control 480
component libraries on an application server 62	Extended Portlet URL control 481
portlets to My webMethods Server 129	Extended Select-Many Listbox control 308
deploying component libraries 38, 39	Extended Select-One Listbox control 310
Design canvas	external portlet references, Control ID Reference 193
moving objects to 140	external Web Browser, specifying 155
design canvas	
debugging from 106	F
Instant Preview 103	Favorites node in Palette view 141
live preview 104	file
manipulating controls 100	jboss-web.xml 60
toolbar 99	login-config.xml 61
using 97	file export sample 162

File Input control 312	Н
Filter Input control 314	Header control 369 Hidden Column control 565
filterable controls 226	
filtering examples	Hidden Command control 271
autocomplete text field 233	Hidden Input control 317
filtered combobox medium list 232	hideable controls 194
filtered dropdown medium list 232	client-side functionality 194
filtered listbox large list 230	instances 196
filtered listbox medium list 229	lazy loading 195
filtered swapbox large list 231	progress bar customization 196
filtered swapbox medium list 230	refreshOnShow property 195
filtered table (current page) 233	server-side properties 194
filtered table (entire table) 233	suppressInputs property 195
Firebug and CAF Logger 219	twoPass property 195
Flash control 363	Hideable Panel control 457
Float converter 644	home link in a Web application 49
Form control 268	Horizontal Rule control 370
Formatted Big Decimal converter 662	HTML Input control 318
Formatted Big Integer converter 662	'
Formatted Boolean converter 663	1
Formatted Date Range Validator 684	Icon control 371
Formatted Date Time converter 664	If control 418
Formatted Double converter 665	Image control 374
Formatted Double Range Validator 685	image URLs 212
Formatted Float converter 665	Import View control 418
Formatted Integer converter 666	import view control 1775
Formatted Long converter 667	In Place Text Input control 320
Formatted Long Range Validator 686	Include HTML control 375
Formatted Message control 365	Include Resource Bundle control 376
Formatted Messages control 366	Include Script control 377
Formatted Number converter 667	Include Stylesheet control 378
Formatted Short converter 668	initial values for a page, customizing 110, 111
Formatted Text control 366	Inline Hideable Panel control 460
Frame control 368	Inline Panel control 462
	Input controls
G	Autocomplete Text Input 290
global variables, client-side model 211	Checkbox 292
Google Map control 431	Checkbox Group 294
Google Map Key control 435	Combobox 297
Grid Panel control 455	Date Input 299, 305
	Date Range Input 301
	Extended Select-Many Listbox 308

Extended Select-One Listbox 310	Iterator control 421
File Input 312	Iterator converter 645
Filter Input 314	
Hidden Input 317	J
HTML Input 318	JavaServer Faces 82
In Place Text Input 320	JBoss
Multi-Line Text Input 322	configuring 57
Option 324	creating Web applications for 27
Option Group 326	installing
Radio Button 328	installing an application server 54
Radio Button Group 330	security domains 60
Secret Input 333	jboss-web.xml file 60
Select-Many Listbox 335	JSF scopes 83
Select-One Button 338	JSR 168 82
Select-One Link 340	
Select-One Listbox 342	L
Select-One Tabs 345	
Simple Schedule Input 347	lazy loading, hideable controls 195 Length Validator 679
Swapbox 349	Link control 379
Text Input 351	List controls
instances	Async List 398
toggle controls 198	Async List 370 Async Listbox 401
instances hideable controls 196	Async Tabs 403
Instant Preview 103	Listbox 406
Integer converter 645	Simple List 408
Integration Server	Tabs 411
authentication at run time 120	Listbox control 406
dissplaying in a browser 125	listeners, client-side model 209
external Web Browser 155	literal HTML references, Control ID Reference 193
modifying authentication 119	live preview 104
Integration Server, connecting to in IS Browser view 116	external Web browser 155
Integration Server, renaming on IS Browser view 123	Load Resource Bundle control 423
Invoke Map Script control 435	locale order sample 162
Invoke Script control 491	localization sample 163
IS Browser view 115	Logic controls
authentication credentials 118	Async Iterator 414
changing properties 124	Content Parameter 416
connecting to Integration Server 116	Control Parameter 416
creating a Web service connector 121	Else 417
deleting a Web service connector 122	If 418
displaying Integration Server 125	Import View 418
renaming Integration Server 123	Iterator 421

Load Resource Bundle 423 Naming Container 424 login-config.xml file 61 Long converter 646 Long Range Validator 680	connecting to My webMethods Server 127 deploying portlets 129 displaying My webMethods Server 136 exporting from My webMethods Server 130 importing to My webMethods Server 132 renaming My webMethods Server 133
M	toolbar 128
managed beans, showing all 111	MWS Control library
Map controls	Extended Portlet Parameter 480
Dynamic Map Marker List 426	Extended Portlet URL 481
Google Map 431	Portal Resource Picker Dialog 280
Google Map 431 Google Map Key 435	Portlet Include 482
Invoke Map Script 435	Principal Picker Dialog 283
Map Marker 438	My webMethods
Map Marker Group 443	perspective, overview 86
Map Navigation Control 444	My webMethods perspective, displaying 66
Return Map Coords Script 445	My webMethods Server
Return Map Value Script 446	connecting to 74
Map converter 646	creating Web applications for 30
Map Marker control 438	displaying in a browser 136
Map Marker Group control 443	exporting from 130
Map Navigation Control control 444	external Web Browser 155
matching controls on the design canvas 101	importing to 132
Message control 381	publishing to 77
Messages control 383	renaming 133
method, creating a new 107	running in debug mode 76
minimizing controls on the design canvas 101	uninstalling portlet applications from 78
Modal Dialog control 276	My webMethods Server, connecting to on MWS Admin view
model references, client-side model 211	127
Modeless Dialog control 278	
Move Row Down Button control 590	N
Move Row Down Icon control 591	Naming Container control 424
Move Row Down Link control 593	navigation in Web applications 47
Move Row Up Button control 594	nearest control algorithm, Control ID Reference 193
Move Row Up Icon control 595	Newsfeed control 565
Move Row Up Link control 596	northwind sample 164
moving controls on the design canvas 101	Number converter 647
moving rows, table row tools 215	
Multi-Line Text Input control 322	0
MWS Admin view 125	One Of Validator 681
authentication credentials 129	Option control 324
changing properties 134	Option Group control 326

Outline view 137	page 43
Output controls	paging, client-side model 208
Applet 356	Palette view 138
Button 357	customizing 142
Control Label 360	displaying deprecated items 143
Custom Element 362	filtering controls 144
Flasht 363	managing Favorites 141
Formatted Message 365	moving objects from 140
Formatted Messages 366	toolbar 143
Formatted Text 366	Panel controls
Frame 368	Access Control Panel 450
Header 369	Attachments Panel 451
Horizontal Rule 370	Block Edge Panel 452
Icon 371	Block Panel 453
Image 374	Disableable Panel 454
Include HTML 375	Grid Panel 455
Include Resource Bundle 376	Hideable Panel 457
Include Script 377	Inline Hideable Panel 460
Include Stylesheet 378	Inline Panel 462
Link 379 3	Overlay Panel 463
Message 381	Page Group 465
Messages 383	Popup Panel 466
Parameterized Text 384	Property Group 467
Refresh Button 385	Property Line 469
Refresh Icon 386	Property Sub-Group 469
Refresh Interval 387	Scrolling Panel 471
Refresh Link 388	Stack Panel 472
Script Block 389	Static Cell 473
Select-One Output Text 389	Static Row 474
Specific Formatted Messages 390	Submit Group 475
Specific Messages 391	Titlebar Tabs Wrapper 476
Text 392	Tooltip 476
Truncated Text 394	Parameter control 494
Overlay Panel control 463	Parameterized Text control 384
•	Popup Menus control 627
P	Popup Panel control 466
page bean, adding a new property to 109	Portal Resource Picker Dialog control 280
Page Group control 465	portlet
page templates	adding a view file 72
content parameters 45	adding to a My webMethods Server page 77
control parameters 46	adding to portlet application 70
templates	creating on the Solutions view 150
(Simpletos	deploying to My webMethods Server 129

preferences, creating 89	perform wiring
view file, creating on the Solutions view 152	perform wiring using preferences 90
portlet application	storing 92
adding a portlet 70	Preview Server 104
adding user attributes 95	Principal Picker Dialog control 283
associating to My webMethods Server 76	Process Dialog control 286
creating 68	program code conventions in this document 17
creating on the Solutions view 150	progress bar customization, hideable controls 196
defined 82	project template, Web applications 40
publishing to My webMethods Server 77	Properties view 145
samples	actions in 146
access control 160	adding converters and validators to a control 147
core controls 161	binding data to a control 148
core providers 161	toolbar 145
file export 162	properties, changing on IS Browser view 124
importing into My webMethods perspective 158	properties, changing on MWS Admin view 134
locale order 162	Property Group control 467
localization 163	Property Line control 469
northwind 164	Property Sub-Group control 469
portlet links 164	publishing to an application server 34
search 165	publishing to My webMethods Server 77
wired config 165	_
wizard 166	R
troubleshooting 79	Radio Button control 328
uninstalling from My webMethods Server 78	Radio Button Group control 330
Portlet Application Configuration editor 88	Raise On Change control 246
creating portlet preferences 89 Portlet controls	Refresh Button control 385
Extended Portlet Parameter 480	Refresh Icon control 386
Extended Portlet URL 481	Refresh Interval control 387
Portlet Include 482	Refresh Link control 388
Portlet Parameter 483	refreshOnShow property, hideable controls 195
Portlet Simple Link 484	RegExpr Validator 681
Portlet URL 487	relative references, Control ID Reference 192
Portlet Include control 482	Remove Row Button control 597
portlet links sample 164	Remove Row Icon control 599 Remove Row Link control 600
Portlet Parameter control 483	
Portlet Simple Link control 484	request scope 83 resizing controls on the design canvas 101
Portlet Tabs Tutorial 170	Return Map Coords Script control 445
Portlet URL control 487	Return Map Value Script control 446
Portlet Url Script control 496	Return Value Script control 500
Portlet Url Script Parameter control 498	Row Headers Column control 566
preferences	Now Houseld Column Control 500

\$	Select Row Checkbox control 604
scope	Select Row Column control 605
request 83	Select Row Link control 606
session 83	Select Row On Click control 607
Script Block control 389	Select Row Quad-State Checkbox control 608
Script controls	Select Row Tri-State Checkbox control 609
Control Reference 490	selecting rows, table row tools 216
Custom Script 490	Select-Many Listbox control 335
Invoke Script 491	Select-One Button control 338
Parameter 494	Select-One Link control 340
Portlet Url Script 496	Select-One Listbox control 342
Portlet Url Script Parameter 498	Select-One Output Text control 389
Return Value Script 500	Select-One Tabs control 345
script placement, client-side model 211	server-side properties, hideable controls 194
Scriptaculous Effects 198	Servlet Command control 272
common effect options 200	session scope 83
effect transitions 201	Short converter 648
standard 199	show a view from the design canvas 103
Scrollbar Column control 568	showing all managed beans 111
Scrolling Panel control 471	Simple List control 408
Search Bar control 514	Simple Schedule Input control 347
Search controls	skinning 213
Async Search Result Table 504	Solutions view 149
Async Search Result Tree 510	creating a portlet 150
Search Bar 514	creating a portlet application 150
Search Result Table 517	creating a portlet view file 152
Search Result Tree 521	creating a Web application 153
Search Result Table control 517	creating a Web application view file 154
Search Result Tree control 521	Specific Formatted Messages control 390
search sample 165	Specific Messages control 391
Secret Input control 333	Stack Panel control 472
security for Web applications 58	Standard Column control 568
security on JBoss 60	Starter Web Application project template 40
security on Tomcat 58	Static Cell control 473
security roles	Static Menus control 629
Access Control Panel control 95	Static Row control 474
binding to a control 94	String Array converter 649
creating 94	string object, client-side libraries 218
security roles for Web applications 61	style sheets in Web applications 41
select a child or parent control on the design canvas 10	Submit Group control 475
Select All Rows Checkbox control 602	suppressInputs property, hideable controls 195
Select All Visible Rows Checkbox control 603	Swapbox control 349
	Synchronize Values control 247

T	Select Row Link 606
Table control 570	Select Row On Click 607
Table controls	Select Row Quad-State Checkbox 608
Async Categorized Table 526	Select Row Tri-State Checkbox 609
Async Table 531	table row tools 214
Async Tree 536	adding and removing rows 214
Atom Feed Icon 541	moving rows 215
Basic Column 547	selecting rows 216
Calendar 549	Tabs control 411
Categorized Table 553	template row, client-side model 208
Column Sort Link 557	Text control 392
Data Pages 558	Text Input control 351
Data Prev/Next 559	Titlebar Tabs Wrapper control 476
Data Total 560	toggle controls 197
Data Total Selected 562	behavior with hideable controls 197
Export Table Button 562	behavior with non-hideable controls 197
Hidden Column 565	control source order 197
Newsfeed 565	instances 198
Row Headers Column 566	Toggle Menus control 631
Scrollbar Column 568	Tomcat
Standard Column 568	configuring 54
Table 570	creating Web applications for 26
Tree 575	installing 54
Tree Toggle 580	re-render automatically 63
Truncating Column 582	security on 58
Table Row controls	toolbar
Add Row Button 586	design canvas 99
Add Row Icon 587	IS Browser view 117
Add Row Link 589	live preview 104
Move Row Down Button 590	MWS Admin view 128
Move Row Down Icon 591	Palette view 143
Move Row Down Link 593	Properties view 145
Move Row Up Button 594	Tooltip control 476
Move Row Up Icon 595	Tree control 575
Move Row Up Link 596	Tree Toggle control 580
Remove Row Button 597	troubleshooting information 17
Remove Row Icon 599	troubleshooting portlet applications 79
Remove Row Link 600	Truncated Text control 394
Select All Rows Checkbox 602	Truncating Column control 582
Select All Visible Rows Checkbox 603	tutorials
Select Row Checkbox 604	create a composite application project 171
Select Row Column 605	create a portlet 171
JOIGGE ROW GOIGHTH GOJ	Portlet Tabs 170

twoPass property, hideable controls 195	W
typographical conventions in this document 17	WAR file defined 82
	Web application
U	adding a home link 49
unstalling portlet applications from My webMethods Server	adding a view file 32
78	adding component libraries 38
user attributes	Application Navigation Configuration Editor 47
adding to a portlet application 95	associating with an application server 34
in a binding expression 95	component libraries for 38
in a binding expression 70	creating
V	for JBoss 27
	for multiple server types 29
Validators	for My webMethods Server 30
adding to a control 147	for Tomcat 26
Credit Card 676	creating on the Solutions view 153
custom, creating 672	CSS 41
Double Range 677	individual control 41
Email 677	view file 42
Equal	deploying component libraries 39
Equal Validator 678	environment variables 51
Exact RegExp 678	hiding breadcrumbs on the home page 49
Formatted Date Range 684	initialization parameters 50
Formatted Double Range 685	navigation in 47
Formatted Long Range 686	project template 40
Length 679	publishing to an application server 34
Long Range 680	removing component libraries 39
One Of 681	security 58
RegExpr 681	security roles 61
view	view file, creating on the Solutions view 154
adding a control to 33, 73	Web server
Bindings view 106	running in debug mode 64
Data Source Explorer view 113	Web service connector
importing 43	creating 121
IS Browser view 115	deleting 122
MWS Admin view 125	Webapp controls
Outline view 137	Breadcrumbs 626
Palette 138	Popup Menus 627
Properties 145	Static Menus 629
Solutions view 149	Toggle Menus 631
view file	wired config sample 165
adding to portlet 72	wizard sample 166
adding to Web application 32	wm_cafshared.war file 55, 57
	WmPortalIURI converter 649