

# Web Development Kit Reference Guide



Version 5.3 SP3  
July 2006

Copyright © 1994-2006 EMC Corporation. All rights reserved.

# Table of Contents

---

<b>Preface</b> .....	5
<b>Chapter 1</b>	
<b>Introduction</b> .....	7
Where to configure controls, actions, and components .....	8
Dynamic component launching .....	9
<b>Chapter 2</b>	
<b>Controls Reference</b> .....	11
Common control attributes .....	11
Event attributes .....	13
Action attributes.....	14
Dynamic action attribute values .....	14
5.2.5 content transfer applet attributes.....	15
Runtime tag attribute override .....	19
HTML controls.....	19
<b>Chapter 3</b>	
<b>Actions Reference</b> .....	203
<b>Chapter 4</b>	
<b>Components Reference</b> .....	481

## List of Tables

Table 2-1.	Common control attributes .....	12
Table 2-2.	Event attributes .....	13
Table 2-3.	Common action control attributes.....	14
Table 2-4.	dmappletattributes .....	16
Table 2-5.	secureapplet attributes.....	16
Table 2-6.	contentxferapplet attributes.....	17
Table 2-7.	downloadcontentapplet attributes .....	18
Table 2-8.	uploadcontentapplet attributes.....	18
Table 2-9.	HTML layout controls.....	19
Table 2-10.	Global date input control configuration elements.....	64
Table 2-11.	Global datetime control configuration elements.....	67
Table 2-12.	Global datetime control configuration elements.....	69
Table 2-13.	Rich text configuration elements.....	158
Table 2-14.	Rich text configuration elements.....	160
Table 2-15.	Rich text editor configuration elements (<editor>).....	161

# Preface

---

This reference manual is part of the documentation set for the Documentum Web Development Kit (WDK). Other materials related to WDK are:

- *Web Development Kit and Client Applications Development Guide*
- *Webtop Development Guide*
- *Web Development Kit and Applications Tutorial*
- Javadocs for WDK, Webtop, and Documentum Foundation Classes (DFC)

The installation program for WDK installs this documentation, and it is also available from the Documentum download site. Check the Documentum download Web site (<https://documentum.subscribenet.com>) for revisions of the documentation.

The Documentum developer Web site, [developer.documentum.com](http://developer.documentum.com), provides sample code, tips, white papers, and a wealth of information to assist you in developing Documentum-enabled applications.

This guide is intended for two tasks:

- **Configuration**

Changes to XML files or modifications of JSP pages to configure controls on the page. Does not require a developer license.

- **Customization**

Extending WDK classes or modifying the JSP pages to include new functionality. Requires a developer license.

# Revision history

The following changes have been made to this document:

## Revision History

<b>Revision Date</b>	<b>Description</b>
March 2005	Initial release
August 2005	Information on changes and additions to controls, actions, and components for SP1
September 2005	Added definitions for configuration and customization
January 2006	Added TLD identity for each control. Added new parameters for various component definitions that changed in SP2.
July 2006	Added numerous components and actions for new Records Manager SP3 features. Modified xforms component reference entry. Modified view action reference entry.

## Introduction

This guide contains reference information on all controls, actions, and components in WDK:

- [Chapter 2, Controls Reference](#)
- [Chapter 3, Actions Reference](#)
- [Chapter 4, Components Reference](#)

For general information on component configuration and customization, with examples, see *Web Development Kit and Client Applications Development Guide*. For information about components specific to a WDK client product, see the development guide for that product.

Each Documentum control is defined in a tag library definition file (\*.tld) located in /WEB-INF/tld. Common control attributes which are shared by many Documentum controls are described in the introduction to controls, [Chapter 2, Controls Reference](#). Attributes that are specific to a control are described for each control.

Each Documentum action or component definition includes common XML elements, such as <scope>, <component id>, and <params>. Some action or component definitions contain supplemental XML elements, which are interpreted by the component class or the action precondition or execution class. For example, the doclist component has configurable <objectfilters> and <column> elements (among others) that supplement the common elements. All parameters, and all configurable elements that are specific to the action or component, are described for each action or component.

**Note:** All actions and components are configurable, but only the configurable features that are specific to a definition are documented.

The following information about actions and components is provided. If a section is not present for the action or component, it is not present in the definition:

- **Parameters**

Components and actions can have required or optional parameters that are defined with a <params> element. Action parameters may be used by the action precondition or execution class or simply passed on to the component that is called by the action. Components can be called by an action, a URL, JavaScript, or server-side code, passing in the parameters. All parameters are passed as strings, but some components may cast the parameter into another type. These expected parameter types are noted.

- **Configuration file**

When you wish to configure a component, copy the definition file to `/custom/config` and use an `extends` attribute on the component element. See *Web Development Kit and Client Applications Development Guide* for inheritance guidelines.

- **Parent**

A component definition can extend a base component definition and inherit all or part of the configuration of the parent definition. If an element is defined in the extended definition, that element overrides the contents of the element in the parent definition. For example, if the parent specifies an `objectId` parameter and the extended component has an empty `<params>` element, then no parameters are defined or utilized for the child component.

- **Elements**

An action or component may define configurable elements that are used by the action or component class.

- **Usage**

This section, if present, describes how to configure or customize the action or component.

## Where to configure controls, actions, and components

The following section describes where to find Configuration files.

- **Controls**

Controls are configured in a JSP page. Most of the configurable control attributes are included in the tag library descriptor files (`*.tld`), which are located in `APP_HOME/WEB-INF/tlds`. If a control extends another control, the child control inherits all of the configurable tag attributes of the parent control.

The control itself is configured in a JSP page. It can be configured differently for each component JSP page.

- **Actions**

Actions are configured in action Configuration files (`/config/*.xml`). The files are located in `APP_HOME/wdk/config` and `APP_HOME/webcomponent/config`. If you have a Web client such as Webtop, the application has a config folder `APP_HOME/Webtop/config`. You should copy the Configuration file to your custom Web application layer, for example, `APP_HOME/custom/config`, extend the definition, and then configure it. For information on how to extend WDK components, see *Web Development Kit and Client Applications Development Guide*.

- **Components**



Components are configured in component Configuration files (/config/\*.xml). Component Configuration files are located in `APP_HOME/wdk/config` and `APP_HOME/webcomponent/config`. If you have a Web client such as Webtop, the Configuration files are located in `APP_HOME/Webtop/config`. You should copy the Configuration file to your custom Web application layer, for example, `APP_HOME/custom/config`, extend the definition, and then configure it. For information on how to extend WDK components, see *Web Development Kit and Client Applications Development Guide*.

## Dynamic component launching

You can specify which component is launched at runtime based on a dynamic filter in the action definition. Use the `<execution>.<dynamicfilter>` element in the action definition to specify a class that extends `LaunchComponentFilter` and implements the filter. The filter uses an evaluation class that implements `ILaunchComponentEvaluator`. The evaluator class evaluates which component to launch from among the options listed in the filter definition by matching the current context to criteria values in the configuration. For example, the class `ContentTransferLaunchComponentEvaluator` `evaluate()` method returns the appropriate type of content transfer component based on the application environment: An HTTP component is launched for portal environments, UCF is launched for UCF content transfer, and an applet is launched for legacy custom applet components.

The contents of the filter element are similar to the following:

```

1<dynamicfilter class=com.mycompany.MyFilter>
  2<option>
    3<criteria>
      4<criterion name="contenttransfer" value="ucf"
        evaluatorclass="com.mycompany.ContentTransferLaunchComponentEvaluator"/>
    </criteria>
    5<selection>
      6<component>import</component>
      7<container>importcontainer</container>
    </selection>
  </option>*
</dynamicfilter>

```

- 1 Specifies the filter class and contains two or more options for launching different components from the same action
- 2 Defines an option that will be launched when criteria are met. You must provide at least two options for the filter. Your evaluator class must return a value to the filter class that matches one of the criterion names in your filter definition.

- 3 Contains zero or more <criteria> elements whose values must match conditions as implemented by the filter class in order for the component specified in the <selection> element to be launched. If this element is empty, the selection is the default selection.
- 4 Defines a criterion that must be matched. The criterion name and value are evaluated by the dynamicfilterclass and the corresponding evaluator class is called. The criterion value is evaluated by the criterion evaluator class, which matches the criterion value against its business logic and determines which component selection should be launched.
- 5 Specifies the component and container that will be launched when the criterion is matched
- 6 Specifies the component that will be launched
- 7 Specifies the container that will be launched

## Controls Reference

The configurable attributes for WDK controls are documented in this section. Control attributes are configured in the JSP page. For examples of the usage and configuration of the individual WDK , see the JSP pages in the WDK and WDK client products such as Webtop and Web Publisher. For information about controls specific to a WDK client product, see the development guide for that product.

The following topics describe attributes that are common to groups of controls:

- [Common control attributes, page 11](#)
- [Event attributes, page 13](#)
- [Action attributes, page 14](#)
- [Dynamic action attribute values, page 14](#)
- [5.2.5 content transfer applet attributes, page 15](#)
- [Runtime tag attribute override, page 19](#)
- [HTML controls, page 19](#)

Portlet component controls are documented in *WDK for Portlets Development Guide*.

## Common control attributes

The following attributes are defined by the ControlTag class and can be set on any control. Some of these attributes may not be used by the tag classes in the rendering of the control, however. For the exact list of all attributes that are supported by a tag class, see the tag library.

Event attributes such as `runatclient`, `onchange`, `onselect`, or `onclick`, are documented in [Event attributes, page 13](#).

The table below describes attributes that are common to most controls. If a common attribute requires a value for a particular control, that requirement is noted in the control description.

**Table 2-1. Common control attributes**

Attribute	Description
cssclass	Sets the CSS class that is applied to the control.
enabled	Sets whether the control is enabled. This attribute is not available for the celllist, panel, actionbutton, actionlink, and actionimage controls.
focus	True to allow the control to accept focus.
id	ID for the control. The ID is generated by the framework. You can also set the ID in the JSP page.
label	Contains a string value or <nlsid> element that maps to a string label that is displayed in the UI.
name	Sets the name for the control. Named controls are cached on the server. Controls with the same name are automatically indexed.
nlsid	Sets the NLS lookup key. The string corresponding to the NLS ID will be used in place of another attribute in the control. See the individual control descriptions for the attribute whose value is substituted by nlsid.
style	A CSS rule enclosed in quotation marks, for example: style="{COLOR: darkmagenta; TEXT-DECORATION: underline}"
tooltip	Sets the string that will be displayed on browser mouseover
tooltipnlsid	Sets the NLS key to look up a localized tooltip string.
tooltipdatafield	Sets the datafield that overrides the value of tooltip and tooltipnlsid.
visible	Sets whether the control is rendered as visible or hidden. Some controls do not implement visibility.

## Event attributes

Many controls accept user input. The user input is submitted by a control event. The value of the event attribute corresponds to the name of an event handler. The event is generally handled in the calling component class unless the control has the `runatclient` attribute set to true. To find the exact event attributes that are supported by a control, consult the tag library descriptor (\*.tld file) for that control.

**Table 2-2. Event attributes**

Attribute	Description
<code>onchange</code>	Sets the event that is fired when the control is changed by the user. The <code>onchange</code> event handler cannot run on the client ( <code>runatclient</code> cannot be true). The <code>onchange</code> event is not handled immediately. It is handled on the server only when the form is submitted (for example, by an <code>onclick</code> event). Some controls do not implement an <code>onchange</code> event.
<code>onselect</code>	Sets the event that is fired when the user selects the control, such as an option in a list. The <code>onselect</code> event is handled immediately, either on the server (when <code>runatclient="false"</code> ) or on the client (when <code>runatclient="true"</code> ). Some controls do not implement an <code>onselect</code> event.
<code>runatclient</code>	Specifies that the event should run on the client, not the server. The <code>onchange</code> event cannot run on the client. Some controls do not implement a <code>runatclient</code> attribute.
<code>onclick</code>	Sets the event that is fired when the user clicks the control, such a button. The <code>onclick</code> event is handled immediately, either on the server (when <code>runatclient="false"</code> ) or on the client (when <code>runatclient="true"</code> ). Some controls do not implement an <code>onclick</code> event.

## Action attributes

The following attributes are common to most Documentum-enabled action controls. For the full list of attributes that are valid for a control, see the tag library descriptor that contains the control definition.

**Table 2-3. Common action control attributes**

Attribute	Description
action	Identifies the ID of the action to be launched by the control. This value matches the action id in the action definition.
showifinvalid	If true, the action item will be displayed but disabled (grayed out) if the configuration service cannot resolve the action definition. If false, the action will not be displayed in a menu if it is disabled for a selected object. Default = false. A value of true overrides the setting <code>visibility="false"</code>
showifdisabled	If true, the action will be displayed but grayed out if one or more of the action preconditions returns false.
disabledstyle	An entire CSS-syntax style sheet rule enclosed in quotes, applied to the disabled control
disabledclass	A CSS class to apply to the disabled control
oncomplete	The action associated with the action control. This attribute is not valid for action controls with a dynamic attribute.

## Dynamic action attribute values

The dynamic attribute can be used to specify when the action should be launched based on the conditions of other controls on the page. The JSP page must include the

JavaScript file `/wdk/include/dynamicAction.js`. The dynamic attribute can have one of the following values:

Attribute	Description
singleselect	The action can apply to only one selection or object.
multiselect	The action can apply to more than one selection or object (only in classic view or objectlist)
generic	The action can be launched regardless of selections by the user. The selections are essentially ignored.
genericnoselect	The action cannot be launched if any items on the page are selected.
false	(Default) The action is not dynamic.

**Note:** Dynamic action controls do not support an oncomplete attribute.

## 5.2.5 content transfer applet attributes

Applet controls for WDK 5.2.5 content transfer build several levels of attributes through applet class inheritance. The base applet control, `dmapplet`, has a basic set of attributes. The secure applet `secureapplet` adds another set of attributes related to security, and the `contentxferapplet` control adds content transfer-related attributes. Download and upload base applets add attributes specific to the direction of content transfer. Finally, each transfer applet can have unique attributes. All content transfer applets that are used in WDK5.2.5 JSP pages are documented in the general controls list that follows this section. The hierarchical set of applet attributes are documented in this section.

**Table 2-4. dmappletattributes**

Attribute	Description
align	Sets the horizontal alignment of applet text. The possible values of this attribute are the same as those for the HTML IMG tag: left, right, top, texttop, middle, absmiddle, baseline, bottom, absbottom.
archive	Jar or zip file that contains the content transfer code. When content transfer is used with Netscape 6.x, all the applets must have the same value for this attribute. Default = /wdk/contentXfer/ContentXfer.jar
code	Name of the file that contains the applet's compiled Applet subclass. This file is relative to the base URL of the applet. It cannot be absolute.
codebase	The base URL of the applet (directory or folder that contains the applet's code.) When content transfer is used with Netscape 6.x, all applets must have the same value for this attribute. Default = /wdk/contentXfer/
height	Applet height in the browser. Set to 0 for hidden applet.
id	ID is generated by the framework

**Table 2-5. secureapplet attributes**

Attribute	Description
errorhandler	Name of JavaScript error handler function. Default = onContentTransferAppletError()
errorhandlerarg	Argument that is passed to the errorhandler function



Attribute	Description
executionmgr classname	A fully qualified class name that implements com.documentum.web.contentxfer.applet.ExecutionMgr. Defaults: com.documentum.web.contentxfer.applet.NetscapeExecutionMgr, com.documentum.web.contentxfer.applet.MicrosoftExecutionMgr
locale	User locale. This is overridden by the user's selection.
successhandler	Server side event handler that handles the applet return.
successhandlerarg	Argument that is passed to the successhandler function
encoderesult	Set to true to encode the applet result in base64. Default = false

**Table 2-6. contentxferapplet attributes**

Attribute	Description
serviceurl	URL of the page that will receive the content from the applet
contentlocationunix	Path to default location for temporary viewed content on UNIX and Mac clients
contentlocationwindows	Path to default location for temporary viewed content on Windows clients.
registrylocationunix	Location of the registry file on UNIX and Mac systems.
buffersize	Size of the buffer used for transferring files
serverversion	The version of the WDK server that the client is connecting to
showprogress	If true, displays a progress bar in a separate window for each transfer
debug	If true, runs in debug mode. Does not delete temporary content in the send and receive directories on the client.

Attribute	Description
progressmonitorclassname	The progress monitor for the content transfer
contentxferlisteners	Comma-separated list of listener class names to be instantiated by the applet
checkoutlocationunix	Default checkout location on UNIX systems
checkoutlocationwindows	Default checkout location on Windows systems. This value will be used if there is no setting from another Documentum client in the registry.

Table 2-7. downloadcontentapplet attributes

Attribute	Description
contentticket	A ticket representing the location on a web server of content to be downloaded
idstoretrieve	Comma-separated list of object IDs to retrieve the content for
expand	Determines whether the archive should be expanded on the client.
enableuniquedircreation	If true, applet creates a unique directory for each operation. If false, all checked out content goes to the send, receive, or checkout directory. Default = false. The value of this attribute be the same on both edit.jsp and checkoutDownload.jsp.
enablesubdircreation	Not implemented

Table 2-8. uploadcontentapplet attributes

Attribute	Description
linkinstructions	Absolute path to the link instructions file

## Runtime tag attribute override

Some attribute values can be overridden by another attribute. This allows the source of a value to come from the JSP page, a localized set of strings, a datafield, or a value based on user context. For example, the datafield attribute is commonly used to override a label or value attribute.

The tag value that will be displayed is evaluated at run time. If more than one of the following attributes are available for a control, the hierarchy of overrides is as follows:

- value or label attribute sets the value
- nlsid overrides the value
- datafield overrides the value and nlsid
- contextvalue overrides the datafield, nlsid, and value

## HTML controls

The tag library `dmlayout_1_0.tld` contains several layout tags that generate HTML. Configurable tags have the same attributes as their HTML counterparts. See the tag library for a list of attributes for each configurable tag. Consult an HTML reference for the usage of these attributes.

**Table 2-9. HTML layout controls**

Control tag	HTML generated	Configurable
html	<html>	No
head	<head>	No
title	<title>	No
body	<body>	Yes. Attributes <code>onload</code> and <code>onunload</code> specify the event handler for the page load or unload event.
table	<table>	Yes
tr	<tr>	Yes
td	<td>	Yes
span	<span>	Yes
div	<div>	Yes
th	<td>	Yes

## absolutefilepathvalidator

**Purpose**       Validates a string file path using DFC utilities

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

controltovalidate	(Required) The name of the control to validate
nlsid	ID of NLS text to use for validation failure ( overrides errormessage)
errormessage	The error message for validation failure

---

## aclclassvalueformatter

**Purpose**       Formats the acl\_class attribute value on an ACL from an int to a string: 0.0 = regular  
| 1.0 = template | 2.0 = template instanance | 3.0 = public

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

This formatter has no configurable attributes.

# actionbutton

**Purpose** Generates a button that has an associated action. The button is not shown if the action cannot be resolved. The button is displayed as disabled if the action cannot be executed. Use the <argument> tag to set context.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

align	(Image action buttons only) Sets the alignment of the button text (default = center)
label	Button label text
datafield	Overrides the label value with a value from a data source
nlsid	ID of NLS text to use as the label value (overrides datafield and label)
width	(Image action buttons only) Sets the width of the button
height	(Image action buttons only) Sets the height of the button
imagefolder	The folder for the button graphics. If the appropriately named graphics are not found in this folder, an HTML button will be drawn.
align	(Image action buttons only) Sets the alignment of the button text (default = center)

---

## actionbuttonlist

**Purpose** Generates a type-based list of action buttons. The actions that will be displayed are configured in the <actionlist> tag of action Configuration files (dm\_ *typename\_action.xml*).

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

datafield	Overrides the scope field. For example, it can be used to display only a certain type of objects in the actionlist on a per row basis in a datagrid.
separatorhtml	Optional HTML string that will be output between each action control for formatting.
listid	Identifies the ID of the actionlist to use.
oncomplete	Identifies an event to be fired when the action has completed.

---

## actionimage

**Purpose** Generates an image that launches an action when selected

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

align	(Image action buttons only) Sets the alignment of the button text (default = center)
label	Optional label to be rendered in addition to the image
datafield	Overrides the label value with a value from a data source
nlsid	ID of NLS text to use as the label value (overrides datafield and label)
width	(Image action buttons only) Sets the width of the button
height	(Image action buttons only) Sets the height of the button
src	URL to the image
srcdisabled	Image to be used when the action is disabled
datafield	Overrides the URL defined in src attribute
datafielddisabled	Overrides srcdisabled value
alttext	Text that will be displayed by mouse hover
action	Specifies the ID of an action to be launched by the link

## actionlink

**Purpose** Generates an HTML link that has an associated action. The link is not shown if the action cannot be resolved. The link is shown as disabled if the action cannot be executed. Use the <argument> tag to set context.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

An optional image can be shown with the link. Most of the attributes below configure this image.

---

align	(Image action buttons only) Sets the alignment of the button text (default = center)
label	Link text
src	(optional) URL to the image
srcdisabled	(optional) Image to be used when the action is disabled
datafield	Overrides label or NLS value with string from a data source
datafielddisabled	Overrides srcdisabled value
alttext	Text that will be displayed by mouse hover
nlsid	ID of NLS text to use (overrides label)
width	(Image action buttons only) Sets the width of the button
height	(Image action buttons only) Sets the height of the button
showlabel	Set to true to show the label
showimage	Set to true to show an image with the link
action	Specifies the ID of an action to be launched by the link
linkspacer	Specifies width in pixels to display as space between dynamic links. Space is collapsed when control is not visible.

---

## actionlinklist

### Purpose

Generates a list of action controls based on type. The actions that are displayed are configured in the <actionlist> element of action Configuration files (dm\_<typename>\_action.xml). You can use more than one actionlinklist control with the



same scope in a JSP page, provided the listid attribute has a different value. This allows you to display different actions for the same object type in different screens.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

datafield	Overrides the scope field. For example, it can be used to display only a certain type of objects in the actionlist on a per row basis in a datagrid.
separatorhtml	Optional HTML string that will be output between each action control for formatting.
listid	Identifies the ID of the actionlist to use.
oncomplete	Identifies an event to be fired when the action has completed.
columns	Specifies the number of columns in which to display the list, in a table
separatorhtml	Optional HTML string that will be output between each action control for formatting purposes.
listid	The ID of the list to use within the scope. If not specified then the value of "default" is used.
showlabels	Set to false to override labels set in component XML definition
showimages	Set to false to override images set in component XML definition

## actionmenuitem

**Purpose** Generates a menu item that launches an action. The actionmenuitem tag has all of the same attributes as actionlink, except the label attribute in actionlink is the value attribute in actionmenuitem. The additional attributes, or differences in attributes with the same name, are shown below.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

value	Menu item string
datafield	Overrides menu item string or nlsid from a data source
nlsid	ID of NLS text to use (overrides value)
imageicon	URL to an image to be displayed as a menu item

---

## actionmultiselect

**Purpose** Supports multiple selection with an associated action. Contains action controls with the dynamic attribute, such as actionmultiselectcheckbox or actionlink. The action on multiply selected items is specified by either the contained action control or by an actionmenuitem control. The actionmultiselect control passes nested arguments (in <argument> tags) to the control that performs the action. Multiple selection is supported only in classic or objectlist view.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

selectionargs	Comma-separated list of arguments that are used to uniquely identify selected items
selectiongroupargs	Comma-separated list of arguments that are used to uniquely identify an item's group. Actions are disabled when the selected item's selectiongroup attribute(s) do not share the same value.

## Usage

Use the optional selectionargs attribute to identify each selected item in the datagrid. This will ensure that the same items are selected when a component returns. The following example from searchresultslist.jsp adds arguments for a selection group, which ensures that the action cannot be performed unless the multiple objects that are selected share the same repository:

```
<dmfx:actionmultiselect name="multi" selectionargs="
  objectId,<%=SearchInfo.ENTRYID_PARAM%>,<%=SearchInfo.
  QUERYID_PARAM%>" selectiongroupargs="
  <%=SearchResultSet.ATTR_DOCBASE_ID%>">
  <dmfx:argument name='objectId' contextvalue='objectId' />
  ...
</dmfx:actionmultiselect>
```

# actionmultiselectcheckall

**Purpose** Generates a checkbox that checks all objects in a list. When checked, an action can be performed on all objects. In accessibility mode, this control generates a link to a global actions page, which displays actions that do not require a selected object.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

**Attributes**

None

## actionmultiselectcheckbox

**Purpose** Generates a checkbox that has one or more associated actions. Must be contained within an `<actionmultiselect>` tag in a datagrid header. All action controls on the page with the dynamic value set to multiselect are associated with the checkbox. In accessibility mode, this control generates a link to an actions page for the object.

**Tag library descriptor**

dmfx (dmformext\_1\_0.tld)

**Attributes**

None

## adminrequiredfieldvalidator

**Purpose** Extends [requiredfieldvalidator](#), page 154 and validates a label control to ensure that the "none selected" option has not been selected.

**Tag library descriptor**

dmfx (dmformext\_1\_0.tld)

## Attributes

controltovalidate	(Required) The control that is validated by this validator
errormessage	The errormessage to be displayed when the validation returns false
indicator	Specifies an indicator to display next to a required field. If empty (indicator = ""), an asterisk will be displayed (&nbsp;*).
indicatornlsid	Specifies an NLSID that resolves a string to be displayed as a required field indicator
indicatorstyle	Specifies a CSS style rule that should be applied to the indicator
indicatorcssclass	Specifies a class in a style sheet that should be applied to the indicator

## argument

**Purpose** Generates arguments that are passed to a control's event handler. The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

value	Argument value
datafield	Overrides value with data from a data source. If the datafield is not available, a null value is used..
nlsid	ID of NLS key to look up a string (overrides value)

## argument

**Purpose** Generates arguments that are passed to the control that contains argument tags

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

value	Argument value
datafield	Overrides the argument value and nlsid and sets the value from a datafield. If the datafield is not available, a null value is used.
contextvalue	Overrides the datafield, nlsid, and value
nlsid	ID of NLS string that represents the argument (overrides value)

---

## attributeselector

**Purpose** Displays a list of attributes from which the user can compose and order items. The items are either a list passed into the control or the attributes of a selected object type. Control state is manipulated on the client and passed to the server via a hidden field when the page is submitted. This control is used in the columnselector component to select attribute columns for display.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

docbasetypeslabel	Label to be displayed for the Docbase type dropdown list
docbasetypesnlsid	NLS string to be displayed as label for the Docbase type dropdown list
items	A serialized list of items to appear in the available items listbox. This is supplied when the user selects an object type or by the component class. The default list before selection is provided by the component definition, in the <columns> element.
itemslabel	Label for the available items list
itemsnlsid	NLS string for the available items list label
buttoncssclass	CSS class to apply to button appearance
buttoncssstyle	CSS rule to apply to button appearance
buttonimagefolder	Specifies the location of the image folder if buttons are used to display items
buttonwidth	Width of the button in pixels
buttonheight	Height of the button in pixels
disabledclass	CSS class to apply to a disabled item
disabledstyle	CSS rule to apply to a disabled item
listboxcssclass	CSS class to apply to listboxes
listboxcssstyle	CSS rule to apply to listboxes
listboxcsswidth	Width of listboxes in pixels. Default = 200.
selecteditems	A serialized list of items to appear in the selected items listbox
selecteditemslabel	Label for the selected items list
selecteditemsnlsid	NLS string for the selected items list label
showlabels	Set to true to displays data dictionary labels for Docbase types and attributes. Set to false to display actual type names and attribute names.
size	The size of the listboxes. Default = 10.

orderingenabled	Set to true to enable to <b>Up</b> and <b>Down</b> buttons that change the order of the list. Default = true
basetype	Specifies the base type to search from when querying for the list of types to fill the Docbase Types dropdown list. Defaults = dm_sysobject

---

## Usage

A serialized list of items can be an array of one string (value) or two strings (value and label). By default, the list of items to be displayed is provided by the component definition, in the columns element. For example, the drilldown component definition specifies that the object\_name attribute will be displayed but the title attribute will not, in the following except:

```
<columns>
  <!-- <loadinvisibleattribute>true</loadinvisibleattribute> -->
  <column>
    <attribute>object_name</attribute>
    <label><nlsid>MSG_NAME</nlsid></label>
    <visible>true</visible>
  </column>
  <column>
    <attribute>title</attribute>
    <label><nlsid>MSG_TITLE</nlsid></label>
    <visible>false</visible>
  </column>
  ...
</columns>
```

## bannerbox

**Purpose** Generates a box containing the Collaborative Edition room banner. (Contains a roombanner tag on the JSP page.)

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)



## Attributes

This component has no specific configurable attributes.

# body

**Purpose** Generates an HTML <body> tag.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

The following HTML <body> tag attributes can be set on the JSP body tag: vlink, title, text, style, link, lang, dir, bgcolor, background, alink, marginheight, marginwidth, topmargin, bottommargin, leftmargin, rightmargin, onload. For descriptions of the usage of these attributes, consult an HTML reference. Additional (non-HTML) attributes are interpreted by the WDK framework.

---

onload	Specifies the JavaScript event handler for the HTML body onload event.
onunload	Specifies the JavaScript event handler for the HTML body onunload event.
showdialogevent	Set to true to allow the page to be rendered in an Application Connector dialog window
sizepreference	Set to true to save the user 's resize preference for an Application Connector dialog window. Default = true

---

# bookmarklink

**Purpose** Generates an icon link that can be saved on the desktop or in the browser to launch the same page in the application. The link is generated in the form of a URL to the DRL (document resource locator) component, similar to the following:

## Tag library descriptor

```
dmf (dmform_1_0.tld)
```

## Attributes

```
http(s):// my_server/ webtop/drl/  
  objectId/ 090000018024e156/format/ xml/  
  versionLabel/ CURRENT
```

---

label	Link text
href	Target of the link. Default target if no href value specified is drl component.
datafield	Link text, overrides nlsid and label
nlsid	ID of NLS key to look up the string for the link text, overrides label
runatclient	Behavior unknown

---

## Usage

The default behavior of the DRL component is to view the passed content or folder object using the action `drlview`. If the object type has a defined action `drlview`, the user is presented with a dialog to select the view or edit action. If the object is versioned, the user is presented with a dialog to choose the current version instead of the version represented by the DRL.

The DRL component class provides a helper method that creates a well-formed DRL. The method `constructDRL()` takes an `objectId` and optional `format` and `versionLabel` parameters and returns a URL.

The DRL syntax has changed in 5.2.5 to support email programs that cannot contain certain characters such as `?`, `&`, and `=`. To support the DRL syntax of all versions of WDK,

a DRL dispatcher converts the new DRL form to the previous URL form. The URL form of the DRL is the following:

```
http(s):// my_server/ webtop/drl.html?
  objectId= 090000018024e156&format= xml&
  versionLabel= CURRENT
```

## booleanformatter

**Purpose** Wraps a boolean value with a formatted output, such as True, T, False, or F

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

truenlsid	ID of NLS key to look up the string for true
falsenlsid	ID of NLS key to look up the string for false

## breadcrumb

**Purpose** Breadcrumb (web—style path navigation). The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

value	Current path displayed by the breadcrumb
displayleaf	Sets whether to display leaf cells when they are selected
separator	Sets the character to separate each path element
onclick	(Required) Event fired when the control has been clicked
onchange	Server-side event that is fired when the user clicks on the breadcrumb. This event is fired after the form is submitted.
runatclient	Set to true to handle the onclick event on the client
leafcssclass	Specifies the CSS class to display a breadcrumb leaf

## Usage

Global configuration for all instances of the breadcrumb control is done in `wdk/config/breadcrumbcontrol_config.xml`. The following elements can be configured:

Element	Description
<code>&lt;breadcrumbcontrolconfig&gt;</code>	Contains configuration elements
<code>&lt;displayleaf&gt;</code>	Specifies whether a breadcrumb control should show its leaf (current folder)
<code>&lt;leafcssclass&gt;</code>	Specifies a CSS class to be used to display the leaf. If <code>&lt;displayleaf&gt;</code> is set to true and no value is supplied for <code>&lt;leafcssclass&gt;</code> , the breadcrumb class will be used to display the leaf.

---

## browserrequirements

**Purpose** Looks up the supported IE and Netscape versions as defined in the <browserrequirements> element of app.xml. If the browser is not supported, is on an unsupported platform, or has no JVM, an error message is displayed.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

This control has no configurable attributes.

## button

**Purpose** Generates an HTML button. To generate an image button, the imageFolder attribute must be set with a valid folder name in a theme directory containing the image.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

label	Value stored in the control and displayed as text on the button
datafield	Overrides label or nlsid with data from a data source
nlsid	ID of NLS text to use (overrides label)

disabledstyle	An entire CSS-syntax style sheet rule enclosed in quotes (applied when button is disabled)
disabledclass	A CSS class to apply to the control (applied when button is disabled).
default	Sets whether this button is the default button on the page. Must be used with a text control or other control that has defaultonenter attribute set to true. When the user hits the <b>Enter</b> key, the default button event will be fired.
width	(Image buttons only) Specifies the button's width
height	(Image buttons only) Specifies the button's height
imagefolder	Sets the folder that the button graphics are stored in. If graphics are not found in this folder, an HTML button will be drawn.
align	(Image label only) Alignment of the label text
onclick	(Required) Event fired when the control has been clicked
src	Image to be used for the button
srcdisabled	Image to be used when the button is disabled

---

## Usage

Button labels are specified in the `nlsid` attribute, which overrides a label attribute value. To change a label for a specific control in a JSP page, you can add a label attribute and value and remove the `nlsid` attribute. The following example overrides the `nlsid` label string by removing the `nlsid` attribute and adding a label attribute:

```
<dmf:button cssclass='buttonLink' label='Finished'.../>
```

If your button text will be localized, you must use the `nlsid` attribute and add a localized properties file for the button.

To use a different image for the button, specify the path to the image in the `imagefolder` attribute values. The `imagefolder` points to the directory that contains the specific

button images, which must be named `left.gif`, `right.gif`, and `bg.gif`. The branding system appends the theme name to the base directory path. The following example points to new images in our custom theme directory:

```
<dmf:button ...imagefolder='images/finishbutton' />
```

You can override specific properties of the CSS class in the style attribute for a button. In the following example, the disabled class is retained and only the font-weight is overridden:

```
<dmfx:button name='checkout' action='checkout' label='Files'
  cssclass="buttonLink" disabledclass="buttonDisabledLink"
  style="font-weight:normal" ...>
```

To change the style of a button or image everywhere it appears in your application, apply your changes in a style sheet. For example, the `webcomponents.css` file in the coolblue theme defines a `drilldownHeader` class that references a background image:

```
.drilldownHeader { BACKGROUND-COLOR: transparent;
  BACKGROUND-IMAGE: url('../images/streamline/tabbarbg.gif') }
```

You would extend the coolblue theme, copy the `webcomponent.css` file to your `/theme/css` directory, and change the URL to the background image.

## cellist

**Purpose** Used within a table row (`<tr>`) or `<dmf:datagridrow>` tag to wrap a list of celltemplate tags. The cell templates will output columns of data dynamically.

### Tag library descriptor

```
dmf (dmform_1_0.tld)
```

### Attributes

---

fields	List of fields that correspond to attribute names in the repository. This list is overridden by a list of columns in the component XML definition.
labels	Comma-separated string of fields to be displayed as labels. The values are

nlsids	overridden by labels set on columns in the component definition file. Comma-separated string of NLS IDs to display. The values are overridden by labels set on columns in the component definition file.
hascustomattr	Set to true if the celllist fields contain custom attributes and the celllist tag is used within a datasortlink tag. This will ensure that the link is displayed correctly for sorting.
enabled	Set to false to display the datafield as a non-sortable column heading. Default=true.

---

## Usage

If you specify columns in the component definition, they will override the fields that are specified

To set the celllist fields programmatically, call the CellList method setFields() during onControlInitialized.

## celltemplate

**Purpose** Within a cell list tag, specifies a template output for a single column of data. The data that is rendered is supplied by the fields attribute. Template values are supplied by a field name, such as object\_name, object type, such as double, or generic (matches everything).

### Tag library descriptor

dmf (dmform\_1\_0.tld)



## Attributes

field	Name of a field corresponding to an attribute name. If field is specified, type attribute cannot be used.
type	Documentum object type. If type attribute is specified, field attribute cannot be used. If neither field nor type attributes are set, then cell is generic. The number type matches either an int or double type returned from DFC.

## Usage

Use within a `<dmf:celllist>` tag. The fields for the template can be supplied as the value of the fields attribute on the JSP page or programmatically from the component configuration using the `ComponentColumnDescriptorList` object. WDK components that have a `<columns>` element in the component XML definition get the list of fields from the configuration. (They get the columns from `ComponentColumnDescriptor`.)

Fields, labels, and nlsids that are configured in the component definition file override tag attributes in the JSP page.

You can configure dynamic sets of data for display using `<dmf:celllist>` and `<dmf:celltemplate>`. If you are displaying the data in columns, you can use a combination of celllist and celltemplate controls within a table header row or within `<dmf:datagridrow>` to provide column headers, sorting, and columns of data. See *Web Development Kit and Client Applications Development Guide* for details.

# checkbox

**Purpose** Generates an HTML checkbox form control. The checkbox supports the onclick event for which you must implement your own server side event handler. Checkbox also supports an onchange event which is fired when the form is resubmitted.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

value	The boolean state of the control (true = checked, false = unchecked). Default = false.
datafield	Overrides value with data from a data source
nlsid	ID of NLS text to use (overrides label)
label	Text label for the checkbox
labelalign	Set alignment of the checkbox label: left or right
onclick	Event fired when the control has been clicked
onchange	Server-side event fired when the form is submitted
runatclient	Set to true to handle the onclick event on the client

---

# checkjvmapplet

**Purpose** Generates an applet that checks the client browser JVM and offers the option of installing the Sun VM if it is not installed. This control is used by applications or components that must use the Sun VM. The applet is used by the Web workflow manager component.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

See [5.2.5 content transfer applet attributes, page 15](#) for attributes.

In addition to applet attributes that can be set on any applet, the following special attribute is defined:

---

successhandler	Server side event handler that handles the applet return.
----------------	---

---

## clientenvpanel

**Purpose** Generates a panel that is displayed or hidden depending on the client environment. For example, the WDK login JSP page has a panel for the Application Connectors environment and one for the non-connector environment.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

environment	Specifies the client environment. Valid values: appintg. To set the environment for browsers that do not have a Application Connector installed, set the reversevisible attribute to true.
reversevisible	Sets the contents of the panel to be visible when the environment is not the specified environment

---

### Usage

Your application can display different content depending on whether the client is using a Application Connector within a Microsoft Office application. For example, the WDK login page contains two login buttons. The button for the non-connector environment specifies a server-side event handler as follows:

```
<dmfx:clientenvpanel environment="appintg" reversevisible="
  true">
  <dmf:button ...onclick="onLogin"/>
</dmfx:clientenvpanel>
```

The button for the connector environment specifies an event handler on the client opens a browser window for the application connector:

```
<dmfx:clientenvpanel environment="appintg">
  <script>
    function onLogin()
    {
      try
      {
        window.external.AppIntgProcessEvent("event=OnLogin");
      }
      ...
      postServerEvent(null, null, null, "onLogin");
    }
  </script>
</dmfx:clientenvpanel>
...
<dmfx:clientenvpanel environment="appintg">
  <dmf:button ..onclick="onLogin" runatclient="true"/>
</dmfx:clientenvpanel>
```

## columnlist

**Purpose** Used by the preference framework when a preference definition has the type "columnlist." Generates two controls for each preference: a label for currently selected columns and a link that nests to the preference editing component. This control is designed to be used within the preference control and should not be used independently.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

value	Overrides value of selectedcolumns attribute
datafield	Overrides value and selectedcolumns attributes
columns	A serialized list of items to populate the available items list box. See multiselector control ( <a href="#">multiselector, page 130</a> ) for details on the list.

---

selectedcolumns	A serialized list of items to populate the selected columns list box.
defaultselectedcolumns	List of default selected columns that the user can reset to if desired
namelabel	Name of the column list. Used by the Column Selector for the dialog title.
namenlsid	NLS key to label for column list
editcomponent	Component for editing the preference that is nested to by the <b>Edit</b> link in the preferences UI. Default = columnselector component
editcontainer	Container that is nested to (contains the editcomponent) by the <b>Edit</b> link in the preferences UI. Default = dialogcontainer
columnlabel	String label for columns list in editing component
columnnlsid	NLS ID that overrides columnlabel
columnspname	Parameter for custom editing component. Default value: columns
selectedcolumnspname	Parameter for custom editing component. Default value: selectedcolumns
defaultselectedcolumnspname	Parameter for custom editing component. Default value: defaultSelectedColumns
basetype	Base Docbase type for attribute columns. Default: dm_sysobject. This attribute value is set by the <columnlist> element in a component definition <preferences> section.
inherits	Set to true to specify that the <b>Inherits</b> checkbox is initially checked. For the checkbox to display, a valid inheriting preference must be displayed. If the checkbox is then checked, the columns from the inheriting preference indicated are used

inheritingpreference	Specifies the component from which to inherit the preferred columns by default
inheritinglabel	Specifies the label to display with the <b>Inherits</b> checkbox

---

## colorpicker

**Purpose** Displays a colored bar that allows user to select a color value from 216 colors. The colorpicker GIF is located in images/color/colorpicker.gif. This control's attributes are set by the Media Transformation Services component transformationdetails. The component has settings for the colorpicker control attributes that override the defaults shown below.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

width	Sets width of colorpicker display. Default = 100 pixels
height	Sets height of colorpicker display. Default = 20 pixels
colorvalue	Sets the initial RGB color value before selection, for example, 76fa34. Default = null
numberofrows	Sets the number of rows in colorpicker. Default = 12

---

---

## combocontainerpanel

**Purpose** Generates a panel to display progress

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

progressbarid	Must correspond to the name of a dmfx:progressbar control on the page
---------------	---

---

### Usage

This control should be used to wrap a <dmfx:containerinclude/> control in order to display a progress bar. The JSP page must also contain a progressbar control.

## columnpanel

**Purpose** Wraps a set of column controls in a table row to allow display or hiding of column data

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

columnname	(Required) Sets the name of this column. All column panels in a single row should use unique names.
------------	---

---

## commentrow

**Purpose** Renders the individual comments for a topic. Must be contained within a topic control.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

indentamount	Specifies the indent amount in pixels for child comments. Default = 32
cssclass	The CSS class to be used for previously read comments
unreadstyle	Sets the CSS style to be used for unread comments. By default it uses defaultUnreadCommentStyle

---

## commentimage

**Purpose** Render a comment **Edit**, **Reply** or **Delete** button.



## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

label	Value stored in the control and displayed as text on the button
datafield	Overrides label or nlsid with data from a data source
nlsid	ID of NLS text to use (overrides label)
disabledstyle	An entire CSS-syntax style sheet rule enclosed in quotes (applied when button is disabled)
disabledclass	A CSS class to apply to the control (applied when button is disabled).
default	Sets whether this button is the default button on the page. Must be used with a text control or other control that has defaultonenter attribute set to true. When the user hits the <b>Enter</b> key, the default button event will be fired.
width	(Image buttons only) Specifies the button's width
height	(Image buttons only) Specifies the button's height
onclick	(Required) Event fired when the control has been clicked
src	Image to be used for the button
type	Specifies the edit mode. Valid values: edit   reply   delete

## Usage

The control takes two arguments that are passed in using argument controls. The control must be contained within the commentrow control. Use the argument list to pass values to the onclick event.

```
<dmfx:topic name="<%=TopicPageView.CONTROL_TOPIC%>">
```

```
<dmfx:comment name="comment_row" cssclass="defaultCommentStyle"
  indentamount="32">
  . . .
  . . .
</dmfx:comment>
</dmfx:topic>
```

## comparevalidator

**Purpose** Validates an input control by comparing its value to a specified value or to an input control. The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

controlovalidate	(Required) The control that is validated by this validator
errormessage	The errormessage to be displayed when the validation returns false
indicator	Specifies an indicator to display next to a required field. If empty (indicator = ""), an asterisk will be displayed (&nbsp;*).
indicatornlsid	Specifies an NLSID that resolves a string to be displayed as a required field indicator
indicatorstyle	Specifies a CSS style rule that should be applied to the indicator
indicatorcssclass	Specifies a class in a style sheet that should be applied to the indicator
type	Type of value: string or integer
operator	Comparison operator. Valid values are: equal, notequal, greaterthan, greaterthanequal, lessthan, lessthanequal

---

comparevalue	Value to compare
comparecontrol	Name of input control to compare. Overrides comparevalue

---

## Usage

The comparison validator checks the value of one control to the value of another. The following example compares the passwords entered in two fields:

```
<dmf:password id='ChangePassword' name='ChangePassword' size='40'
  defaulttonenter='true'/>
<dmf:password id='ConfirmPassword' name='ConfirmPassword' size='40'
  defaulttonenter='true'/>
<dmf:comparevalidator name="validator" controltovalidate="ConfirmPassword"
  type="string" operator="equal" comparecontrol="ChangePassword"
  nlsid="MSG_CONFIRMPASSWORD_NOTEQUAL"/>
```

# componentinclude

**Purpose** Includes a component within a form. If the component to be included is a container, you must use containerinclude instead of componentinclude.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

component	ID of component to include
page	Component start page name

---

## componenturl

**Purpose** Component navigation URL (link). Use the argument tag to pass arguments and set the context.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

component	The target component to be displayed.
page	Component page to display
target	Name of target frame

---

## containerinclude

**Purpose** Used in a container JSP as a placeholder for the contained components. The contained components are specified in the container XML definition file.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

None

# datadropdownlist

**Purpose** Generates an HTML drop down list control. Holds option tags or a single dataoptionlist control that contains a databound option tag.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

value	The selected value.
width	Horizontal length of the drop down list control
height	Vertical length of the drop down list control
onselect	Event fired when the control has been clicked on. The form is not submitted until the onselect event is fired.
query	(optional) Displays query or recordset as a drop down list. Use dataoptionlist tag within datadropdownlist for databound values. Use option tag within datadropdownlist for static output. The DQL query that is passed to the data provider has the following form: query="select r_object_id, user_name from dm_user where user_name like 'a%' "
onchange	Specifies a server-side event handler after the form has been submitted
runatclient	Set to true to handle the onselect event on the client

# datagrid

**Purpose** Displays results of a query or recordset as a table, generating opening and closing <table> tags. Can contain datagridrow, nodatarow, row, or HTML table row (<tr>) tags. The data can be paged by adding a datapaging control and sorted by adding a datasortimage or datasortlink control into the grid control. The name attribute is required.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

cellspacing	Spacing between the cells in the grid
cellpadding	Space between cell contents and cell border
width	Width of the datagrid
height	Height of the datagrid
bordersize	Size of the datagrid's border
query	DQL query that is passed to the data provider, for example: "select group_name, r_modify_date from dm_group where group_name like 'a%' "
paged	States whether paging is turned on. Default = false (no paging, all data returned).
pagesize	Sets the number of result items in a page when paging is turned on (paged='true')
recordcachecount	Overrides the default number of records retrieved by the query and sets the number of records to be fetched
sortcolumn	Name of column to sort on

---

sortdir	Sets the sort direction (ascending or descending). (IDataboundParams public constants): 0 (forward), 1 (reverse)
sortmode	Initial sort mode to use (IDataboundParams public constants): 0 (text), 1 (case-insensitive text), 2 (numeric)
preservesort	Sets whether the datagrid should preserve the current sort settings between data refresh events. True to preserve sort settings. (Default = false)
rendertable	Set to false to suppress the render of <table> tags by the datagrid control. When the data rows are rendered within separate panes within a paneset control, set this value to false and insert <table> tags manually within the pane control. Default = true.

---

## datagridRow

**Purpose** Generates rows of data inside a datagrid. Generates <tr> opening and closing tags within the <table> tags generated by datagrid. Can contain <td>, <columnpanel>. Displays the contents of its databound controls for all rows of data that are available or that are shown per page.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

cellspacing	Spacing between the cells in the grid
cellpadding	Space between cell contents and cell border
width	Width of the datagrid

height	Height of the datagrid
bordersize	Size of the datagrid's border
query	DQL query that is passed to the data provider, for example: "select group_name, r_modify_date from dm_group where group_name like 'a%' "
paged	States whether paging is turned on. Default = false (no paging, all data returned).
pagesize	Sets the number of result items in a page when paging is turned on (paged='true')
columns	The number of columns to display the data. If this value is greater than zero, then the datagrid will format the data in the specified number of columns.
align	Horizontal alignment of the cell
valign	Vertical alignment of the cell
altstyle	CSS rule for alternating lines if required, for example: <dmf:datagridRow style='{BACKGROUND-COLOR: white}' altstyle='{BACKGROUND-COLOR: red}'>
altclass	Sets the class for alternating lines if required, for example: altclass='databoundExampleDatagridRowAlt'
tooltip, tooltipnsid, tooltipdatafield	String, NLSID element, or datafield that provides a title for each row.

---

## Usage

The number of rows displayed depends on data set size and paging options set on the parent datagrid tag.

When the accessibility option is turned on in the user preferences, the datagridrow tag generates a title for the row based on the tooltip. This allows reader programs to navigate the generated rows of data.



**Caution:** Use only one datagridRow tag within a datagrid.



# datalistbox

**Purpose** Generates an HTML list box control. Use the dataoptionlist tag within dataoptionlist for databound values. Use the option tag within dataoptionlist for static output.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

value	The selected value.
width	Horizontal length of the drop down list control
height	Vertical length of the drop down list control
onchange	Specifies the server-side event that is fired when the value has changed after the form is submitted
onselect	Event fired when the control has been clicked on. The form is not submitted until the onselect event is fired.
size	The number of rows of selectable data visible in the listbox. Scrollbars are used if more rows are available.
query	DQL query to retrieve data for a dataoptionlist control, for example: for example: query="select r_object_id, user_name from dm_user"
runatclient	Set to true to handle the onselect event on the client

---

## dataoptionlist

**Purpose** Iterator that renders options as a HTML form option elements. Renders all of the data from the enclosing datadropdownlist or datalistbox.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

None

## datapagesize

**Purpose** Generates a page size control that allows the user to select the number of items in a page. The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

pagesizevalues	Comma-separated list of integer page-size values that the user can select
onselect	Event that is fired when the user selects a page size value

---

preference	Sets a default value for the preferred data page size that can be overridden by the user. The value can be read from a configuration file such as /custom/app.xml as the content of a dot-separated path of elements in the file. For example, preference='application.display.streamline' would be read from the following app.xml element path: <pre>&lt;application&gt; ... &lt;display&gt; &lt;streamline&gt;10&lt;/streamline&gt; &lt;/display&gt; &lt;/application&gt;</pre>
showdisplayinginfo	Set to true to show the total item count
runatclient	Set to true to handle the onselect event on the client

---

## Usage

Cache size for the number of objects returned by a query is configurable in Databound.properties, in /WEB-INF/classes/com/documentum/web/form/control. Paging is configured on a JSP page that contains a datagrid. You can limit the choices for page sizes by setting the pagesizevalues of the datapagesize JSP tag. You can set a default value for page size as the value of the preference attribute on the datapagesize tag.

## datapaging

**Purpose** Generates navigation links for a paged datagrid. Must be used within the datagrid control. Links will be rendered only when multiple pages of data are returned from the query. The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

gotopageclass	A CSS class to apply to the optional gotopage text box.
showgotopage	Sets whether to show the optional gotopage text box. This allows the user to jump to any page in the dataset by entering a page value in the text box.
showdisplayinginfo	Sets whether to show the "displaying X of Y items" information.
showfirstandlast	Sets whether to show the "first" and "last" page links.
showimages	Specifies that images instead of text links should be displayed. The images should be placed in: <code>/wdk/ theme_folder /images/paging</code> . Default = true. If the images are not found, text links will be displayed.

---

## datasortimage

**Purpose** Generates a datasortimage control within the datagrid control. Provides an image that launches a sort of the dataset.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

The name attribute of the tag must be unique on the JSP page among the sorting columns.

---

label	Sets the label for the data sort link.
datafield	Overrides the label and nlsid with data from the current celltemplate match. The value of CURRENT for this attribute gets the value for the label from the data source.
nlsid	ID of NLS text to use (overrides label)
column	The column name to bind to. Can use datafield instead of this attribute if within a celltemplate tag.
image	(Required) Specifies the image file name. The path to the image is resolved by the branding service.
mode	The sort mode to use (IDataboundParams publicconstants): text, numeric, caseintext (case-insensitive text)
reversesort	Specifies the initial sort order of the displayed column. true = reverse, false (default) = normal.
forwardicon	URL to the forward sort icon.
reverseicon	Url to the reverse sort icon.
onclick	Event fired when the control has been clicked
runatclient	Set to true to handle the onclick event on the client

---

## datasortlink

**Purpose** Generates a datasortlink control within the datagrid control. Provides a column header sort link that allows the user to sort the results by column name.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

The name attribute is required and must be unique on the JSP page among the sorting columns.

---

label	Sets the label for the data sort link.
datafield	Overrides the label and nlsid with data from the current template match. The value of CURRENT for this attribute gets the value for the label from the data source.
nlsid	ID of NLS text to use (overrides label)
column	The column name to bind to. Can use datafield instead of this attribute if within a celltemplate tag.
mode	The sort mode to use (IDataboundParams publicconstants): text, numeric, caseintext (case-insensitive text)
reversesort	Specifies the initial sort order of the displayed column. true = reverse, false (default) = normal.
forwardicon	URL to the forward sort icon.
reverseicon	Url to the reverse sort icon.
onclick	Event fired when the control has been clicked
runatclient	Set to true to handle the onclick event on the client

---

## Usage

If the datasortlink contains a celllist tag whose contents are resolved at runtime, make sure that the celllist tag properly reflects the type of datafield providing the link. For example, if the datafield maps to a custom attribute, set the hascustomattr attribute on the celllist tag to true.

# dateinput

**Purpose** Generates a date input control. The order (day and month) will be determined by the user's locale. The months and days will be listed as integers and their labels are looked up in the localization bundle `form.control.DateInputNlsProp`. The name attribute is required.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

dateselector	Displays the date as dropdown fields (value = fields) or as a popup calendar (value = calendar). Default = calendar
dateformat	Specifies how the date is formatted in the UI and the control value. Valid values: short (for example, 1/2/02), medium (for example, 2 Jan, 2002), long (for example, 2 January, 2002)
value	The date value displayed by the control. The displayed value is formatted for the user's locale.
width	The width of the control. Default = 200 pixels.
day	Sets the default day to display. If not specified, 1 is selected.
month	Sets the default month to display. If not specified, 1 is selected.
year	Sets the default year to display. If not selected, the fromyear will be displayed.
fromyear	Sets the start year in the available date range. Default = 1997.

toyear	Sets the end year in the available date range. Default = 2005.
onselect	Event fired when the control has been clicked

---

## Usage

In addition to the configuration available on individual date controls, global date input display can be configured by extending `/wdk/config/datetimecontrol_config.xml`. The following table describes date input control configuration in this file:

**Table 2-10. Global date input control configuration elements**

---

<code>&lt;dateinput&gt;</code>	Configures all instances of <code>dateinput</code> tag including those generated by the <code>docbaseattributelist</code> control. Can be overridden on an individual tag. Contains <code>&lt;default-year-from&gt;</code> , <code>&lt;default-year-to&gt;</code> , and <code>&lt;default_type&gt;</code> .
<code>.&lt;default-year-from&gt;</code>	Sets the default beginning year in the date input dropdown list
<code>.&lt;default-year-to&gt;</code>	Sets the default end year in the date input dropdown list

---

Editable attributes are rendered as `dateinput` controls. To configure the rendered control, set the beginning and end year in the `<dateproperties>` element of `/wdk/app.xml`.

When you want to set a date programmatically in your component class, clear the `dateinput` control by calling `clear()`, which clears day, month, and year. For example:

```
private void resetDateInputControl(DateInput date)
{
    if (date != null)
    {
        date.clear();
    }
}
```



---

# datetime

**Purpose** Generates a date/time control that allows the user to select month, day, year, hour, minute, and second. The date and time labels are looked up in the localization bundle `form.control.DateTimeNlsProp`. The name attribute is required.

## Tag library descriptor

`dmf (dmform_1_0.tld)`

## Attributes

---

<code>dateselector</code>	Displays the date as dropdown fields (value = fields) or as a popup calendar (value = calendar). Default = calendar
<code>dateformat</code>	Specifies how the date is formatted in the UI and the control value. Valid values: short (for example, 1/2/02), medium (for example, 2 Jan, 2002), long (for example, 2 January, 2002)
<code>value</code>	The date value displayed by the control. The displayed value is formatted for the user's locale.
<code>width</code>	The width of the control. Default = 200 pixels.
<code>day</code>	Sets the default day to display. If not specified, 1 is selected.
<code>month</code>	Sets the default month to display. If not specified, 1 is selected.
<code>year</code>	Sets the default year to display. If not selected, the <code>fromyear</code> will be displayed.
<code>fromyear</code>	Sets the start year in the available date range. Default = 1997.
<code>toyear</code>	Sets the end year in the available date range. Default = 2005.

timeselector	Specifies whether the time is shown as fields, list, or text: Default dropdown lists display hour, minute, and day (increments are set by the value of the timestep attribute). Single dropdown list of discrete times is displayed as the values for timeformat, fromtime, totime, and timestep. Text box accepts text input of any kind.
timestep	Specifies the increment in seconds between times that are displayed when the timeselector is list or fields. If the timeselector is fields and the timestep value is a multiple of 60 or 3600, the seconds and minutes fields are not hidden. Default = 30 min.
timeformat	Specifies how the time is formatted in the UI and the control value (in locale-specific format). Valid values: short (for example, rendered as 2:40 PM in English locale), medium (for example, 2:40:23 PM), long (for example, 2.40:23 PM PST)
fromtime	Specifies the beginning time for a range when timeselector is list, for example, hh:mm:ss
totime	Specifies the end time for a range when timeselector is list, for example, hh:mm:ss
datafield	Overrides all of the date and time values with data from a data source
hour	The default hour to display. If not set, the current time is displayed.
minute	The default minute to display. If not set, the current time is displayed.
second	The default second to display. If not set, the current time is displayed.
onselect	Event fired when the control has been clicked

---

## Usage

In addition to the configuration available on individual date controls, global date display can be configured by extending `/wdk/config/datetimecontrol_config.xml`. The following table describes datetime control configuration in this file:

**Table 2-11. Global datetime control configuration elements**

<code>&lt;datetime&gt;</code>	Sets the default date format type for all instances of the datetime control, including those generated by <code>docbaseattributelist</code> control. Overridden by the <code>dateformat</code> and <code>timeformat</code> attributes on the datetime control.
<code>.&lt;default_type&gt;</code>	Specifies the type of Java <code>DateFormat</code> to apply (examples in parentheses). Valid values: <code>short</code> (12.13.52 or 3:30 pm) <code>medium</code> (Jan 12, 1952 3:30 pm) <code>long</code> (January 12, 1952 3:30:32 pm) <code>full</code> (Tuesday, April 12, 1952 AD 3:30:42 pm PST)

When you want to set a date and time programmatically in your component class, clear the datetime control by calling `clear()`, which clears day, month, and year. For example:

```
private void resetDateTimeControl(DateTime dt)
{
    if (dt != null)
    {
        dt.clear();
    }
}
```

Not all fields must be set for the control to be valid. The following table shows the state of each field and whether the combined state returns true for `isValid()`, `isValidDate()`, `isValidTime()`, and the value that will be displayed:

Yr/Mo/Day	Hour	Min	Sec	isValid	isValid Date	isValid Time	Value
not set				F	F	F	blank
all set	null	set	set	F	T	F	blank
all set	null	null	set	F	T	F	blank
all set	set	null	set	F	T	F	blank

Yr/Mo/ Day	Hour	Min	Sec	isValid	isValid Date	isValid Time	Value
all set	null	null	null	T	T	F	date,00: 00:00
all set	set	null	null	T	T	T	date, hr: 00:00
all set	set	set	null	T	T	T	date, hr: min:00
all set	set	set	set	T	T	T	date, hr: min:sec

## datetimevalidator

**Purpose** Verifies that the date input for a control matches a valid pattern, such as "mm dd yyyy", or "mm dd yyyy hh:mi:ss". This validator does not need to be used on datetime controls, because the input control displays only valid choices for the user.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

controltovalidate	(Required) The name of the control to validate
nlsid	ID of NLS text to use for validation failure ( overrides errormessage)
errormessage	The error message for validation failure

# datevalueformatter

**Purpose** Datevalueformatter control. Wraps a raw date value in milliseconds since 1970 and replaces the value with a formatted date value.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

type	The type of date to display. Maps to the Java DateFormat types, such as "short", "medium", "long", "full".
------	--

## Usage

**Table 2-12. Global datetime control configuration elements**

<datevalueformatter>	Sets the default date format type for all instances of the datevalueformatter control. Overridden by the type attribute on the datevalueformatter control.
.<default_type>	Specifies the type of Java DateFormat to apply (examples in parentheses). Valid values: short (12.13.52 or 3:30 pm) medium (Jan 12, 1952 3:30 pm) long (January 12, 1952 3:30:32 pm) full (Tuesday, April 12, 1952 AD 3:30:42 pm PST)

## datevalidator

**Purpose** Verifies that the dateinput control value is a valid date for the user's locale. This validator does not need to be used on dateinput controls, because the input control displays only valid choices for the user. The name and controltovalidate attributes are required. This control cannot be used to validate any control other than dateinput.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

controltovalidate	(Required) The name of the control to validate
nlssid	ID of NLS text to use for validation failure ( overrides errormessage)
errormessage	The error message for validation failure

---

## div

**Purpose** Generates HTML <div> open and close tags

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

HTML attributes

# docbaseattribute

**Purpose** Displays the label and value of an object specified in the associated `<docbaseobject>` tag. This control generates the following controls: `docbaseattributevalue` (which validates against the data dictionary), `docbaseattributelabel`, and some HTML table tags. This control is used within a table or within a component class.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

object	Name of associated <code>docbaseobject</code> control on the JSP. For example, <code>&lt;dmfx:docbaseobject name="obj"/&gt;</code> sets the object and <code>&lt;dmfx:docbaseattribute object="obj" attribute="object_name" ...&gt;</code> sets the attribute for that object.
attribute	Documentum attribute name
lines	Specifies the number of lines to display for an attribute. Use the <code>lines</code> and the <code>size</code> attributes to display the attribute.
size	Sets size of control
readonly	Sets whether the control is read-only. This setting does not override or change the status of the attribute; it only applies to the rendering of the attribute in the UI. The value of <code>readonly = "false"</code> can be overridden by the attributes component <code>readOnly</code> parameter.
required	Sets whether the attribute is required
pre	Sets the HTML to render before control. Default = <code>&lt;tr&gt;&lt;td&gt;</code> (opens the row).

col1	Sets the HTML to be rendered between label and value. Default = <code>&lt;/td&gt;&lt;td&gt;</code> . For example: <code>col1="&lt;/td&gt;&lt;td&gt;&lt;b&gt;:&lt;/b&gt; &lt;/td&gt;&lt;td&gt;"</code>
col2	Sets the HTML to be rendered between col1 and the value. Default = an empty cell: <code>&lt;/td&gt;&lt;td&gt;</code>
post	Sets the HTML to render after control. Default = <code>&lt;/td&gt;&lt;/tr&gt;</code> (closes the row).
datafield	Overrides attribute value
docbaseattribute valueassistance	Specifies the name of a custom value assistance list. The list is provided within a <code>docbaseattributevalueassistance</code> tag. Values are specified by <code>docbaseattributevalueassistancevalue</code> tags or by the component class.
validate	Set to false to prevent data dictionary validation. Default = true
showobjectidasname	Set to true to display the object ID as the name of the object. This value can be overridden by a formatter in a <code>docbaseobjectConfiguration</code> file. Default = true

---

## Usage

The `docbaseattribute` tag generates a table that displays an attribute. The attribute can be displayed as readonly or editable, based on the value of the `readonly` attribute. The following example from `checkin.jsp` has the `docbaseobject` tag, which gets information about the object from the repository. The `docbaseattribute` tag displays the format:

```
<dmfx:docbaseobject name="object"/>
<table border="0" cellpadding="2" cellspacing="0">
...
<tr>
  <td>
    <dmfx:docbaseattribute object="object" attribute="a_content_type"
      readonly="true"/>
  </td>
</tr>
</table>
```



To format the attributes that are displayed by this control, wrap the control in a formatter control. For example, you can format a boolean value with `booleanformatter`, a date with `datevalueformatter`.

If you wish to generate a list of attributes from the data dictionary rather than using individual attribute controls, use a `docbaseattributelist` control.

If you need to do custom (not data dictionary) validation for the attribute, do not use this control. Use the `docbaseattributevalue` tag with `docbaseattributelabel`. The `docbaseattributevalue` tag performs data dictionary validation and accepts other validators.

If you need to provide non-data dictionary value assistance for the attribute, provide a name for your value assistance set as follows:

```
<dmfx:docbaseattribute object="obj" attribute="keywords"
  docbaseattributevalueassistance="my_resultset"
  size="48" cssclass="defaultLabelStyle"/>
```

The values for the value assistance list are provided by `docbaseattributevalueassistance` value tags, either in the JSP page or in the component class. Refer [docbaseattributevalueassistancevalue](#), [page 82](#) for details.

## docbaseattributedelegate

**Purpose** Displays in a datagrid pseudoattributes and attributes for which there is a `docbaseobjectconfiguration` file. If there is no custom configuration, this control works the same as the `docbaseattribute` control.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

object	Name of associated docbaseobject control on the JSP. For example, <code>&lt;dmfx:docbaseobject name="obj"/&gt;</code> sets the object and <code>&lt;dmfx:docbaseattribute object="obj" attribute="object_name" ...&gt;</code> sets the attribute for that object.
attribute	Documentum attribute name
lines	Specifies the number of lines to display for an attribute. Use the lines and the size attributes to display the attribute.
size	Sets size of control
readonly	Sets whether the control is read-only. This setting does not override or change the status of the attribute; it only applies to the rendering of the attribute in the UI. The value of <code>readonly = "false"</code> can be overridden by the attributes component <code>readOnly</code> parameter.
required	Sets whether the attribute is required
pre	Sets the HTML to render before control. Default = <code>&lt;tr&gt;&lt;td&gt;</code> (opens the row).
col1	Sets the HTML to be rendered between label and value. Default = <code>&lt;/td&gt;&lt;td&gt;</code> . For example: <code>col1="&lt;/td&gt;&lt;td&gt;&lt;b&gt;:&lt;/b&gt; &lt;/td&gt;&lt;td&gt;"</code>
col2	Sets the HTML to be rendered between col1 and the value. Default = an empty cell: <code>&lt;/td&gt;&lt;td&gt;</code>
post	Sets the HTML to render after control. Default = <code>&lt;/td&gt;&lt;/tr&gt;</code> (closes the row).
datafield	Overrides attribute value
docbaseattribute valueassistance	Specifies the name of a custom value assistance list. The list is provided within a <code>docbaseattributevalueassistance</code> tag. Values are specified by <code>docbaseattributevalueassistancevalue</code> tags or by the component class.

---

validate	Set to false to prevent data dictionary validation. Default = true
showobjectidname	Set to true to display the object ID as the name of the object. This value can be overridden by a formatter in a doibaseobjectConfiguration file. Default = true

---

## Usage

To format the attributes that are displayed by this control, wrap the control in a formatter control. For example, you can format a boolean value with `booleanformatter`, a date with `datevalueformatter`. Other formatters include `htmlsafetextformatter`, `numberformatter`, `stringlengthformatter`, `multilinestringlengthformatter`, `docformatvalueformatter`, `docsvalueformatter`, `folderexclusionformatter`, `rankvalueformatter`, and `vdmbindingruleformatter`.

If you need to do custom (not data dictionary) validation for the attribute, do not use this control. Use the `repositoryattributevalue` tag with `doibaseattributelabel`. The `doibaseattributevalue` tag performs data dictionary validation and accepts other validators.

If you need value assistance for the attribute, provide a name for your value assistance set as follows:

```
<dmfx:doibaseattribute object="obj" attribute="keywords"
  doibaseattributevalueassistance="my_resultset"
  size="48" cssclass="defaultLabelStyle"/>
```

The values for the value assistance list are provided by `doibaseattributevalueassistance` value tags, either in the JSP page or in the component class. See [doibaseattributevalueassistancevalue](#), page 82 for details.

# doibaseattributelabel

**Purpose** Displays the label of an object specified in the associated `<doibaseobject>` tag

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

object	(Required) Name of associated doibaseobject control
attribute	Documentum attribute name
datafield	Overrides attribute value

---

## doibaseattributelist

**Purpose** Displays all visible attributes of the Documentum object specified as the associated doibaseobject control. Attributes are rendered by category as defined in the data dictionary or Configuration file. Each attribute is generated as a doibaseattribute control instance that encapsulates data dictionary information.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

name	(Required) Name of the control.
object	(Required) Name of associated doibaseobject control on the JSP page.
attrconfigid	(Required) A configuration ID to identify which control definition file should be used. Must correspond to the value for id in <attributelist id=" <i>attributes</i> "> in the corresponding control definition file.

readonly	Set to true to display attributes as read-only. This setting does not override or change the status of the attributes in the repository; it only applies to the rendering of the attribute in the UI. The value of <code>readonly = "false"</code> can be overridden by the attributes component <code>readOnly</code> parameter.
separator	Specifies HTML to render as a separator between attributes. Default = <code>&lt;tr&gt;&lt;td colspan = \"3\"&gt;&lt;hr&gt;&lt;/td&gt;&lt;/tr&gt;</code> . Valid for 5.2 repositories only.
size	Sets the size of the control in pixels
pre	Specifies an HTML string to render before each <code>doibasattribute</code> control. Scripts are not supported. Default = <code>&lt;tr&gt;&lt;th align=\"right\"&gt;</code>
col1	Specifies an HTML string to render between the label and value. Scripts are not supported. Default = <code>&lt;/th&gt;&lt;td&gt;&lt;b&gt;:&lt;/b&gt;&amp;nbsp;&lt;/td&gt;&lt;td&gt;</code>
col2	Specifies an HTML string to render between the value and validator. Scripts are not supported. Default = <code>&lt;/td&gt;&lt;td&gt;</code>
post	Specifies an HTML string to render after each <code>doibaseattribute</code> control. Scripts are not supported. Default = <code>&lt;/td&gt;&lt;/tr&gt;</code> (closes the row)
categorypre	Specifies an HTML string to render before each category. Scripts are not supported. Default = <code>&lt;tr&gt;&lt;td colspan=\"3\"&gt;&lt;hr&gt;&lt;/td&gt;&lt;/tr&gt; &lt;tr&gt;&lt;th align=\"left\"&gt;</code>
categorypost	Specifies an HTML string to render after each category. Scripts are not supported. Default = <code>&lt;/th&gt;&lt;/td&gt;&lt;/tr&gt;</code> (closes the row)
name	(Required) Name of the control.

morelinkpre	Specifies an HTML string to render before the <b>Show More</b> link. Scripts are not supported. Default = <code>&lt;tr&gt;&lt;td colspan="4"&gt;</code>
morelinkpost	Specifies an HTML string to render after the <b>Show More</b> link. Scripts are not supported. Default = <code>&lt;/td&gt;&lt;/tr&gt;</code>
visiblecategory	Comma-separated string that specifies the categories that are visible. Default = null (all categories are visible)
showcategorynames	Set to true to display category names in the UI. Category name is not shown if there is only one category or if <code>&lt;showpagesastabs&gt;</code> in the component definition equals true.
showobjectidasname	Set to true to display the object ID as the name of the object. This value can be overridden by a formatter in a <code>docbaseobjectConfiguration</code> file. Default = true

---

## Usage

A single `docbaseattributelist` control in the JSP eliminates the need for a separate `docbaseattribute` control for each attribute to be listed. Attributes are displayed in categories, and categories are displayed in the UI as tabs. You can configure the attributes that are displayed in categories using the data dictionary or an `attributelist Configuration` file. You can display a different set of attributes for each component that displays attributes and for any context that is defined by a qualifier in your application, for example, `checkin`, `checkin by administrator`, `checkin of a custom type`.

To set a different `docbaseattributelist` configuration in your behavior class depending on business rules, for example, to display a particular attribute only when a business condition is met, put two or more `docbaseattributelist` controls within a panel control on the page. Show or hide each panel using the panel `setVisible` method to use the proper attribute list, for example:

```
Panel attrlst1 = (Panel) getControl(ATTRLST1PANEL, Panel.class);
attrlst1.setVisible(true);
Panel attrlst2 = (Panel) getControl(ATTRLST2PANEL, Panel.class);
attrlst1.setVisible(false);
```

`Docbaseattributelist` controls are configured in a configuration file. For more information on configuring `attributelist` controls, see "`docbaseattributelist`" in *Web Development Kit and Client Applications Development Guide*.

## docbaseattributeseparator

**Purpose** Generates a separator. This control is rendered by the docbaseattributelist control.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

separator	Specifies HTML to render as a separator between attributes. Default = <code>&lt;tr&gt;&lt;td colspan = \"3\"&gt;&lt;hr&gt;&lt;/td&gt;&lt;/tr&gt;</code> .
-----------	---

## docbaseattributevalidator

**Purpose** Renders several validators for attributes: inputmaskvalidator, multivaluesinputmaskvalidator, requiredfieldvalidator, and varequiredfieldvalidator. This validator (and its included validators) is generated by the docbaseattributevalue control, so that validation is performed automatically on attributes.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

controltovalidate	Name of input control to validate
-------------------	-----------------------------------

## Usage

The docbaseattributevalidator tag is generated by the docbaseattribute and docbaseattribute value tags, so you do not have to add it explicitly to validate repository attribute values. Any value that is entered into a docbaseattribute tag field is automatically validated by inputmaskvalidator, multivaluesinputmaskvalidator, requiredfieldvalidator, and varequiredfieldvalidator.

# docbaseattributevalue

**Purpose** Displays the value of an object that is specified in the associated <docbaseobject> tag

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

object	Name of associated docbaseobject control
attribute	Documentum attribute name
size	Sets size of control
lines	Specifies the number of lines to display for an attribute value. Use the lines and the size attributes to display the value returned for an attribute.
readonly	Sets whether the control is read-only. This setting does not override or change the status of the attribute value in the repository; it only applies to the rendering of the attribute in the UI. The value of readonly = "false" can be overridden by the attributes component readOnly parameter.
required	Sets whether a value is required in the UI. Does not override data dictionary enforcement.



datafield	Overrides attribute value
docbaseattributevalueassistance	Specifies the name of a custom value assistance list. The list is provided within a docbaseattributevalueassistance tag. Values are specified by docbaseattributevalueassistancevalue tags or by the component class.
validate	Set to false to prevent data dictionary validation. Default = true
showobjectidasname	Set to true to display the object ID as the name of the object. This value can be overridden by a formatter in a docbaseobjectConfiguration file. Default = true

---

## Usage

Use this tag, and a docbaseattributelabel tag, if you need to use a custom validator for your attribute value. This tag generates a docbaseattributevalidator control, so it will be automatically validated against data dictionary constraints that have been set up for the attribute.

For information on using the docbaseattributevalueassistance attribute to set up non-data dictionary value assistance, see [docbaseattribute](#), page 71.

# docbaseattributevalueassistance

**Purpose** Displays an open-ended or closed set of values for value assistance. The attribute must not have value assistance set up in the DocApp.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

isListComplete	True to specify a closed list, false to specify open-ended (user can add values)
----------------	--

---

## Usage

Specify the name of your list as the value of the docbaseattributevalueassistance attribute on the docbaseattribute control. In the following example, the value assistance list is named my\_resultset:

```
<dmfx:docbaseattribute object="obj" attribute="keywords"
  docbaseattributevalueassistance="my_resultset"
  size="48" cssclass="defaultLabelStyle"/>
```

Add the docbaseattributevalueassistance control. The values are specified by docbaseattributevalueassistancevalue tags or by the component class. See [docbaseattributevalueassistancevalue, page 82](#) for an example of providing values in the component class. Values in the JSP page are shown in the following example. If the values are to be provided by the component class, the docbaseattributevalueassistance control must be empty.

```
<dmfx:docbaseattributevalueassistance name="my_resultset"
  islistcomplete="false">
  <dmfx:docbaseattributevalueassistancevalue
    label="Value A" value="valA"/>
  <dmfx:docbaseattributevalueassistancevalue
    label="Value B" value="valB"/>
</dmfx:docbaseattributevalueassistance>
```

# docbaseattributevalueassistancevalue

**Purpose** Displays a value for a non-data dictionary list of values for an attribute. The attribute must not have data dictionary value assistance set up.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

value	Specifies a valid value for the attribute
datafield	Specifies the datafield that provides the value for the attribute
nlsid	Specifies an NLS lookup key for the value label. Overrides label.
label	Specifies the label for the value.
labeldatafield	Specifies a datafield that provides a label for the value. Overrides label and nlsid.
locale	Specifies the locale for which the value is valid. If no locale attribute is specified, the value is displayed for all locales.

## Usage

Specify the name of your value assistance list as the value of the `doibaseattributevalueassistance` attribute on the `doibaseattribute` control. In the following example, the value assistance list is named `my_resultset`:

```
<dmfx:doibaseattribute object="obj" attribute="keywords"
  doibaseattributevalueassistance="my_resultset"
  size="48" cssclass="defaultLabelStyle"/>
```

Add the `doibaseattributevalueassistance` control. The values are specified by `doibaseattributevalueassistancevalue` tags or by the component class. Values in the JSP page are shown in the following example.

```
<dmfx:doibaseattributevalueassistance name="my_resultset"
  islistcomplete="false">
  <dmfx:doibaseattributevalueassistancevalue
    label="Value A" value="valA"/>
  <dmfx:doibaseattributevalueassistancevalue
    label="Value B" value="valB"/>
</dmfx:doibaseattributevalueassistance>
```

If the values are to be provided by the component class, the `doibaseattributevalueassistance` control must be empty, for example:

```
<dmfx:doibaseattributevalueassistance name="my_resultset"
  islistcomplete="false">
</dmfx:doibaseattributevalueassistance>
```

The values are set in the component class similar to the following example:

```
DocbaseAttributeValueAssistance attributeVA =
```

```
(DocbaseAttributeValueAssistance) getControl(
    "my_resultset", DocbaseAttributeValueAssistance.class);
attributeVA.setMutable(true);

DocbaseAttributeValueAssistanceValue attributeVAValue =
    new DocbaseAttributeValueAssistanceValue();
attributeVAValue.setLabel("Value A");
attributeVAValue.setValue("valA");
attributeVA.addValue(attributeVAValue);

DocbaseAttributeValueAssistanceValue attributeVAValue2 =
    new DocbaseAttributeValueAssistanceValue();
attributeVAValue2.setLabel("Value B");
attributeVAValue2.setValue("valB");
attributeVA.addValue(attributeVAValue2);
```

## docbaseicon

**Purpose** Generates a Documentum icon for the type and format of the object. The tag will resolve the icon based on format first, then type. Both format attributes override type attributes if both are resolved to icons. You can set the format or type programmatically using `setFormat()` or `setType()`, respectively, on the `DocbaseIconTag` class.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

typedatafield	Overrides type attribute value. For example: typedatafield='r_object_type'.
formatdatafield	Overrides format attribute value. For example, formatdatafield='a_content_type'.

---

linkcntdatafield	Specifies the datafield that supplies a link count to test for virtual documents. If you wish to display the virtual document icon for virtual documents, databind the linkcntdatafield attribute to 'r_link_ct'.
isvirtualdocdatafield	Specifies a datafield that identifies virtual documents. For example: isvirtualdocdatafield='r_is_virtual_doc'.
assembledfromdatafield	Specifies a datafield that identifies an object as a snapshot, for example, assembledfromdatafield='r_assembled_from_id'.
isfrozenassemblydatafield	Specifies a datafield that identifies a frozen snapshot object, for example, isfrozenassemblydatafield='r_has_frzn_assembly'.
isreferencedatafield	Specifies a datafield that identifies whether the object is a reference object
isreplicadatafield	Specifies a datafield that identifies whether the object is a replica object
folderiddatafield	Specifies a datafield that supplies the folder ID
format	Format code, for example, 'msw97'
size	Valid values: 16 or 32 (pixels)
type	Documentum type, for example, dm_folder

---

## docbasefoldertree

### Purpose

Generates a folder tree hierarchy of the cabinets and folders in a repository. You can set the limit for number of folders to be displayed in the tree in the file Environment.properties, located in /WEB-INF/classes/com/documentum/web/formext. Larger numbers will be replaced with a More Folders link.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

value	Sets the selected node ID. Stores the name of the selected tree item. This attribute is generally managed programmatically.
width	The width of the entire tree control. Use with height to put the tree in an IE Iframe of the specified width and height.
height	The height of the entire tree control. Use with width to put the tree in an IE Iframe of the specified width and height.
nodewidth	The width of an individual node graphic (default = 17). If node images exist, default width = width of Tplus.gif.
nodeheight	The height of an individual node graphic (default = 17). If node images exist, default height = height of Tplus.gif.
showroot	Determines visibility of the root node (default = true)
showicons	Determines visibility of icons for each tree item (default = false)
showlines	Determines visibility of tree lines (default = true)
cachechildren	Determines whether children are cached when nodes are collapsed (default = true)
normalstyle	An entire CSS-syntax style sheet rule enclosed in quotes and applied to a normal tree item
selectedstyle	An entire CSS-syntax style sheet rule enclosed in quotes and applied to a selected tree item
disabledstyle	An entire CSS-syntax style sheet rule enclosed in quotes and applied to a disabled tree item

---

normalclass	CSS class to apply to text of normal rows
selectedclass	Class to apply to text of selected rows
disabledclass	Class to apply to text of disabled rows
rootpath	Repository path to the root folder for the tree (default = the repository root itself)
frameborder	When width and height are specified to put the tree in an iFrame, specifies whether a frame border is required (default = true).
onchange	Server-side event0 that is fired when the control value has been modified (fired after the form is submitted)
onexpand	Fired when the user attempts to expand a node. If you override this node, then you need to write code to expand the appropriate node.
oncollapse	Fired when the user attempts to collapse a node. If you override this node, then you need to write code to collapse the appropriate node.
includevirtualdocuments	Set to true to include virtual documents in the tree
includeassemblydocuments	Set to true to include virtual document snapshots in the tree
multiselect	Set to true to allow multiple expansion of tree nodes. May have performance impact.
multiselectdepth	Specifies the level at which multiple expansion is supported. For example, a level of 2 prevents multiple Docbases (level 1) from being expanded in the tree.
ondrop	Specifies the name of an event handler for the drop event in the component class
dragenabled	Set to true to enable drag and drop to and from the tree
showlockedicon	Set to true to show lock icon next to locked object in the tree

---

For more information about drag and drop, see *Web Development Kit and Client Applications Development Guide*.

## docbaselockicon

**Purpose** Generates an icon for locked objects.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

datafield	Specifies a datafield that supplies the lock owner name and overrides lockowner attribute value. For example: datafield='r_lock_owner'. You can set the datafield programmatically using setLockOwner() on the DocbaseLockIconTag class.
lockowner	The owner of the lock.
size	Valid values: 16 or 32 (pixels)

---

## docbaseobject

**Purpose** Hidden control that specifies a Documentum object. Reference this control in docbaseattribute, docbaseattributevalue, and docbaseattributelist controls to display attributes of the object. Validation and data dictionary values are read for the object type if you specify the type attribute.



## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

objectid	Documentum object ID
src	DQL query to get the object
datafield	Overrides the objectid value
type	Name of a Documentum type. Specify this attribute only if the object is used as a type object.
lifecycleid	ID of lifecycle, if a lifecycle type is specified or istype equals true
lifecyclestate	Name of lifecycle if a lifecycle type is specified or istype equals true
istype	Indicates that the object should be used as a type object, valid only if src or datafield is specified. If src returns a list of object ID, you must set istype to false.
validate	Specifies whether validation should be performed for this object
modifyonversion	Set to true if the object will be checked in. This will allow a user to modify attributes during checkout.
configid	ID of a repositoryobject Configuration file whose root element is <docbaseobjectconfiguration>

---

### Usage

You can modify how certain attributes or attribute types are displayed by using a formatter class. You can modify how the attribute is saved by using a value handler class. These classes are used to display or handle data in a docbaseattribute, docbaseattributevalue, or docbaseattributelist control. You can modify which control is used to render an attribute by specifying a tag class that extends DocbaseAttributeValueTag. You can add configuration elements that are used by the formatter, handler, or tag class.

**Note:** Docbaseobject configuration files are used only for docbaseobject controls on a JSP page. Custom rendering is not performed in a listing view such as the doclist component in which multiple objects are displayed.

Custom formatters, handlers, and custom tag classes for objects must be registered in a docbaseobject configuration file whose root element is <docbaseobjectconfiguration>. A JSP page references this configuration by setting the configid attribute of <dmfx:docbaseobject> to the same value as the id of the <docbaseobjectconfiguration> element. If this attribute is not specified on the docbaseobject tag, the default configuration is used. This default configuration has an id of "attributes" and is found in /webcomponent/config/library/docbaseobjectconfiguration\_dm\_sysobject.xml.

The DocbaseObject methods setType() and setObjectId() cannot be used together. An IllegalArgumentException will be thrown.

To get all of the docbaseattributevalue controls that use a particular docbaseobject in the page, use code in similar to the following your component class:

```
DocbaseObject docbaseObj = (DocbaseObject) getControl("obj",
    DocbaseObject.class);
FindValidAttributeValue v = new FindValidAttributeValue(
    docbaseObj.getName());
this.visitDepthFirst(v);
Iterator it = v.getAttributeValues();
while (it.hasNext() == true)
{
    DocbaseAttributeValue value = (DocbaseAttributeValue) it.
        next();
    System.out.println(value.getAttribute() + ":" + value.
        getValue());
}
```

## docbaseobjectattributevalidator

**Purpose** Validator used to validate any control that is associated with a docbaseobject.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

attribute	Attribute against which the control's selected value is to be validated.
controltovalidate	Name of the input control to validate
header	Header for error message
nlsid	ID of message for validation failure
errormessage	Message for validation failure
object	Name of the docbaseobject tag on the JSP page. In the following example, the object being validated is named "obj", so the value of the object parameter for this validator would be "obj":

```
<dmfx:docbaseobject name="obj"
configid="attributes"/>
```

## Usage

This control is used only to validate attributes that are listed as ignored attributes in a docbaseattributelist configuration. For example, the list of available formats is displayed by the datadropdownlist control named "formatList" in the Webtop import JSP page, and the user's selection should be validated against object constraints for the selected object type. The format attribute is listed in the <ignore\_attributes> element of import\_docbaseattributelist.xml. In order for the selection to be validated, use the validator to validate the control on the JSP page, similar to the following:

```
<dmf:docbaseobjectattributevalidator
name="docbaseobjectvalidator_attr1"
object="docbaseObj" controltovalidate="
formatlist" attribute="a_content_type"/>
```

# docbaseobjectvalidator

## Purpose

Validates all modified attribute control values against all rules that have been set up in the Content server. Renders a list of error messages for all failed validators. This validator is embedded in the docbaseobject control, so validation is performed automatically on attributes that have been modified by the user.

**Tag library descriptor**

dmfx (dmformext\_1\_0.tld)

**Attributes**

---

controtovalidate	Input control name to validate
header	Header for error message
nlsid	ID of message for validation failure
errormessage	Message for validation failure

---

**doabasepriorityicon****Purpose**      Obtains an icon that represents the priority of a Documentum object**Tag library descriptor**

dmfx (dmformext\_1\_0.tld)

**Attributes**

---

datafield	Specifies a datafield that supplies the lock owner name and overrides lockowner attribute value. For example: datafield='r_lock_owner'. You can set the datafield programmatically using setLockOwner() on the DoabaseLockIconTag class.
prioritydatafield	Overrides priority value. Specifies a datafield that supplies the priority for the object, for example, prioritydatafield='priority'. You can set the priority

priority	programmatically using <code>setPriority()</code> on the <code>DocbasePriorityIconTag</code> class. Sets the priority for the object
----------	---

---

## docbaseselector

**Purpose** Generates a dropdown list of available repositories

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

onselect	Specifies the event to be fired when the user selects a Docbase
----------	---

---

## docformatvalueformatter

**Purpose** Wraps a raw Documentum format code, such as `msw97`, with a full description of the format from the data dictionary (in this example, Microsoft Word 97)

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

This formatter has no configurable attributes.

## docsizevalueformatter

**Purpose** Wraps a raw Documentum object size value, such as 12500.0, with a formatted value in KB or MB as appropriate (in this example, 12.5 KB)

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

type	Documentum type. If set to a value of dm_folder or any subtype, the formatter will display a blank value instead of 0 KB for objects of the specified type.
datafield	Overrides type property

## dragdrop

**Purpose** Indicates that one or more dragdropregion tags, or controls that implement drag and drop interfaces, are defined on the page

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

This control has no configurable attributes.

## Usage

To support drag and drop on the page, add the dragdrop tag to the page and a dragdropregion tag that contains controls to support a drag or drop (or both) operation. If you have a browsertree in your page, you do not need a dragdrop region tag: the browsertree control has an ondrop attribute.

For more information about drag and drop, see *Web Development Kit and Client Applications Development Guide*.

## dragdropregion

**Purpose** Specifies that the contained control or controls support drag and/or drop operations. The specific operations are named in the control attributes.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

label	Label for dragdrop region
datafield	Overrides label or NLS value with string from a data source
nlsid	ID of NLS text to use (overrides label)
ondrop	Specifies the name of a method that handles the ondrop event. You must enter a value here and set a value for enableddroppositions to enable drop operations.
ondropdatafield	Alternative method of specifying the name of the drop handler function. The value of the attribute is used as a key to retrieve the actual drop handler function

enableddroppositions	Indicates the drop positions that are supported by the control. Valid values: any comma separated combination of above   below   over
enableddroppositionsdatafield	Alternative method of specifying the type of drop positions which are supported by the control. The value of the attribute is used as a key to retrieve the action drop positions.
dragenabled	Indicates whether dragging of the control is supported
dragenableddatafield	Alternative method of specifying whether the control supports dragging. The value of the attribute is used as a key to retrieve whether dragging of the control is supported.
isbackground	Set to true if the dragdrop region to the page background

---

## Usage

To support drag and drop on the page, add the dragdrop tag to the page and a dragdropregion tag that contains controls to support a drag or drop (or both) operation. If you have a browsertree in your page, you do not need a dragdrop region tag: the browsertree control has an ondrop attribute.

To enable drop operations, set an ondrop event handler for the ondrop attribute and specific one or more supported drop positions for the enableddroppositions attribute. To enable drag operations, set the dragenabled attribute to true. You can enable both operations for a region.

Following are some examples of drag and drop regions:

- label in clipboard.jsp  
dragenabled='true'
- datagrid in /rooms/doclist\_body.jsp  
enableddroppositions="<%=IDropTarget.DROP\_POSITION\_OVER%>"  
ondrop='onDrop'

**Note:** This dragdropregion tag contains another nested dragdropregion tag that enabled drag

- link in vdmlist\_streamline.jsp  
ondrop='onDrop' enableddroppositions='above,over,below' dragenabled='true'



---

For more information about drag and drop, see *Web Development Kit and Client Applications Development Guide*.

## dropdownlist

**Purpose** Generates an HTML drop-down list control. Contains option tags or a single dataoptionlist control that contains a databound option tag. The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

value	The selected value.
width	Horizontal length of the drop down list control
height	Vertical length of the drop down list control
onselect	Event fired when the control has been clicked on. The form is not submitted until the onselect event is fired.
forceselection	Set to true to force the user to select an item from the dropdown list. If true, no item will be selected. If false, the first item in the list will be selected.
onchange	Specifies a server-side event handler that will be called when the form is submitted. The onchange event will not submit the form.
runatclient	Set to true to handle the onselect event.

---

## Usage

Options in the list can be set programmatically using the method `addOption()`. For example:

```
DropDownList dropdown = (DropDownList) get Control (
    DOCBASE, DropDownList.class);
Option option1 = new Option();
option1.set Value(value1);
option1.set Label (value1);
dropdown.addOption(option1);
```

The `onchange` event does not submit the form, so you cannot use the `onchange` event to update other lists or controls on the JSP page. If you need to update another control, handle the `onselect` event in your component class with a method that does nothing, and handle the `onchange` event with a method that performs the control update. For example:

```
public void onSelectTypeFilter(DropDownList control, ArgumentList args)
{
    String strMode = control.getValue();
    // Control update logic here
    updateControls();
}
```

When the dropdown list is rendered for a required attribute value with value assistance, `forceselection` is set to `true` and the default value is selected. If the required attribute does not have a default value, no item will be selected and the user will be forced to select an item.

# duplicatenamevalidator

**Purpose** Ensures that the specified attribute value for the specified object type does not have a duplicate value in the repository.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

attribute	Attribute that must have a unique value in the repository, such as object_name for a public ACL
type	Type of the object that must have a unique attribute value, such as dm_acl

## erasethevalueimage

**Purpose** Extends [image, page 116](#) and renders an image to click in order to erase the value of a label control

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

In addition to the attributes of [image, page 116](#), this control specifies the following attribute:

labeltoerase	Specifies the name of the label control whose value will be erased when the user clicks the image
--------------	---

## extpermvalueformatter

**Purpose** Formats an extended permission level from an int to a string: 0 = execute | 1 = change location | 16 = change state | 17 = change permissions | 18 = take ownership

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

This formatter has no configurable attributes.

# filebrowse

**Purpose** Launches a file browser on the user's system. The name attribute is required.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

size	Size of file browser control
value	File name, including path, that is selected in the browser
datafield	Overrides the label or nlsid on the browse button with data from the data dictionary
nlsid	Overrides the label on the browse button
onchange	Specifies a server-side event to be fired after the form is submitted
defaulttonenter	If true, the keyboard enter keypress calls the JavaScript handleKeyPress() function. Must be used with a button on the page whose default attribute is set to true. The button event will be fired when the user hits the <b>Enter</b> key in the defaulttonenter control
onselect	Event that is fired when the user selects a file

fileupload	Set to true to specify upload mode, in which the file will be uploaded when a file path is specified and a form submitted. Set to false to specify the default filename only mode, in which only the file name is uploaded and not the file contents.
runatclient	Set to true to handle onselect event on the client

---

## fileselectorapplet

**Purpose** Generates a file selector for browsing to a file on the user's local system. The user can drag files or directories onto the browser when the applet is loaded, and the files will be imported. This applet is used in the importcontainer component.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

See [5.2.5 content transfer applet attributes, page 15](#) for attributes.

Additional attributes:

---

alt	Specifies text that should be displayed if the browser understands the applet tag but can't run Java applets
vspace	Margin in pixels above and below the applet in the browser window
hspace	Margin in pixels for the right and left sides of the applet in the browser window
object	Object to be serialized and encoded as a string
value	Preset value for the applet

folderselect	Set to true to enable adding directories. Must be true to enable drag and drop of folders.
folderselectmode	Valid values: root (only current folder node is selected, not descendants)   tree (root folder and all its children are selected including their folder structure)   contents (only the files contained in the folder are selected.)
fileselect	Select files only, not folders
multiselect	Set to true to allow multiple file selection
showbuttonpanel	Set to true to show the <b>Change</b> button for HTTP import
onfileselect	Specifies the event to fire when a file has been selected
launchonstart	Valid values: selectFolders   selectFiles   null. Displays a <b>Select Files</b> or <b>Select Folders</b> dialog when the applet is started. If the attribute is not defined or empty, no dialog is launched. If the attribute value collides with the "fileselect" or "folderselect" attributes, then no action is taken.

---

## Usage

The file selector applet look and feel is configured in the fileSelectorAppletStyle.properties file located in the /theme/css directories of each application layer. The image paths for the file selector are specified in /WEB-INF/classes/com/documentum/web/form/control/fileselector/FileSelectorImageProp.properties.

## fireclientevent

### Purpose

Fires a client JavaScript event when the JSP page is loaded. Can contain one or more <dmf:argument> controls to pass values to the event handler. Used by the application integrations framework.

**Tag library descriptor**

dmf (dmform\_1\_0.tld)

**Attributes**


---

event	(Required) Specifies the name of an event handler on the client
includeargname	Set to true to pass event argument names and values to handler. Set to false to pass only the value (default).

---

**firepresubmitclientevent**

**Purpose** Fires a client JavaScript event before submitting the control event to the server. Can contain one or more <dmf:argument> controls to pass values to the event handler. Used by the application integrations framework.

**Tag library descriptor**

dmf (dmform\_1\_0.tld)

**Attributes**


---

controlid	Name of the control for which to fire the event. If not specified, the client event is fired before form submission. If specified, event will be fired for any control onclick event.
event	(Required) Specifies the name of an event handler on the client. If a controlid is

	specified, the event will fire only for the named control.
includeargname	Set to true to pass event argument names and values to handler. Set to false to pass only the value (default).

---

## folderexclusionformatter

**Purpose** Excludes the contents if the object type is folder. Can be used to hide inappropriate attributes for folders, such as version or authors.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

type	Folder object type
datafield	Specifies the datafield that supplies the folder object type

---

## folderselectionvalidator

**Purpose** Ensures that the user has sufficient permission to select the folder and to import into it. Checks user permission and folder security. For example, in the import component, this control validates the multidocbasetree control.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)



## Attributes

controltovalidate	(Required) The name of the control to validate
nlsid	ID of NLS text to use for validation failure ( overrides errormessage)
errormessage	The error message for validation failure

## form

**Purpose** Generates an HTML form on a Web page

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

enctype	Form data encoding type. Valid values: multipart/form-data   application/x-www-form-urlencoded   text/plain. Set to "multipart/form-data" on pages that upload file content such as check in, import and import rendition.
keepfresh	Set to true to keep the form refreshed. If the browser loads the form without sending a request back to the server, for example, when the user hits the <b>Back</b> button, a server post-back event is generated to render the component again. If the component is within a container, the container URL will be refreshed.
onkeepfresh	Specifies a JavaScript (client-side) event handler for the keepfresh event that is triggered when the user hits the <b>Back</b> button. In the Form class implementation,

this method provides a URL for the refresh event. Your handler can override this implementation.

---

## formatmessageparams

**Purpose**        Formats a message by inserting parameters into the message at runtime

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

datafield	Overrides nlsid
nlsid	ID of an NLS string that provides the fixed portion of the message, with dynamic placeholder within numbered brackets ({}).
width	Width of the message field, in pixels
height	Height of the message field, in pixels
encodelabel	True to specify that the label should be encoded (use if the label can contain HTML). Default = false

---

### Usage

The parameters are contained within the body of the control using argument controls. The place holders in an nlsid string are filled in at runtime with the argument values. For example: "MSG\_CHECKED\_IN\_BY =Checked in by {}" has a placeholder for 1 parameter, {}. During runtime, the placeholder {} gets filled in with first argument in control body. For example:

```
<dmfx:formatmessageparams nlsid="MSG_CHECKED_IN_BY">  
  <dmfx:argument name="vm_owner" datafield="owner_name"/>  
</dmfx:formatmessageparams>
```

The NLS string in the properties file is as follows:

```
MSG_CHECKED_IN_BY=checked in by {0},
```

## forminclude

**Purpose** Includes the contents of another form at the point of the tag insertion. The name attribute is required.

### Tag library descriptor

```
dmf (dmform_1_0.tld)
```

### Attributes

src	(Required) The JSP file to be included
formclass	(Required) Java class containing the event handler functions called when events are fired in the included JSP page.
nlsbundle	NLS bundle used by all the controls in the included form.
nlsclass	Java NLS class used by all the controls in the included form.

## formurl

**Purpose** Generates a form navigation link (URL). The argument tag can be used to pass arguments to the form.

### Tag library descriptor

```
dmf (dmform_1_0.tld)
```

## Attributes

---

url	(Required) The form JSP file. For example, url='<%=strContextPath + "/wdk/samples/control_pen/formurlPen.jsp"%>'
label	The text displayed as a link
target	The target frame to display the form. If not specified, the current frame will be used.

---

## fragment

**Purpose** Includes a fragment of a JSP page in the current page. The actual fragment that is dispatched at runtime is dependent on the client environment (clientenv).

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

visible	Specifies whether the fragment is visible. Set to false if you will turn on visibility from your component class.
src	Specifies the path of the JSP fragment. An absolute path that begins with "/", such as src="/wdk/fragments/modal/ModalContainerStart.jsp" will always include the specified fragment. A relative path, such as src="modal/ModalContainerStart.jsp" will dispatch a fragment based on runtime context.

---

## Usage

The user's clientenv context, and the fragment bundle lookup mechanism that is used for runtime dispatching, is described in detail in *Web Development Kit and Client Applications Development Guide*.

# frameset

**Purpose** Generates HTML <frameset> tags

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

onload	Specifies the event to be fired in the frameset onload event
onunload	Specifies the event to be fired in the frameset onunload event
cols	Specifies the pixel width of each column. Cannot be used with the rows attribute. Use asterisk (*) to specify that the width is determined by the browser. For example: rows="24,*"
rows	Specifies the pixel height of each row. Cannot be used with the cols attribute. Use asterisk (*) to specify that the height is determined by the browser. For example: rows="24,*"
frameborder	True to display a frame border
border	Width in pixels of border if frameborder is set to true
framespacing	Width in pixels of spacing between frames

## frame

**Purpose** Generates HTML <frame> tags

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

title	Specifies the title to be displayed for the frame
src	Source of frame content. Can be URL to component, HTML page, or JSP page (HTML or JSP path must be relative to application root). For example: src='/component/toolbar' or src='/webtop/classic/workarea.jsp'
frameborder	True to display a frame border
border	Width in pixels of border if frameborder is set to true
marginheight	Height in pixels between frame and browser window margin
marginwidth	Width in pixels of spacing between frame and browser window margin
noresize	True to make the frame not resizable
scrolling	True to display scroll bars for the frame

---

## governedicon

**Purpose** Generates an icon indicating governance. Requires Collaborative Edition enabled for the repository.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

label	Link text
src	(optional) URL to the image
srcdisabled	(optional) Image to be used when the action is disabled
datafield	Overrides label or NLS value with string from a data source
datafielddisabled	Overrides srcdisabled value
alttext	Text that will be displayed by mouse hover
nlsid	ID of NLS text to use (overrides label)
width	(Image action buttons only) Sets the width of the button
height	Sets the height of the button
showlabel	Set to true to show a label with the icon
showimage	Set to true to show the icon, false to show the label only
action	Specifies the ID of an action to be launched by the icon
oncomplete	Specifies the event handler to be called after the action completes

---

## head

**Purpose** Generates HTML <head> open and close tags

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

This tag has no configurable attributes

# helpimage

**Purpose**      Renders an image that launches context-sensitive help

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

See [image](#), page 116.

# hidden

**Purpose**      Generates a hidden control at the bottom of the page that stores serialized context information

## Tag library descriptor

dmf (dmform\_1\_0.tld)



## Attributes

value	Value stored in the control
datafield	Overrides value with data from a data source
onchange	Specifies the server-side event handler for a change in the value of this control. This event is fired after the form is submitted.
encrypt	Set to true to ensure that data passed in the hidden control cannot be viewed by the browser "View Source" option. The encrypt property can be used only on named controls. Default = true.

## head

**Purpose** Generates HTML <html> open and close tags

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

This tag has no configurable attributes

## htmlsafetextformatter

**Purpose** Formats text into HTML safe form by replacing strings that may be misinterpreted by the browser, such as <, >, &, newline characters, and single and double quotes.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

None

# iconwell

## Purpose

Generates an HTML icon well that displays icons either horizontally or vertically and associates an onclick event with the icon. This control replaces the component tabs in the 5.2.x locator components. Each component that uses an iconwell control must configure it in the component definition.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

border	Sets a border in pixels around the image if it is a link. Default = 0 pixels (no border).
renderorientation	Specifies the orientation of the well of icons. Valid values: horizontal (default)   vertical
selectedbgimage	Overrides the image specified for selected icon background specified in the component definition
normalbgimage	Overrides the image specified for normal icon background in the component definition
highlightedbgimage	Overrides the image specified for highlighted icon background in the component definition

---

disabledbgimage	Overrides the image specified for disabled icon background in the component definition
bgimage	Overrides the image specified for icon background in the component definition
onclick	Event fired when the icon has been clicked. When onclick is used, the image is turned into a link.
runatclient	Set to true to handle the onclick event on the client

---

## Usage

The iconwell control can be used to render a group of icons that launch actions or components. The set of icons is configured in the definition file for the component that includes an iconwell tag. This allows you to have a different set of icons for each component. The iconwell configuration elements are described below:

```
<component>
...
  1<iconwell>
    <iconwellicon>
      2<id> component_or_action_id</id>
      3<icon>path_to_icon</icon>
      4<label>string_or_nlsid_element</label>
      5<tooltip>string_or_nlsid_element</tooltip>
    </iconwellicon>
    6<normalbgimage>path_to_background_image</normalbgimage>
    7<disabledbgimage>path_to_disabled_background_image</disabledbgimage>
    8<selectedbgimage>path_to_selected_background_image</selectedbgimage>
    9<highlightedbgimage>path_to_highlighted_background_image</highlightedbgimage>
    10<defaultid> component_or_action_id</defaultid>
  </component>
```

The locator and locatorcontainer classes, Locator and LocatorContainer respectively, add a configuration value that can be read by locator components. Add the element `<breadcrumbiconwellmode>` and set the value to true to display the component name instead of the repository name as the breadcrumb in a locator.

- 1 Contains configuration for the component icons
- 2 ID of an action or component that will be launched by the icon. This value must match the value of the action or component ID in its XML definition.
- 3 Path within a theme directory to the icon. For example, a path of `/icons/perspective/inbox32.gif` would map to a file at `/root_directory/custom/theme/`

mytheme/icons/perspective\inbox32.gif. See *Web Development Kit and Client Applications Development Guide* for information on how themes are located.

- 4 Specifies a string value or <nlsid> element whose key represents a label for the icon
- 5 Specifies a string value or <nlsid> element whose key represents a tooltip for the icon
- 6 Path within a theme directory to the image that supplies the background for icons in the normal (unselected and enabled) state
- 7 Path within a theme directory to the image that supplies the background for icons in the disabled state (action not valid for user or context)
- 8 Path within a theme directory to the image that supplies the background for icons in the selected state
- 9 Path within a theme directory to the image that supplies the background for icons in the highlighted state (mouseover)
- 10 ID of an action or component that will initially be selected when the icon is rendered (default action or component)

## image

**Purpose** Generates an HTML image tag (<img>)

### Attributes

---

src	(Required) URL to the image on the server. Can be a path on the server or in the repository, for example: src="/webtop/component/getcontent?objectId=0900303980005172" (repository icon may affect performance).
datafield	Overrides value of src attribute with URL from a data source
alttext	Specifies alternate text to be displayed if the image fails to load, and as a tooltip
label	Overrides alttext and nlsid
nlsid	ID of NLS text to use (overrides alttext)

---

width	Width of the image control
height	Height of the image control
border	Sets a border in pixels around the image if it is a link. Default = 0 pixels (no border).
hspace	Set a horizontal margin (white space padding to the left and right) around the image.
vspace	Set a vertical margin (white space padding to the top and bottom) around the image.
onclick	Event fired when the control has been clicked. When onclick is used, the image is turned into a link.
runatclient	Set to true to handle the onclick event on the client

---

## Usage

To change the style of a button or image everywhere it appears in your application, apply your changes in a style sheet. For example, the webcomponents.css file in the coolblue theme defines a drilldownHeader class that references a background image:

```
.drilldownHeader { BACKGROUND-COLOR: transparent;
BACKGROUND-IMAGE: url('../images/streamline/tabbarbg.gif') }
```

You would extend the coolblue theme, copy the webcomponent.css file to your /theme/css directory, and change the URL to the background image.

# inlinewebform

**Purpose** Forces the JSP engine to buffer the output stream for webform controls and clear the buffer before a jump or nest to another component. This tag is required for portlets. Error messages are forwarded to the portal's registered error handler.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

This control has no configurable attributes.

## Usage

A wrapper JSP page, `/wdk/inlinewebform.jsp`, contains the `<dmf:inlinewebform>` tag, which redirects to the actual target URL. The wrapper page is called by the `<dmf:form>` tag for portal environments.

You do not need to add this tag to every JSP pages that is used in your portal application. The wrapper page will be called transparently for all pages containing a `<dmf:form tag>` in an application uses a portal environment.

# inputmaskvalidator

**Purpose** Validates a control's input values based on an input mask. The name and `controltovalidate` attributes are required.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

<code>controltovalidate</code>	(Required) The name of the control to validate
<code>nlsid</code>	ID of NLS text to use for validation failure ( overrides <code>errormessage</code> )

---

errormessage	The error message for validation failure
inputmask	The mask character string: # : numeric characters & : all characters A : alphanumeric characters only ? : alphabetic characters only U : uppercase alphabetic characters only L : lowercase alphabetic characters only Example: date mask ##/##/## permits the input date 12/24/95 To use one of the mask characters as a literal member of the mask string, place a double slash (\) preceding the character.

---

## Usage

You can apply the input mask validator to verify that the user's input matches an expected pattern. In the following example, the input mask rejects ZIP codes that do not match the mask pattern:

```
<dmf:text name='zip' size='10' defaultonenter='true'/>
<dmf:inputmaskvalidator name="zipval" controltovalidate="zip"
  inputmask="#####" enabled='true' errormessage="Must have 5 digits"/>
```

## label

**Purpose** Text label control that can be used to label an image, user input control, or other UI element

### Tag library descriptor

```
dmf (dmform_1_0.tld)
```

## Attributes

---

label	Text displayed as the label
datafield	Overrides label with data from a data source
nlsid	ID of NLS text to use (overrides label)
width	(Image label only) Width of the label. Default = width of label text.
height	(Image label only) Height of the label. Set this to the height of the graphic so that it will line up correctly.
imagefolder	The folder that the label graphics are stored in. If left.gif, right.gif and bg.gif are not found, a normal label will be drawn instead of an image label.
align	(Image label only) Alignment of the label text: HTML horizontal alignment attribute. Default = center
encodelabel	True to specify that the label should be encoded (use if the label can contain HTML). Default = false

---

## link

**Purpose** Generates a hyperlink to an event handler

### Tag library descriptor

dmf (dmform\_1\_0.tld)



## Attributes

label	Label for the link
datafield	Overrides label and nlsid with a string from a data source
nlsid	ID of NLS text to use (overrides label)
onclick	(Required) Event fired when the control has been clicked on

## listbox

**Purpose** Holds option objects or a single dataoptionlist control holding a databound option. The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

value	The selected value.
width	Horizontal length of the drop down list control
height	Vertical length of the drop down list control
multiselect	Set to true to enable multiple selection in the listbox
onchange	Server-side event that is fired when the form is submitted
onselect	Event fired when the control has been clicked. The form is not submitted until the onselect event is fired.

size	The number of rows of selectable data visible in the listbox (scrollbars are used if more rows are available).
runatclient	Set to true to handle the onselect event on the client

---

## macclientdetect

**Purpose** Places a Macintosh client detector applet on the page. The applet detects the browser version number and stores the OS name in the session.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

This control has no configurable attributes.

### Usage

The tag class has a public static method, `MacClientDetector.getMacOsName()`, that makes the Mac OS version available. This method returns one of three values: `MacClientDetector.MAC_OSX`, `MacClientDetector.MAC_OS9`, or null (the client is not Mac).

## menu

**Purpose** Generates a JavaScript menu or sub-menu. This tag must be contained within a `menugroup` tag. The name attribute is required. Menus can also be generated from `<actionlist>` elements within an action Configuration file for an object type.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

value	Menu title
datafield	Overrides value and nlsid with a value from a data source
nlsid	ID of NLS text to use (overrides value)
width	Specifies the control's width (default = length of text)
height	Specifies the control's height. This value is ignored for graphic menus.
accessible	Set to true for accessible mode (menus are not rendered).

---

## menugroup

**Purpose** Specifies the visual appearance of nested menu tags

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

imagefolder	The folder that the label graphics are stored in. If left.gif, right.gif and bg.gif are not found, a normal label will be drawn instead of an image label.
menuclass	The class defining the normal style of a popup menu. The default class is named 'menu'.

menuhighlightclass	The class defining the highlight style of the selected menu item. The default class is named menuHighlight
menudisabledclass	The class that defines the style used when the menu is disabled.
menuseparatorclass	The class that defines the separator style. The BACKGROUND-COLOR specifies the separator color.
menubarclass	The class that defines the normal style of a menu bar (menu button). The default class is named menubar.
menubarhighlightclass	The class that defines the highlighted menu bar
menubardisabledclass	The class defining the disabled style of disabled menu bar (menu button). The default class is named menubarDisabled.
target	Specifies the name of the frame where the popup should appear. By default, the popup appears in the same frame.

---

## menugroupactionmap

**Purpose** Used by components in the Application Connectors (appintg) environment to create an action map and return it to the authoring application in the form of an event argument

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

menugroupid	Specifies a menu that is defined the <menugroupid> element of the appintgcontroller component definition
-------------	--

---

## menugroupconfig

**Purpose** Generates the initial menu map that is passed to the Application Connector authoring application

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

menugroupid	Specifies a menu that is defined the <menugroupid> element of the appintgcontroller component definition
-------------	--

## menuitem

**Purpose** Renders a menu item. Must be used within a menu tag. The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

value	Menu item text
datafield	Overrides menu item value and nlsid with data from a data source
nlsid	ID of NLS text to use (overrides value)

imageicon	URL to an image to display on the left hand side of the menu item. The URL should not include the custom folder.
onclick	Event fired when the control has been clicked
runatclient	Set to true to handle the onclick event on the client
accessible	Set to true for accessible mode (menu item is not rendered).

---

## menuseparator

**Purpose** A menu separator control used within menu. The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

accessible	Set to true for accessible mode (separator is not rendered).
------------	--

---

## multidocbasetree

**Purpose** Generates the display of browser tree nodes including custom nodes. You can set the limit for number of folders to be displayed in the tree in the file Environment.properties, located in /WEB-INF/classes/com/documentum/web/formext. Larger numbers will be replaced with a More Folders link for better performance.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

value	Sets the selected node ID. Stores the name of the selected tree item. This attribute is generally managed programmatically.
width	The width of the entire tree control. Use with height to put the tree in an IE Iframe of the specified width and height.
height	The height of the entire tree control. Use with width to put the tree in an IE Iframe of the specified width and height.
nodewidth	The width of an individual node graphic (default = 17). If node images exist, default width = width of Tplus.gif.
nodeheight	The height of an individual node graphic (default = 17). If node images exist, default height = height of Tplus.gif.
showroot	Determines visibility of the root node (default = true)
showicons	Determines visibility of icons for each tree item (default = false)
showlines	Determines visibility of tree lines (default = true)
cachechildren	Determines whether children are cached when nodes are collapsed (default = true)
normalstyle	An entire CSS-syntax style sheet rule enclosed in quotes and applied to a normal tree item
selectedstyle	An entire CSS-syntax style sheet rule enclosed in quotes and applied to a selected tree item
disabledstyle	An entire CSS-syntax style sheet rule enclosed in quotes and applied to a disabled tree item

normalclass	CSS class to apply to text of normal rows
selectedclass	Class to apply to text of selected rows
disabledclass	Class to apply to text of disabled rows
rootpath	Repository path to the root folder for the tree (default = the repository root itself)
frameborder	When width and height are specified to put the tree in an iFrame, specifies whether a frame border is required (default = true).
onchange	Event fired when the control has been modified (fired when an onclick occurs)
onexpand	Fired when the user attempts to expand a node. If you override this node, then you need to write code to expand the appropriate node.
oncollapse	Fire when the user attempts to collapse a node. If you override this node, then you need to write code to collapse the appropriate node.
includevirtualdocuments	Set to true to include virtual documents in the tree
includeassemblydocuments	Set to true to include virtual document assemblies in the tree
multiselect	Set to true to allow multiple expansion of tree nodes. May have performance impact.
multiselectdepth	Specifies the level at which multiple expansion is supported. For example, a level of 2 prevents multiple Docbases (level 1) from being expanded in the tree.
ondrop	Specifies the name of an event handler for the drop event in the component class
dragenabled	Set to true to enable drag and drop to and from the tree
showlockedicon	Set to true to show lock icon next to locked object in the tree

---

For more information about drag and drop, see *Web Development Kit and Client Applications Development Guide*.



---

# multilinestringlengthformatter

**Purpose**      Formats strings that span multiple lines

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

maxlen	Specifies the cut-off length for the string
maxlennsid	Localized value of maxlen (integer). You can have a different maximum length for strings depending on locale.
postfix	Defines a postfix string appended to the truncated string, for example, '...'
postfixnsid	Localized value of postfix string.
showastooltip	Specifies whether the untruncated value is to be shown as a mouseover tooltip when the string gets truncated. Default = true.
wrapped	Specifies whether maxlen applies to the entire string (wrapped='false') or to each word in the string (wrapped='true'). For example, if maxlen='5' and wrapped='true', the string "hello world" would not be truncated, because the string can wrap between words and no word exceeds the length of 5. If wrapped='false', the string would be truncated after "hello".
lineseparator	Specifies the line separator in the string to be formatted.
lineseparatoroutput	Specifies the line separator to be used in the output string.

---

# multiselect

**Purpose** Displays two lists. User can select items from a list and order the selected items. Items are moved between the unselected and selected list with **Add** and **Remove** buttons. Items are ordered on the selected items list with **Up** and **Down** buttons.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

items	A serialized list of items to appear in the available items listbox. List be a String array with one value or two (value and label, in that order)
itemslabel	Label for the available items list
itemsnlsid	NLS string for the available items list label
buttoncssclass	CSS class to apply to button appearance
buttoncssstyle	CSS rule to apply to button appearance
buttonimagefolder	Specifies the location of the image folder if buttons are used to display items
buttonwidth	Width of the button in pixels
buttonheight	Height of the button in pixels
disabledclass	CSS class to apply to a disabled item
disabledstyle	CSS rule to apply to a disabled item
listboxcssclass	CSS class to apply to listboxes
listboxcssstyle	CSS rule to apply to listboxes
listboxwidth	Width of listboxes in pixels. Default = 200.
selecteditems	A serialized list of items to appear in the selected items listbox
selecteditemslabel	Label for the selected items list
selecteditemsnlsid	NLS string for the selected items list label

---

size	The size of the listboxes. Default = 10.
orderingenabled	Set to true to enable to <b>Up</b> and <b>Down</b> buttons that change the order of the list. Default = true
multiselect	Set to true to enable selection of multiple items for <b>Add</b> or <b>Remove</b> . If a move operation is chosen, only the first selection is moved. Default = false

---

## Usage

The component can retrieve a list of selected items from the control by calling `getSelectedItems()` on the control. The buttons trigger client side events so that the client doesn't have to make a round trip to the server every time a button is selected. When a trip to the server is made the control will update its data on the server side via a hidden field that keeps an updated item list.

The List object that holds items has the following features:

- Each element in a List represents one item.
- The element can be a String, which is a value with no label.
- The element can be an array of one String
- A serialized list of items can be an array of one string (value) or two strings (value and label).

If you supply the lists of available and selected columns in the JSP page, then the list must be serialized to a string before passing it in to the control. (You can create the serialized string from an ArrayList using the MultiSelectorUtil class static method `encodeItemlist()`). If you supply a list in your component class, pass it in as ArrayList.

In the following example from the VisibleRepositoryPreferences class, the class gets the controls on the JSP page, calls methods that get the ArrayList for each control, and initializes the items and selected items lists:

```
public void onInit(ArgumentList args)
{
    super.onInit(args);
    ...
    MultiSelector mltRepositories = (MultiSelector)getControl(
        REPOSITORY_SELECTOR, MultiSelector.class);
    mltRepositories.setSize("10");

    //Get selected connection broker and port
    DataDropDownList docbrokerControl = (DataDropDownList) getControl(
        CONTROL_DOCBROKER, DataDropDownList.class);
    Label docbrokerPortNumberControl = (Label) getControl(
        CONTROL_DOCBROKER_PORT, Label.class);
```

```
//Get all repositories for this connection broker
ArrayList arrDocbases = getInvisibleRepositories(
    docbrokerControl.getValue(), docbrokerPortNumberControl.
    getLabel());
//Write the list to the available repositories list
mltRepositories.setItems(arrDocbases);

//Get previously selected repositories, set to selected items list
mltRepositories.setSelectedItems(getVisibleRepositories());
}
```

## multivaluesinputmaskvalidator

**Purpose** Validates multiple values in a label using an input mask. The name and controltovalidate attributes are required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

controltovalidate	(Required) The name of the control to validate
nlsid	ID of NLS text to use for validation failure ( overrides errormessage)
errormessage	The error message for validation failure
inputmask	The mask character string: # : numeric characters & : all characters A : alphanumeric characters only ? : alphabetic characters only U : uppercase alphabetic characters only L : lowercase alphabetic characters only Example: date mask ##/##/## permits the input date 12/24/95 To use one of the mask characters as a

---

	literal member of the mask string, place a double slash (\\) preceding the character.
separator	Character that separates input values

---

## multivaluesrequiredfieldvalidator

**Purpose** Validates that all of the required values in a list box have been set

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

controltovalidate	(Required) The control that is validated by this validator
errormessage	The errormessage to be displayed when the validation returns false
indicator	Specifies an indicator to display next to a required field. If empty (indicator = ""), an asterisk will be displayed (&nbsp;*).
indicatornlsid	Specifies an NLSID that resolves a string to be displayed as a required field indicator
indicatorstyle	Specifies a CSS style rule that should be applied to the indicator
indicatorcssclass	Specifies a class in a style sheet that should be applied to the indicator

---

## nodatarow

**Purpose** Supplies a row of static data that will be automatically displayed if the enclosing datagrid returns no data rows from the query. Must be contained within the datagrid control. You must provide <td> tags and content within the opening and closing nodatarow tags.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

height	Height of the row
align	Horizontal alignment of the cells
valign	Vertical alignment of the cells

---

## numberformatter

**Purpose** Formats numeric data that is returned in result sets a doubles, including integer display and float values

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

type	(Required) Type of number. Valid values: integer, percent, or float (currency not implemented).
format	Format mask for the display, for example, format="\$###.##"
blankifzero	True to display a blank space if the number to be formatted is zero

---

## open

**Purpose** Opens a URL with the specified URL and optional event. Can be an external HTML page.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

label	Contains a string or NLSID to display as a label
datafield	Overrides the label attribute to display a label for the link. For example: uridatafield='<%=SearchResultSet.ATTR_URL%>'
uri	URI for the new window. This URI can be supplied by a component class or externalized to the component definition. For example: uri;'http://localhost:8080/wdk/component/inboxdrilldown'
uridatafield	Datafield that supplies the URI, overrides the uri attribute

options	HTML options for the new browser window
sendcontext	Set to true to add context to the URI. For external targets, set to false.
target	Specifies the target window. Default = <code>_dmfChild</code>

---

## option

**Purpose** Contains a single static name/value pair entry. Use this tag inside a dropdownlist, listbox, datadropdownlist, or datalistbox control. The name attribute is required. The value can be set on the tag, or it can be passed in from a result set.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

value	Value that is displayed in a drop down list or list box
datafield	Overrides value with data from a data source
nlsid	ID of NLS text to use (overrides label)
label	Text that is displayed to the user in the drop down list or list box
labeldatafield	Overrides label with data from a data source

---



---

## pane

**Purpose** Pane control contained within a paneset control. Displays content that can be fixed or scrolled. For information about general scrollable paneset design, see *Web Development Kit and Client Applications Development Guide*.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

overflow	Sets the scrollability of the pane. The default value is 'auto', which generates a scroll bar if the content exceed the scrolling area. The value 'hidden' clips excessive content. Use 'hidden' for fixed panes such as headers and footers.
----------	---

---

## panel

**Purpose** Panel control that wraps a group of controls as a mechanism for hiding or displaying the group

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

datafield	Sets the visibility of the panel. Overrides the 'visible' attribute value.
mode	Specifies how the datafield value is treated. Valid values are 'string' and 'boolean'. (Default = string). In string mode, the visibility is true if the datafield string length > 0. In boolean mode, the datafield value is treated as a boolean and evaluates to true for the values 1, T, and true.
reversevisible	Specifies whether panel visibility should be the boolean complement of the datafield value. For example, with this attribute set to true and the datafield value in boolean mode returns 1 (true), the panel visibility would be false. Default = false (turns off reversevisibility).

---

## Usage

Panels can be used to display or hide elements within the JSP. For example, the newnotepage component hides a panel in onInit():

```
Panel panel = (Panel) getControl("optionspanel", Panel.class);
panel.setVisible(false);
```

The panel is then displayed in onClickShowHideOptions(), which is called when the user clicks the Show/Hide options checkbox:

```
Panel panel = (Panel) getControl("optionspanel", Panel.class);
if (m_bShowOptions == true)
{
    panel.setVisible(true);
    linkShowHideOptions.setLabel(getString("MSG_HIDE_OPTIONS"));
}
else
{
    panel.setVisible(false);
    linkShowHideOptions.setLabel(getString("MSG_SHOW_OPTIONS"));
}
```

The options panel and link that makes it visible are coded in the JSP page is as follows:

```
<table border="0" cellpadding="2" cellspacing="0" width="100%">
<% /* Options */ %>
```

```

<tr>
  <td colspan="2" nowrap>
    <dmf:link name="showhideoptions" nlsid="
      MSG_SHOW_OPTIONS" onclick="onClickShowHideOptions"/>
  </td>
</tr>
<dmf:panel name="optionspanel">
<tr>
  <td>
    <dmf:checkbox name="subscribe" nlsid="
      MSG_SUBSCRIBE_TO_CABINET"/>
  </td></tr></dmf:panel></table>

```

## paneset

**Purpose** Contains pane controls to allow scrolling, so that buttons stay in view and content within a pane is scrolled.. One outermost paneset to a JSP page, which can contain nested panesets. For information about general scrollable paneset design, see *Web Development Kit and Client Applications Development Guide*.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

rows	Sets the height in pixels for the contained panes, for example, rows='110, *, 40'. The asterisk (*) indicates that the middle pane uses the remaining browser space. You must specify either rows or columns but not both.
cols	Sets the width for the contained panes, for example, cols='110, *, 40'. The asterisk (*) indicates that the middle pane uses the remaining browser space. You must specify either rows or columns but not both.

toppadding	Specifies the top padding of the paneset control in pixels. Applies only to the outermost paneset control in a page.
bottompadding	Specifies the bottom padding of the paneset control in pixels. Applies only to the outermost paneset control in a page.
rightpadding	Specifies the right padding of the paneset control in pixels. Applies only to the outermost paneset control in a page.
leftpadding	Specifies the left padding of the paneset control in pixels. Applies only to the outermost paneset control in a page.
minheight	Specifies the minimum height of the paneset control in pixels. Applies only to the outermost paneset control in a page.
minwidth	Specifies the minimum width of the paneset control in pixels. Applies only to the outermost paneset control in a page.

---

## Usage

Controls on the JSP page that are outside the `dmf:paneset` control may not be visible.

If you are displaying data from a datagrid in multiple panes, set the `rendertable` attribute to `false` so that the datagrid does not render `<table>` markup. Instead, manually add `<table>` start and close tags within your `<dmf:pane>` tags.

You can globally configure all paneset controls in your application by extending the paneset configuration file `/wdk/config/panesetcontrol_config.xml`. The following attributes can be globally configured for the outermost paneset and the inner panesets independently: `toppadding`, `bottompadding`, `leftpadding`, and `rightpadding`.

## password

**Purpose** Text control whose characters are displayed as `'*` in the browser. The `name` attribute is required.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

size	Length of the password control
value	Value stored in the control and displayed in the password box
onchange	Specifies the server-side event handler. This event is fired after the form is submitted.
defaulttonenter	If true, the keyboard <b>Enter</b> keypress calls the JavaScript handleKeyPress() function. Must be used with a button on the page whose default attribute is set to true. The button event will be fired when the user hits the <b>Enter</b> key in the defaulttonenter control
forcereenter	If set to true, the password box is cleared when it is re-drawn, forcing the user to re-enter a password.
required	If true, forces user to provide input to the password field on login before the authentication service is invoked. Default = true. If set to false, you must also remove the requiredfieldvalidator tag for the password control on the login JSP page.
runatclient	Behavior unknown

## permissionvalueformatter

**Purpose** Formats an ACL permission level from an int to a string: 1 = none | 2 = browse | 3 = read | 4 = relate | 5 = version | 6 = write | 7 = delete | 8 = unknown

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

This formatter has no configurable attributes.

## policystatenameformatter

**Purpose** Displays the state name using the policy ID

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

datafield	Specifies a datafield that provides the state name
-----------	--

## postserverevent

**Purpose** Renders a postServerEvent() JavaScript call to a method in a class on the J2EE application server. This tag is used within a JavaScript function.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

formid	ID of the current form, for example, '<%=form.getElementName()%>'
controlid	ID of the control that fires the event
handlerid	ID of the control that handles the event (optional)
handlermethod	Required for standalone Web application, optional for WDK for Portlets. Server method to be called.
clientsidehandlermethod	(WDK for Portlets only) Sets the name of the client side JavaScript handler function
eventargname#	Event argument name. Four numbered event arguments can be configured (eventargname1, etc.)
eventargvalue#	Value for the corresponding event argument

## Usage

The following example from advsearchex.jsp renders a JavaScript function call to postServerEvent(), which in turns calls the event handler executeSavedSearch() in the AdvSearchEx class. The class containing the event handler must be imported in the JSP page::

```
<%@ page import="com.documentum.webcomponent.library.advsearch.AdvSearchEx" %>
...
function executeSavedSearch()
{
  <dmf:postserverevent formid='<%=strPostToForm%>' handlerid='
  <%=advsearch.getElementName()%>' handlermethod='
  executeSavedSearch' />
}
```

## preference

**Purpose** Displays a preference for a component and saves user entries. The control embeds a columnlist control that generates a label and value in each row and, optionally, a description label below this row.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

preferenceid	(Required) ID of the preference that will be retrieved. One of the following attributes must have a value: preferenceid, datafield, or value..
scopeid	(Required) Name of the preferencescope control.
datafield	Datafield that provides preference to be displayed. Overrides value and preferenceid attributes.
value	Preference to be displayed, overrides preferenceid

---

### Usage

The preference control instantiates the columnlist control and provides the label to display the current column selection and a link that nest to the Column Selector component to modify the list of columns.

Preferences for a component, and default preference values, must be defined in the component Configuration file, within a <preferences> element. See *Web Development Kit and Client Applications Development Guide* for more information on defining component preferences. User preferences that apply across components are defined in the general preferences component.



## preferencescope

**Purpose** Used internally by the preference control to retrieve preference data. This control is hidden on the JSP page, which must contain one or more preference controls.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

Not documented

### Usage

This control should not be used independently. It is called by the WDK preferences framework.

## preferencedisplaygroups

**Purpose** Renders preferences that have display group configuration definitions in the <preferencedisplaygroups> element of a component definition (for example, the display\_preferences component definition). This control loops around the preference groups defined in XML and renders the enclosed body content for each iteration.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

scopeid	The value of this attribute is used to bind the <code>preferencedisplaygroup</code> to the <code>preferencscope</code> control and has essentially the same role as the <code>scopeid</code> on the preference control (defines the scope for the the <code>preferencedisplaygroup</code> configuration lookup)
scopeidattrname	The value of this attribute defines the name of the page context attribute in which the <code>scopeid</code> for the <code>preferencedisplaygroup</code> tag is stored.
prefidattrname	The value of this attribute defines the name of the page context attribute in which the current preference id is stored.

## Usage

Preference can be displayed in groups. The following elements can be added to the component definition to display groups:

```
<preferencedisplaygroups>
  <group id=xxx>
    <members>
      <preference-ref id=a1 />
      <preference-ref id=a2 />
      ...
    </members>
    <label><nlsid>MSG_FOO</nlsid></label>
  </group>
  <group id=yyy>
    ...
  </group>
  ...
</preferencedisplaygroups>
```

The order of groups and the order of preferences inside a group controls the display of preference groups. The `preference-ref` element must have an ID that matches an existing preference id in the `<preferences>` element of the component definition. The group `<label>` is used for rendering the group title.

## primaryfolderpathlink

**Purpose** Displays for a given sysobject the first full primary folder path on which the user has full browse permissions on all parts of the path. If the sysobject id refers to a cabinet, the rendered parent folder path is '/'.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

label	Folder path (truncated or full) that is displayed for the link
datafield	Datafield that supplies the folder path
showfullpath	Specifies whether to render the whole path or to truncate it. When false, the tag displays the two top folder names in the path including the cabinet, and the two bottom folder names when the path exceeds four folders. For example: /cabinet/folder1/.../folder4/folder5.
onclick	(Required) Specifies the event that will be fired when the user clicks the folder path link.

## privilegevalidator

**Purpose** Ensures that the user has been assigned the specified privilege

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

**Attributes**

---

privilege	Privilege that has been assigned to the user, such as sysadmin or superuser
-----------	---

---

**progressbar**

**Purpose** Displays a progress bar. Should be used with a progressbarupdater or a combocontainerpanel control for which the progressbarid attribute value is set to the same value as the progressbar id.

**Tag library descriptor**

dmf (dmform\_1\_0.tld)

**Attributes**

---

message	Contains a string message to be displayed with the progress bar
nlsid	Overrides message attribute value
value	Current percent complete.
minimum	Specifies minimum percent value on progress bar scale. Default = 0
maximum	Specifies maximum percent value on progress bar scale. Default = 100
height	Height in pixels of progress bar
width	Width in pixels of progress bar

---

## progressbarupdater

**Purpose** Updates a progress bar at specified steps. Should be used with a progressbar control.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

progressbarid	Value should match the ID attribute on the progressbar control in the JSP page
stepbarid	ID of a step bar
steplabelid	Current percent complete.

---

## quotevalidator

**Purpose** Tests the string contents of an input control for single or double quotation marks (' or "). Returns false if the strings contains a quotation mark. The name and controltovalidate attributes are required.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

controltovalidate	(Required) The name of the control to validate
nlsid	ID of NLS text to use for validation failure ( overrides errormessage)
errormessage	The error message for validation failure
ignore	Value to be ignored. Valid values: single quote, double quote, or empty string (default). For example, if you wish to test for double quotes only, set the value of this attribute to a single quote.

---

# radio

**Purpose** Generates an HTML radio element

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

value	The state of the radio element (true to display the radio as selected)
datafield	Overrides value with data from a data source
nlsid	ID of NLS key to use (overrides value)
label	Label text for the radio button
labelalign	Sets alignment of the radio button label: Uses HTML text alignment attributes.
onchange	Server-side event fired when the radio value is changed, after the form has been submitted

---

group	(Required) A group of radio buttons that the radio button is a member of. Only one radio in the group can be selected.
onclick	Event fired when the control has been clicked
runatclient	Set to true to fire the onclick event on the client

---

## Usage

The user can select only one radio widget in the group. To check which radio has been selected by the user in your component class, check the state of each radio, similar to the following:

```
if (((Radio)getControl("ctrl1", Radio.class)).getValue())
{
    // do something for ctrl1
}
else if (((Radio)getControl("ctrl2", Radio.class)).getValue())
{
    // do something for ctrl2
}
```

# rangevalidator

**Purpose**      Validates an input control by ensuring that its value is within a specified range

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

controltovalidate	(Required) The control that is validated by this validator
errormessage	The errormessage to be displayed when the validation returns false

indicator	Specifies an indicator to display next to a required field. If empty (indicator = ""), an asterisk will be displayed (&nbsp;*).
indicatornlsid	Specifies an NLSID that resolves a string to be displayed as a required field indicator
indicatorstyle	Specifies a CSS style rule that should be applied to the indicator
indicatorcssclass	Specifies a class in a style sheet that should be applied to the indicator
type	Type of value. Valid values: string, integer, double
minvalue	Minimum value of range
minvaluecontrol	Input control containing minimum value of range
maxvalue	Maximum value of range
maxvaluecontrol	Input control containing maximum value of range

---

## rankvalueformatter

**Purpose** Wraps a raw score (from 0.0 to 1.0) of a Verity search (pre-5.3 repository only) with a formatted HTML table containing a score bar graph. The ranking is based upon the number of occurrences of the search string in a given object's content or attributes and indicates the relevance of objects returned by a search.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

None



# regexvalidator

**Purpose** Verifies that the input control value matches the specified Apache regular expression pattern. For information about Apache expression syntax, see [the Apache Web site](#).

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

controltovalidate	(Required) The control that is validated by this validator
errormessage	The errormessage to be displayed when the validation returns false
indicator	Specifies an indicator to display next to a required field. If empty (indicator = ""), an asterisk will be displayed (&nbsp;*).
indicatornlsid	Specifies an NLSID that resolves a string to be displayed as a required field indicator
indicatorstyle	Specifies a CSS style rule that should be applied to the indicator
indicatorcssclass	Specifies a class in a style sheet that should be applied to the indicator
expression	(Required) The regular expression to match against

---

# repositorysearch

**Purpose** Top level container control that uses the other search controls to form a search query. All search controls on the JSP page must be contained within this control.

## Tag library descriptor

dmfxs (dmfxsearch\_1\_0.tld)

## Attributes

This control has no specific configurable attributes.

# requiredfieldvalidator

**Purpose** Validates an input control by ensuring that its value has been set. The name and controltovalidate attributes are required.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

controltovalidate	(Required) The control that is validated by this validator
errormessage	The errormessage to be displayed when the validation returns false
indicator	Specifies an indicator to display next to a required field. If empty (indicator = ""), an asterisk will be displayed (&nbsp;*).
indicatornlsid	Specifies an NLSID that resolves a string to be displayed as a required field indicator
indicatorstyle	Specifies a CSS style rule that should be applied to the indicator
indicatorcssclass	Specifies a class in a style sheet that should be applied to the indicator

---

## Usage

**Required field** — You can require the user to enter data by adding a required field validator. The following example from newCabinet.jsp validates that the user has entered a name for the new cabinet:

```
<dmf:text name="attribute_object_name" size="30"></dmf:text>
<dmf:requiredfieldvalidator name="validator"
  controltvalidate="attribute_object_name"
  nlsid="MSG_MUST_HAVE_NAME">
</dmf:requiredfieldvalidator>
```

In this example, the text control is named "attribute\_object\_name". This name is used as the value for the validation control controltvalidate attribute value. The error message is specified as the value of the nlsid attribute, which can be an externalized string, as in the example, or a hard-coded string.

## richtextdocbaseattribute

**Purpose** Displays the label and value of a rich text pseudoattribute of an object specified in the associated docbaseobject control. tag. Pseudo attributes are described in the object type's docbaseattributelist xml file. The richtextdocbaseattribute control is only valid in the property sheet context. This control generates the following controls: richtextdocbaseattributevalue, richtextdocbaseattributelabel, and some HTML table tags. Requires Collaborative Edition on the Content Server.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

object	Name of associated docbaseobject control on the JSP. For example, <dmfx:docbaseobject name="obj"/> sets the object and <dmfx:docbaseattribute
--------	---

	object="obj" attribute="object_name"...> sets the attribute for that object.
attribute	Pseudo attribute name
value	Specifies the text that is being edited, in HTML format. Can be inherited from StringInputContol or re-implemented in the RTE control class if it doesn't extend StringInputControl.

---

## richtextdocbaseattributelabel

**Purpose** Displays the label of a rich text pseudoattribute of an object specified in the associated docbaseobject tag. Pseudo attributes are described in type's docbaseattributelist xml file, so this tag is only valid in the property sheet context. Requires Collaborative Edition on the Content Server.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

object	(Required) Name of associated docbaseobject control
attribute	Pseudo attribute name
datafield	Datafield that provides the label for the attribute

---

## richtextdocbaseattributevalue

**Purpose** Displays the value of a rich text pseudoattribute of an object specified in the associated docbaseobject tag. Pseudo attributes are described in type's docbaseattributelist xml file, so this tag is only valid in the property sheet context. Requires Collaborative Edition on the Content Server.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

object	Name of associated docbaseobject control
attribute	Pseudo attribute name

## richtextdisplay

**Purpose** Displays rich text attributes as read-only. Requires Collaborative Edition on the Content Server.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

datafield	Specifies a pseudoattribute to be displayed, for example, content_value.
-----------	--

## Usage

You can configure the behavior of the rich text display control by extending the control configuration file `/wdk/config/richtext.xml`, which defines the following elements:

**Table 2-13. Rich text configuration elements**

---

<code>&lt;inputfilter&gt;</code>	Remove the comments from this element to use the <code>RichTextInputFilter</code> class, which processes images and links that are entered by the user
<code>&lt;outputfilter&gt;</code>	Remove the comments from this element to use the <code>RichTextOutputFilter</code> class, which processes image URLs for display in IE and Mozilla
<code>&lt;html_input&gt;</code>	Contains elements that govern HTML input
<code>&lt;allowed_tags&gt;</code>	Contains all HTML tags that are allowed within rich text input. Must include start and closing tag, for example, <code>&lt;td&gt;&lt;/td&gt;</code> .
<code>&lt;allowed_attributes&gt;</code>	Contains all attributes that are allowed within HTML tags. Attributes are represented as though they were HTML tags, for example, <code>&lt;href&gt;&lt;/href&gt;</code> .
<code>&lt;allowed_protocols&gt;</code>	Contains protocols that are permitted in links within rich text. For example, to prevent ftp URLs, remove <code>&lt;ftp&gt;&lt;/ftp&gt;</code> from the list.
<code>&lt;invalid_stylesheet_constructs&gt;</code>	Contains stylesheet constructs that are not permitted, such as those that contain external links. For example: <code>&lt;div style="background: url('http://www.somewebsite.com/image/someimage.jpg');"&gt;</code> <code>&lt;/div&gt;</code>

---

---

# richtexteditor

**Purpose** Generates a rich text editing add-in. Requires Collaborative Edition on the Content Server.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

value	Specifies the text that is being edited, in HTML format. Can be inherited from StringInputControl or re-implemented in the RTE control class if it doesn't extend StringInputControl.
hasFonts	Boolean value that determines whether the font and font-size menus are included in the RTE toolbar.
hasCharFormatting	Boolean value that determines whether the character formatting buttons are included in the toolbar (bold, italic, underline, text color, highlight color). Default = true.
hasParaFormatting	Boolean value that determines whether the paragraph formatting buttons are included in the toolbar (alignment, bullet list, numbered list, indent, outdent). Default = true
hasClipboard	Boolean value that determines whether the clipboard button is included in the toolbar. Default = true
hasImages	Boolean value that determines whether the image button is included in the toolbar. Default = true

hasSpellcheck	Boolean value that determines whether the spell check button is included in the toolbar. (Windows only) Default = true
hasLinks	Boolean value that determines whether the link button is included in the toolbar Default = true

---

## Usage

You can configure the behavior of the rich text editor by extending the control configuration file `/wdk/config/richtext.xml`, which defines the following elements:

**Table 2-14. Rich text configuration elements**

---

<code>&lt;inputfilter&gt;</code>	Remove the comments from this element to use the <code>RichTextInputFilter</code> class, which processes images and links that are entered by the user
<code>&lt;outputfilter&gt;</code>	Remove the comments from this element to use the <code>RichTextOutputFilter</code> class, which processes image URLs for display in IE and Mozilla
<code>&lt;html_input&gt;</code>	Contains elements that govern HTML input
<code>&lt;allowed_tags&gt;</code>	Contains all HTML tags that are allowed within rich text input. Must include start and closing tag, for example, <code>&lt;td&gt;&lt;/td&gt;</code> .
<code>&lt;allowed_attributes&gt;</code>	Contains all attributes that are allowed within HTML tags. Attributes are represented as though they were HTML tags, for example, <code>&lt;href&gt;&lt;/href&gt;</code> .



---

<allowed_protocols>	Contains protocols that are permitted in links within rich text. For example, to prevent ftp URLs, remove <ftp></ftp> from the list.
<invalid_stylesheet_constructs>	Contains stylesheet constructs that are not permitted, such as those that contain external links. For example: <div style="background: url('http://www.somewebsite.com/image/someimage.jpg');"> </div>

---

Elements that configure the rich text editor are described in the table below. If no minimum version is provided, the browser will not be allowed to use the rich text editor.

**Table 2-15. Rich text editor configuration elements (<editor>)**

---

<ieminversion>	Minimum version of IE supported by editor
<mozillaminversion>	Minimum version of Mozilla supported by editor
<netscapeminversion>	Minimum version of Netscape supported by editor
<safariminversion>	Minimum version of Safari supported by editor
<iemaxversion>	Maximum version of IE supported by editor
<mozillamaxversion>	Maximum version of Mozilla supported by editor
<netscapemaxversion>	Maximum version of Netscape supported by editor
<safarimaxversion>	Maximum version of Safari supported by editor

---

## richtextpanel

**Purpose** Surrounds the rich text editor and the rest of the HTML, such as the field name, so it will not be generated if rich text is not enabled. Rich text editing or display requires Collaborative Edition on the Content Server.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

This control has no configurable attributes.

### Usage

To display an editable rich text attribute, add a richtextpanel similar to the following example (table tags place this panel within a larger table in the component):

```
<dmfx:richtextpanel>
<tr>
  <td scope="row" width="10%" align="right">
    <b><dmf:label nlsid="MSG_DESCRIPTION_COLON"/><b></b>
  </td>
  <td width="90%" align="left">
    <dmfx:richtexteditor name='<%=DocList.
      FOLDER_DESCRIPTION%' hasImages='true' />
  </td>
</tr>
</dmfx:richtextpanel>
```

## roombanner

**Purpose** Generates a room banner. Requires Collaborative Edition to be enabled for the repository.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

height	Height of the banner, in pixels
objectid	ID of a governed folder containing the room banner

## row

### Purpose

Generates a static header or footer HTML table row inside the data grid control. Adds the static data row above or below the dynamic datagridrows. Can be used in place of <tr>. Use only in pages where your component needs to access controls in the row.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

height	Height of the row
align	Horizontal alignment of the cells
valign	Vertical alignment of the cells

## searchattribute

### Purpose

Displays a specific repository object attribute for search criterion

## Tag library descriptor

dmfxs (dmfxsearch\_1\_0.tld)

## Attributes

---

attribute	Specifies an attribute in one of the repositories that can be searched
andorvisible	Specifies whether AND/OR selector is shown. Default = true
removable	Set to true to allow the attribute to be removed from the containing group
label	Specifies a string to be displayed for the attribute
nlsid	Overrides label
casevisible	Removes the checkbox <b>Match case</b> in advanced search. The search is by default case sensitive for faster performance. Default match case is set in /wdk/config/advsearchex.xml.

---

## Usage

Data dictionary value assistance is supplied, although conditional value assistance is not supported. To add non-data dictionary value assistance, you can configure custom classes to perform the assistance. The assistance is configured in /wdk/config/advsearchex.xml. For information on setting up non-data dictionary value assistance for search, see "Programmatic Search Value Assistance" in *Web Development Kit and Client Applications Development Guide*.

For case-sensitive attributes, configure default case-sensitivity in /wdk/config/advsearchex.xml as follows:

```
<defaultmatchcase>>false</defaultmatchcase>
```

**Note:** Case-sensitive queries can slow performance.

# searchattributegroup

**Purpose** Container of other attribute-related search controls. It creates the child controls when the user clicks **Add**. It contains tags for specific types of attributes and can exclude other, specified types of attributes. For information on how to display specific attributes for search, refer to "Configuring advanced search" in *Web Development Kit and Client Applications Development Guide*.

## Tag library descriptor

dmfxs (dmfxsearch\_1\_0.tld)

## Attributes

---

excludetypes	A comma separated list of attribute types to be excluded from the auto-populated dropdown for attribute names. Valid attribute types are: Boolean or bool   id   double   int or integer   date   time   string   number (DM_DOUBLE or DM_INTEGER)
excludenames	A comma separated list of attribute names to be excluded from the auto-populated dropdown for attribute names.
addvisible	Specifies whether the <b>ADD</b> and <b>REMOVE</b> buttons should be shown. Default = true
casevisible	Removes the checkbox <b>Match case</b> in advanced search. The search is by default case sensitive for faster performance. Default match case is set in /wdk/config/advsearchex.xml.
opened	Set to true to display the attribute group as expanded. Default: false (closed)
defaultattribute	Specifies the attribute that will be selected by default. Must not an attribute named in the excludenames attribute.

---

## Usage

By default, when the user clicks **Add Property**, the default operator is AND. Your component class can call `setLogicalOr` on the `searchattributegroup` control to change the default to OR.

The precedence of constraints is given to the last constraint added. For example:

```
A
OR B
AND C
is equivalent to ((A OR B) AND C)
```

```
A
AND B
OR C
AND D
is equivalent to ((A AND B) OR C) and D)
```

# searchdateattribute

**Purpose** Subclass of `SearchAttribute` control. It can be used to input search values on date attributes.

## Tag library descriptor

dmfxs (dmfxsearch\_1\_0.tld)

## Attributes

---

attribute	Specifies a date attribute in one of the repositories that can be searched
andorvisible	Specifies whether <b>AND/OR</b> selector is shown. Default = true
removable	Set to true to allow the attribute to be removed from the containing group
label	Specifies a string to be displayed for the attribute
nlsid	Overrides label

---

## Usage

This control by default does not have a dropdown item to allow the user to select a date value that is "In the next" day/week/month/year for effective date and expiration date attributes. The visibility of the dropdown items is configured in `/wdk/config/advsearchex.xml` as follows:

```
<scope type="dm_sysobject">
  <inthenextdate>
    <a_effective_date>true</a_effective_date>
    <a_expiration_date>true</a_expiration_date>
  </inthenextdate>
</scope>
```

## searchfulltext

**Purpose** Accepts keywords for a full-text query. Searches indexed documents and indexed attributes. You can use AND and OR operators, and you can search on exact strings within quotation marks, for example, "this string".

### Tag library descriptor

dmfxs (dmfxsearch\_1\_0.tld)

### Attributes

value	Keyword for full-text search
datafield	Specifies a datafield that supplies the keyword, overriding value and nlsid
nlsid	Overrides value

## searchlocation

**Purpose** Extends the label control and displays the current search source and folder path as a static string. The location is expressed as a semi-comma separated list in the format of soucname[:]/cabinet/folder.

### Tag library descriptor

dmfxs (dmfxsearch\_1\_0.tld)

### Attributes

---

label	Label for the location display
datafield	Specifies a datafield that supplies the label, overriding label and nlsid
nlsid	Overrides label

---

## searchobjecttypedropdownlist

**Purpose** Generates a dropdown list from which the user can select the object type to search on

### Tag library descriptor

dmfxs (dmfxsearch\_1\_0.tld)



## Attributes

---

forselection	Set to true to force the user to select an item from the dropdown list. If true, no item will be selected. If false, the first item in the list will be selected.
includetypes	Specifies a comma-delimited list of object type names that will be displayed in the dropdown list, for example, "mytype,yourtype". If a listed type is selected, subtypes of that type will also be queried. If a value is not a valid type in the reference repository (see below), it will not be displayed.

---

## Usage

The options in this list can be set programmatically. See [dropdownlist, page 97](#) for an example. This control has also a method `setOptionFromScrollableResultSet()`.

If multiple search locations are set in the user preferences, a reference repository is used to present the list of types. This repository is either the first one selected by the end user or the repository that is used as a BOF registry if the user has not selected a repository. If a selected type is not known to the reference repository that is used to drive the composition of the advanced search, it will be ignored.

# searchscopecheckbox

**Purpose** Sets a value for a search setting or capability. Currently it is used to enable finding all versions or hidden objects.

## Tag library descriptor

dmfxs (dmfxsearch\_1\_0.tld)

## Attributes

---

label	Label for the checkbox
labelalign	Set alignment of the checkbox label: left or right
datafield	Specifies a datafield that supplies the label, overriding label and nlsid
nlsid	Overrides label
scopename	The name of the global setting. Valid values: type   findAllVersions   findHidden
checkedvalue	Specifies a valid value for the scopename. Set to true or false for findAllVersions or findHidden. Set to a specific object type to set the base type for search.
uncheckedvalue	Specifies a valid value for the scopename. Set to true or false for findAllVersions or findHidden. Set to a specific object type to set the base type for search.

---

## searchsizeattribute

**Purpose** Subclass of SearchAttribute control. It can be used to input search values on size attributes. You can configure the global size values that are displayed, and their labels, by extending /wdk/config/advsearchex.xml and copying the <searchsizeattributerange> element and its contents.

## Attributes

---

attribute	Specifies a size attribute in one of the repositories that can be searched
andorvisible	Specifies whether <b>AND/OR</b> selector is shown. Default = true
removable	Set to true to allow the attribute to be removed from the containing group

label	Specifies a string to be displayed for the attribute
nlsid	Overrides label

## Usage

Size options can be configured for all instances of this control by extending the control configuration file `/wdk/config/advsearchex.xml`. The following control attributes can be configured to add a search size option:

Element	Description
<option>	Defines a size option for display. Contains <label>, <value>, <operator>, and <unit> elements.
<label>	Contains a string or <nlsid>. The NLS ID key is resolved by the NLS resource file <code>SearchControlNlsProp.properties</code>
<operator>	Specifies the size operator. Valid operators: LT (less than)   GT (greater than)
<value>	Default value to be displayed
<unit>	Unit for value. Valid units: MB   KB

# serviceprogressfeedback

**Purpose** Generates a JavaScript progress bar that is updated during content transfer

## Tag library descriptor

dmxfer (dmcontentxfer\_1\_0.tld)

## Usage

This control is used to display content transfer progress in content transfer components. The control is used in the JSP page `serviceprogress.jsp`. Include this JSP page in your custom content transfer container definition, for example:

```
<pages>
  <serviceprogress>
    /webcomponent/library/contentxfer/serviceProgress.jsp
  </serviceprogress>
</pages>
```

In your container class, call `setComponentPage("serviceprogress")` when the transfer is about to begin. For example, the export container class calls `setComponentPage()` when the user clicks the OK button:

```
final public void onOk(Control button, ArgumentList args)
{
    if ( canCommitChanges() && onCommitChanges() )
    {
        // initializes the tag for service progress
        m_progress.setServiceMgr(this);
        setComponentPage("serviceprogress");
    }
}
private ServiceProgressFeedback m_progress = null;
```

## showtopicdocbaseattribute

### Purpose

Displays the label and value of a rich text pseudoattribute of an object specified in the associated `docbaseobject` control. `tag`. Pseudo attributes are described in the object type's `docbaseattributelist.xml` file. The `richtextdocbaseattribute` control is only valid in the property sheet context. This control generates the following controls: `richtextdocbaseattributevalue`, `richtextdocbaseattributelabel`, and some HTML table tags. Requires Collaborative Edition on the Content Server.

### Tag library descriptor

`dmfx (dmformext_1_0.tld)`

## Attributes

---

object	Name of associated docbaseobject control on the JSP. For example, <dmfx:docbaseobject name="obj"/> sets the object and <dmfx:docbaseattribute object="obj" attribute="object_name" ...> sets the attribute for that object.
attribute	Pseudo attribute name
value	Specifies the text that is being edited, in HTML format. Can be inherited from StringInputControl or re-implemented in the RTE control class if it doesn't extend StringInputControl.

---

## showtopicdocbaseattributelabel

**Purpose** Displays the label of show topic pseudoattribute of an object specified in the associated docbaseobject tag. Pseudo attributes are described in type's docbaseattributelist xml file, so this tag is only valid in the property sheet context.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

object	(Required) Name of associated docbaseobject control
attribute	Pseudo attribute name
datafield	Datafield that provides the label for the attribute

---

## showtopicdocbaseattributevalue

**Purpose** Displays the value of a show topic pseudoattribute of an object specified in the associated docbaseobject tag. Pseudo attributes are described in type's docbaseattributelist xml file, so this tag is only valid in the property sheet context.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

object	Name of associated docbaseobject control
attribute	Pseudo attribute name

---

## td

**Purpose** Generates HTML <span> open and close tags

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

HTML attributes

---

# stringlengthformatter

**Purpose** Truncates a string and provides a mouseover popup that contains the untruncated string

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

maxlen	Specifies the cut-off length for the string
maxlenlsid	Localized value of maxlen (integer). You can have a different maximum length for strings depending on locale.
postfix	Defines a postfix string appended to the truncated string, for example, '...'
postfixlsid	Localized value of postfix string.
showastooltip	Specifies whether the untruncated value is to be shown as a mouseover tooltip when the string gets truncated. Default = true.
wrapped	Specifies whether maxlen applies to the entire string (wrapped='false') or to each word in the string (wrapped='true'). For example, if maxlen='5' and wrapped='true', the string "hello world" would not be truncated, because the string can wrap between words and no word exceeds the length of 5. If wrapped='false', the string would be truncated after "hello".

---

## symbolicversionlabelformatter

**Purpose** Validates a version label and invalidates a symbolic version label

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

controltovalidate	(Required) The name of the control to validate
nlsid	ID of NLS text to use for validation failure (overrides errormessage)

---

## symbolicversionlabelvalidator

**Purpose** Ensures that the symbolic version label entered by the user is not an implicit version label

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

controltovalidate	(Required) The control that is validated by this validator
-------------------	--

---



# tab

**Purpose** Generates a tab control within a tabbar control. The name attribute is required.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

value	Text that is displayed on the tab
datafield	Overrides value and NLS ID with data from a data source
nlsid	ID of NLS text to use (overrides value)
width	Sets the width of the tab and overrides the tabwidth property in tabbar.
height	Sets the height of the tab and overrides the tabheight property in tabbar.
onchange	Specifies the server-side event that is fired after the user changes the tab. The event is fired after the form has been submitted.
onclick	(Required) Event fired when the control has been clicked
runatclient	Set to true to handle the onclick event on the client

# tabbar

**Purpose** Generates a tab bar control (non-graphical)

## Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

value	Text displayed on the selected tab. Sets the name of the initial tab.
normalstyle	Style to apply to text of normal tabs
selectedstyle	Style to apply to text of selected tabs
disabledstyle	Style to apply to text of disabled tabs
normalclass	Class to apply to text of normal tabs
selectedclass	Class to apply to text of selected tabs
disabledclass	Class to apply to text of disabled tabs
tabposition	Sets where the tabs will be on the page. Valid values: top, bottom, left, right. Default = top
align	Sets the alignment of the text in all of the tabs. Valid values: left, right, or center (default = center)
tabwidth	Sets the width for all the tabs (default = 60). Can be overridden by the width property on tab.
tabheight	Set the height for all the tabs (default = 10). Can be overridden by the height property on tab.
imagefolder	The folder that the tab bar graphics are stored in. If the graphics are not found in this folder, a non-graphical tab bar will be drawn.
onclick	Event fired when the control has been clicked
onchange	Specifies the server-side event that is fired when a tab has been selected. This event is fired after the form is submitted.

---

showtabs	Set to true to render tabs (default). Set to false to display non-clickable, non-graphic "roadmap" for use in wizard containers only. If this attribute is set to false, the following attribute values will be ignored for rendering: tabposition, imagefolder, onclick.
scrollable	Set to true to enable scrolling through tabs if all tabs are not visible within the browser page. Supported only for tabposition=top, both graphical and text. Container must define a width (paneset, table). Scrolling arrow (chevrons) images can be placed in tabbar image folder for a theme. Default = false
runatclient	Set to true to handle the onclick event on the client

---

## Usage

You can add scrolling arrows (chevrons) to a theme. The images must be in the folder `/theme_name/images/tabbar/top` where `theme_name` is the name of your theme. The images must have the following names: `backward_enabled.gif`, `backward_disabled.gif`, `forward_enabled.gif`, and `forward_disabled.gif`.

**Note:** To have a scrollable tabbar, the containing HTML element must have a fixed width. For example, the containing table cells in Webtop's `tabbar.jsp` set fixed widths as follows:

```
<td align="left" valign="bottom" nowrap="">
  <dmf:tabbar name="<%=TabBar.CONTROL_TABBAR%>"...>
  </dmf:tabbar>
</td>
<td ...width="1%">
  <!-- label here -->
</td>
```

The Webtop streamline view has its own set of images. The images must be in the folder `/theme_name/imagesstreamline/tabbar/top` where `theme_name` is the name of your theme. The images must have the following names: `backward_enabled.gif`, `backward_disabled.gif`, `forward_enabled.gif`, and `forward_disabled.gif`.

## table

**Purpose** Generates HTML <table> open and close tags

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

datapagesize	Set the HTML datapagesize attribute on the table tag
--------------	--

---

## td

**Purpose** Generates HTML <td> (table cell) open and close tags

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

HTML attributes

## termshighlightingformatter

**Purpose** Formats a document by highlighting matching terms (exact matching and case insensitive). Ignores text between "<>" characters.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

controltovalidate	(Required) The name of the control to validate
nlsid	ID of NLS text to use for validation failure (overrides errormessage)
datafield	Overrides the values of the terms attribute
terms	List of terms to be highlighted
separator	String to be used between two matching terms

---

## text

**Purpose** Renders an HTML text box. The name attribute is required.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

size	Length of the text control
value	String that is displayed in the text box
datafield	Overrides value and nlsid with a string from a data source
nlsid	ID of NLS text to use (overrides value)
onchange	Specifies the server-side event handler for the change event, to be fired after the form has been submitted

defaulttonenter	If true, the keyboard enter keypress calls the JavaScript handleKeyPress() function. Must be used with a button on the page whose default attribute is set to true. The button event will be fired when the user hits the <b>Enter</b> key in the defaulttonenter control
runatclient	Behavior unknown

---

## textarea

**Purpose** Generates an HTML multi-line text box. The name attribute is required.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

rows	Number of rows in the text area control
wrap	True to set word-wrapping for display and submitted data from the textarea field.
cols	Number of columns in the text area control
value	Value displayed in the text area box
datafield	Overrides value and NLS ID with a value from a data source
nlsid	ID of NLS text to use (overrides value)
onchange	Specifies the server-side onchange event handler fired after the form is submitted
runatclient	Behavior unknown

---

---

## th

**Purpose** Generates HTML <th> (table head) open and close tags.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

HTML attributes

## thumbnail

**Purpose** Displays a small (maximum 100x100 pixel) thumbnail image generated and stored on the thumbnail server integrated with a Content Server

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

---

objectid	ID of the object to find the thumbnail image for.
datafield	Overrides objectid with data from a data source
alttext	Specifies alternate text to be displayed on browser mouseover
label	Overrides alttext
nlsid	NLS lookup key that substitutes a string for the label (overrides alttext)

width	Width of the image control
height	Height of the image control
border	Sets a border (in pixels) around the image if it is a link. Default = 0 pixels (no border).
hspace	Sets a horizontal margin (white space padding to the left and right) around the image.
vspace	Sets a vertical margin (white space padding to the top and bottom) around the image.
onclick	When a value for onclick is present, the image is turned into a link.

---

## title

**Purpose** Generates HTML <title> open and close tags

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

This tag has no configurable attributes

## topic

**Purpose** Displays the threaded list of comments found for an object's topic. The control should contain the comment control.



## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

This control has no configurable attributes.

## Usage

The control data is set at runtime from a component by setting its dataprovider with a `TopicResultSet` of data that makes up the topic. For example:

```
public void setTopicControl(
    ITopic topic, String[] attributes)
{
    topicData = new TopicResultSet(attributes, topic);
    Topic topicControl = (Topic)getControl(
        CONTROL_TOPIC, Topic.class);
    DataProvider dataProvider = topicControl.getDataProvider(
    );
    dataProvider.setScrollableResultSet(topicData);
}
```

# topicpanel

**Purpose** Hides a topic for repositories that do not have Collaborative Edition enabled

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

This control has no configurable attributes.

## Usage

To display a panel of topics, add the topicpanel and collaboration component similar to the following example:

```
<dmfx:topicpanel>
  <dmfx:componentinclude name='showtopic' component='
    embeddedtopic'>
    <dmfx:argument name="objectId" contextvalue='objectId' />
  </dmfx:componentinclude>
</dmfx:topicpanel>
```

# topicstatus

**Purpose** Generates an action icon that indicates the status of a discussion topic. A topic may be read, unread, or nonexistent. If an indicator icon is present, then clicking it will bring user to the page containing the topic. The control is used within the datagrid control and appears as a preference column that can be turned on or off.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

align	(Image action buttons only) Sets the alignment of the button text (default = center)
label	Optional label to be rendered in addition to the image
datafield	Overrides the label value with a value from a data source
nlsid	ID of NLS text to use as the label value (overrides datafield and label)
width	(Image action buttons only) Sets the width of the button

---

height	(Image action buttons only) Sets the height of the button
src	URL to the image
srcdisabled	Image to be used when the action is disabled
datafield	Overrides the URL defined in src attribute
datafielddisabled	Overrides srcdisabled value
alttext	Text that will be displayed by mouse hover
action	Specifies the ID of an action to be launched by the link

---

## tr

**Purpose** Generates HTML <tr> (table row) open and close tags

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

HTML attributes

## transformationparameter

**Purpose** Displays parameters for media profiles (dm\_media\_profile type) such as ike font size, color, width, and height. Media profiles are located in /System/Media Server in repositories that have installed Media Transformation Services.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

indentamount	Specifies the indent amount in pixels for child comments. Default = 32
cssclass	The CSS class to be used for previously read comments
unreadstyle	Sets the CSS style to be used for unread comments. By default it uses defaultUnreadCommentStyle

---

## Usage

The control data is set at runtime from a component by setting its dataprovider with a TopicResultSet of data that makes up the topic. For example:

```
public void setTopicControl(
    ITopic topic, String[] attributes)
{
    topicData = new TopicResultSet(attributes, topic);
    Topic topicControl = (Topic)getControl(
        CONTROL_TOPIC, Topic.class);
    DataProvider dataProvider = topicControl.getDataProvider(
    );
    dataProvider.setScrollableResultSet(topicData);
}
```

# tree

**Purpose** Generates a tree control. The name attribute is required.

## Tag library descriptor

dmf (dmform\_1\_0.tld)

## Attributes

---

value	Value stored in the control (stores the name of the selected tree item)
width	The width of the entire tree control. Puts the tree in an IE Iframe of the specified width and height or, if no Iframe is used, renders the tree inline. If you set width, you must also set height.
height	The height of the entire tree control. Puts the tree in an IE Iframe of the specified width and height. If you set height, you must also set width.
nodewidth	The width of an individual node graphic (default = 17). If node images exist, default width = the width of Tplus.gif.
nodeheight	The height of an individual node graphic (default = 17). If node images exist, default height = height of Tplus.gif.
showroot	Determines whether the root node is displayed (default = true)
showicons	Determines whether icons associated with each tree item are shown (default = false)
showlines	Determines whether tree lines are shown (default = true)
cachechildren	Determines whether children are cached when nodes are collapsed (default = true)
baseimagefolder	The location of the tree images (default = /wdk/theme/)
normalstyle	An entire CSS-syntax style sheet rule enclosed in quotes and applied to a normal tree item
selectedstyle	An entire CSS-syntax style sheet rule enclosed in quotes and applied to a selected tree item
disabledstyle	An entire CSS-syntax style sheet rule enclosed in quotes and applied to a disabled tree item
normalclass	Class to apply to text of normal rows

selectedclass	Class to apply to text of selected rows
disabledclass	Class to apply to text of disabled rows
frameborder	When width and height are specified to put the tree in an iFrame, this attribute specifies whether a frame border is required (default = true).
onclick	Event fired when a tree item has been clicked
onexpand	Client event fired when the user attempts to expand a node. If you override this node, then you need to write code to expand the appropriate node.
oncollapse	Client event fired when the user attempts to collapse a node. If you override this node, then you need to write code to collapse the appropriate node.
runatclient	Set to true to handle the onclick event on the client
multiselect	Set to true to enable expansion of multiple nodes in the tree. The ctrl key adds the clicked node to or removes it from the selection. The shift key selects all the nodes between the last selected node and the one clicked. A parent and child cannot be selected at once.
multiselectdepth	Specifies the level in the tree at which multiple nodes can be expanded. For example, a level of 2 prevents multiple repositories from being expanded in the tree, level 1 being repositories.
ondrop	Unsupported
dragenabled	Unsupported
inlinerefresh	Set to true to cause partial refresh of tree rather than full when user expands or collapses a node

---

For more information about drag and drop, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

## userstatevalueformatter

**Purpose** Formats the user state value from an int to a string: 0 = inactive | 1 = active

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

This formatter has no configurable attributes.

## utf8stringlengthvalidator

**Purpose** Verifies that the string in its UTF8 representation does not exceed the field limit, to prevent data loss when accented or double-byte characters are entered. Converts the user input string value to its corresponding UTF-8 byte array before comparing it with the specified maximum byte length value

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

None

### Usage

This validator can be used to replace the `regexpvalidator` to validate that string character length confirms to the field size constraint in the data dictionary. For multi-byte and accented characters, the `regexpvalidator` cannot do this validation and can result in data loss.

For example, the checkin page allows the user to enter a version label that, according to the data dictionary, must be no longer than 32 single-byte characters. This validator will not allow a version label that exceeds 32 bytes.

## validationsummary

**Purpose** Displays a summary list of failed validation messages for all validated input controls

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

header	Summary header message
nlsid	ID of summary header message string

---

## varequiredfieldvalidator

**Purpose** Verifies that the target control contains a value. Use for non-repeating attributes that have value assistance.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

None



## vdmbindingruleformatter

**Purpose**        Formats a virtual document binding rule

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

vdmtreegridname	Specifies the name of thevdmtreegrid control for which the binding rule output is formatted
-----------------	---

## vdmtreegrid

**Purpose**        Generates a grid display of a virtual document tree or snapshot (assembly). The grid is a data container similar to the data grid and must contain vdmtreegridrow tags to display the data.

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

cellspacing	Spacing between the cells in the grid
cellpadding	Spacing between the cell content and cell border
width	The width of the entire tree control. Using this with height puts the tree in an IE Iframe of the specified width and height.
height	The height of the entire tree control. Using this with width puts the tree in an IE Iframe of the specified width and height.
bordersize	Size of the datagrid border
showroot	Determines whether the root node is displayed (default = true)
paged	Data is displayed in pages
pagesize	Number of objects per page, if pagesize is set to true
vdmid	ID of the virtual document root node
expanddepth	Sets the number of node levels to expand (integer)

---

## vdmtreegridrow

**Purpose** Generates a table row for each node in a virtual document

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

width	The width of the row
height	The height of the row

---

align	Horizontal alignment of the row
valign	Vertical alignment of the row
altstyle	Inline CSS rule for alternating rows
altclass	CSS class for alternating rows
showindent	Sets the display to indent based on node level (outline style). True to indent. Default = false.
showplusminus	Sets the display to show +/- symbols for nodes that can be expanded or contracted. Default = false.

---

## webform

**Purpose** Outermost tag containing all other taglib tags on a page. This represents the entire HTML page. Generates included JavaScript file references.

### Tag library descriptor

dmf (dmform\_1\_0.tld)

### Attributes

---

formclass	Java class containing the event handler functions that are called when events are fired. If the JSP page is in a component, the component class handles events and the formclass should not be specified.
nlsbundle	NLS bundle that is used by all the controls in the form. If the JSP page is in a component, the component class specifies the nlsbundle.

nlsclass	Java NLS class that is used by all the controls in this form. If the JSP page is in a component, the component class specifies the nlsclass.
validation	Set to false to prevent validation of controls on form, for development only (default = true)

---

## Usage

By default, validation is performed on all validated controls in the form when a server-side action event is fired on a form. You can override form validation by adding the webform tag attribute "validation" and setting it to false. (By default this attribute has a value of true.) You may want to do this if validation is slowing down the UI redraw or if all events do not need validation. For example:

```
<dmf:webform formclass="com.documentum..."  
  validation="false"/>
```

# workflowdisabledformatter

**Purpose** Wraps the integer that signifies the workflow in or out

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

This formatter has no specific configurable attributes.

## workflowdurationformatter

**Purpose** Wraps the integer that signifies the workflow duration

**Tag library descriptor**

dmfx (dmformext\_1\_0.tld)

**Attributes**

This formatter has no specific configurable attributes.

## workflowcompletetypeformatter

**Purpose** Wraps a value for the integer that signifies the type of workflow completion

**Tag library descriptor**

dmfx (dmformext\_1\_0.tld)

**Attributes**

This formatter has no specific configurable attributes.

## workfloweditorapplet

**Purpose** Generates an applet that displays the workflow editor

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

Has the [Table 2-4, page 16](#) and the following additional attributes:

---

vspace	Vertical space between applet and surrounding text
hspace	Horizontal space between applet and surrounding text
viewonly	All editing controls are disabled. The user can perform <b>File New</b> , <b>File Open</b> , and <b>File Save As</b> .
tracelevel	Set to 0 to turn off tracing, 1 to turn it on
successhandler	Server side event handler that handles the applet return

---

# workflowstateicon

**Purpose**      Obtains an icon that represents the workflow state of a Documentum object

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

statedatafield	Overrides state value. Specifies a datafield that supplies the state for the object, for example, statedatafield='w.r_runtime_state'.
state	Sets the object state
size	Size of the icon in pixels

---

## workflowtaskstatusvalueformatter

**Purpose** Wraps a workflow task status value

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

This formatter has no specific configurable attributes.

## workqueueflagvalueformatter

**Purpose** Wraps a work queue flag value (isAssigned)

### Tag library descriptor

dmfx (dmformext\_1\_0.tld)

### Attributes

This formatter has no specific configurable attributes.

## workqueuetaskstatusvalueformatter

**Purpose** Wraps a work queue task status value

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

This formatter has no specific configurable attributes.

# xforms

## Purpose

Renders a form for viewing or editing using the Chiba open source xforms engine. The application server must have Java 1.4.x to display forms. The user must have the role form\_user to view or edit a form. Forms are designed through the Documentum Forms Builder product.

## Tag library descriptor

dmfx (dmformext\_1\_0.tld)

## Attributes

---

readonly	Set to true to render the form for viewing only.
xformsconfigid	Specifies an ID that corresponds to an xforms configuration
formstemplate	ID of the template
objectid	Supplies an object ID for the form

---

## Usage

When a form is saved, the control passes the form content to the Chiba engine for validation. Errors are passed to the form. Each button on the form has a configurable event that is fired when the button is clicked. The events are configured in an XML Configuration file whose ID is specified in the control xformsconfigid attribute.



---

If a component must respond to a button event, the component should register an `XFormsControlListener` and set the control event handler. The listener is required if the component would like to listen to events that the control fires. The events are named on *TriggerId* where *TriggerId* is the trigger name.

Your component should add the listener in the `onControlInitialized()` method of the component class. For example, if the Xforms Configuration file specifies a trigger listener for the Submit button with the ID `submit-trigger`, the component can register an event that responds to this trigger as follows:

```
public void onControlInitialized(Form form, Control control)
{
    if (control instanceof XForms)
    {
        XForms value = (XForms)control;
        value.setEventHandler("onsubmit-trigger", "onSubmitTrigger", form);
        addControlListener(new XFormsControlListener());
    }
}
public void onSubmitTrigger(Control control, ArgumentList arg)
{
    //handle event here
}
```

The XForms Configuration file is a Chiba Configuration file. For more information on Chiba configuration, see [the Chiba Web site](#).



## Actions Reference

### abortfailedautoworkflowtask (type dm\_task)

**Purpose** Aborts a failed automatic workflow task.

#### Configuration File

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

#### Scope

---

type	dm_task
------	---------

---

#### Caller

taskmanagercontainer

#### Parameters

---

canCompleteTask	(Optional) Specifies that the task is active and can be completed.
isFailedAutoTask	(Optional) Specifies that an automatic task has failed and can be re-run.
objectId	(Required) ID of the task.

userCost	(Optional) Customer-defined cost data, stored as double in Server. Can be audited and summed up for completed workflows.
userTime	(Optional) Customer-defined time data, stored as integer in Server. Can be audited and summed up for completed workflows.

---

## abortjob

**Purpose** Aborts an asynchronous action or component job.

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

### Caller

jobstatus

### Parameters

---

jobId	(Required) ID of the job to abort.
-------	------------------------------------

---

## abortworkflow (type dm\_workflow)

**Purpose** Aborts a workflow.

### Configuration File

webcomponent\config\actions\dm\_workflow\_actions.xml

### Scope

type	dm_workflow
------	-------------

### Caller

reportmainlist, reportdetailsummarylist, reportdetailsummarydrilldown, and menubar components

### Parameters

objectId	(Required) ID of the workflow.
supervisor	(Optional) User name of workflow supervisor
workflowRuntimeState	(Required) State of the workflow. 0=dormant   1=running   2=finished   3=halted   4=terminated

## acceptworkflowtask (type dm\_task)

**Purpose** Accepts a workflow task for the user.

**Configuration File**

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

**Scope**

---

type	dm_task
------	---------

---

**Caller**

taskmanagercontainer

**Parameters**

---

canAcquireTask	(Optional) Boolean flag that indicates that the task is in a state such that it can be acquired (not paused, already acquired, or finished)
isGroupOwned	(Optional) Boolean flag that indicates that the task is assigned to a group of which the user is a member
objectId	(Required) ID of the task.

---

## accessibilityaction

**Purpose** Handles accessibility option for buttons and menu items in a component.

**Configuration File**

webcomponent\config\actions\generic\_actions.xml

**Scope**

All.

**Caller**

WDK Component class

**Parameters**


---

accessible	(Optional) Set to true to call the menubar component in accessible mode
------------	---

---

**accessibilityaction (type dm\_acedm\_requiredgroupdm\_requiredgroupsetdm\_restriction)****Purpose** Not used.**Configuration File**

webcomponent\config\actions\dm\_accessor\_actions.xml

**Scope**


---

type	dm_ace, dm_requiredgroup, dm_requiredgroupset, dm_restriction
------	---

---

**Caller**

Not used.

**Parameters**


---

accessible	(Optional) Not used.
accessorName	(Required) Not used.
aceavailableactions	(Optional) Not used.

basicpermit	(Optional) Not used.
extpermit	(Optional) Not used.
extpermit_readonly	(Optional) Not used.
objectId	(Optional) Not used.
objectName	(Optional) Not used.
permit	(Optional) Not used.
permitType	(Optional) Not used.

---

## aclbrowse (privilege sysadmin, type dm\_user)

**Purpose** Allows the administrator to browse a list of ACLs in the repository

### Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

### Scope

---

privilege	sysadmin
type	dm_user

---

### Caller

userattributes and userimport components

## addaceaction (type dm\_ace)

**Purpose** Adds a user or group entry to an ACL



**Configuration File**

webcomponent\config\admin\acl\acl\_actions.xml

**Scope**


---

type	dm_ace
------	--------

---

**Caller**

aclacelist and permissions components

**Parameters**


---

flatlist	(Optional) Boolean: True to list all objects in a flat list, false to display a drop down list.
multiselect	(Required) Boolean: True to support multiple object selection
objecttype	(Optional) Type of the selected object
permit	(Optional) Set of working and new permits. Valid values: DELETE_PERMIT   WRITE_PERMIT   VERSION_PERMIT   RELATE_PERMIT   READ_PERMIT   BROWSE_PERMIT   NONE_PERMIT
roomId	(Optional) ID of the room, in the case of a room ACL

---

**addattachment (type dmi\_package)**

**Purpose** Adds an attachment to a workflow package

## Configuration File

webcomponent\config\actions\dm\_package\_actions.xml

## Scope

---

type	dm_package
------	------------

---

## Caller

taskattachment component

## Parameters

---

formTemplateId	(Optional) ID of the XForm template
objectId	(Required) Task ID
objecttype	(Optional) Type of the attachment object
packageName	(Required) Name of the workflow package
runtimeState	(Required) State of the workflow. 0=dormant   1=running   2=finished   3=halted   4=terminated. Only 0 or 1 can be suspended.
versionlabel	(Optional) Version label for the attachment

---

# addcomponent (type dm\_sysobject)

**Purpose** Adds a child node to a virtual document. The child object must be in the same repository.

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**


---

type	dm_sysobject
------	--------------

---

**Caller**

addnewvirtualdocumentnode component class

**Parameters**


---

compoundArchitecture	(Optional) Boolean specifying whether the object is a compound virtual document
insertAfterNodeId	(Optional) ID of the virtual document node after which the new node is added
lockOwner	(Optional) User who has checked out the root object
newNodeObjectIds	(Required) IDs of the objects to be added to the virtual document.
nodeId	(Optional) Node ID of the parent node for the new object.
objectId	(Required) Object ID of the parent node for the new object.
preconditionShouldDisplayErrors	Set to true to display errors when the precondition criteria are not met
vdmRootObjectId	(Optional) Object ID of the root of the virtual document.

---

**addcomponentfromclipboard (type dm\_sysobject)**

**Purpose** Adds a child node to a virtual document from the clipboard

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

menubar and component pages that contain actionlinklist tags

## Parameters

---

compoundArchitecture	(Optional) Boolean specifying whether the object is a compound virtual document
lockOwner	(Optional) User who has checked out the root object
nodeId	(Optional) ID of the object to be added as a child node
objectId	(Required) Parent node.
vdmRootObjectId	(Optional) Object ID of the root of a virtual document (when operating on a child node).

---

# addcomponentfromfileselector (type dm\_sysobject)

**Purpose** Adds a child node to a virtual document from the file selector.

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar and component pages that contain actionlinklist tags

**Parameters**

compoundArchitecture	(Optional) Boolean specifying whether the object is a compound virtual document
lockOwner	(Optional) User who has checked out the root object
nodeId	(Optional) ID of the object to be added as a child node
objectId	(Required) Parent node.
vdmRootObjectId	(Optional) Object ID of the root of a virtual document (when operating on a child node).

**addgroupmembership (type dm\_ace)**

**Purpose** Adds a user or group to a group

**Configuration File**

webcomponent\config\admin\acl\acl\_actions.xml

**Scope**

type	dm_ace
------	--------

**Caller**

aclacelist and aclproperties components

## Parameters

---

accessorName	(Required) Name of user or group
objectId	(Optional) ID of the ACL (permission set)

---

## addgroupmembership (type dm\_restriction)

**Purpose** Adds a user or group to a group that has restricted access. Trusted Content Services must be enabled in the repository.

## Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

## Scope

---

type	dm_restriction
------	----------------

---

## Caller

aclacelist and aclproperties components

## Parameters

---

accessorName	(Required) Name of user or group
objectId	(Optional) ID of the ACL (permission set)

---

## addnewvirtualdocumentnode (type dm\_sysobject)

**Purpose** Create content and add it to a virtual document

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

type	dm_sysobject
------	--------------

## Caller

menubar and component pages that contain actionlinklist tag

## Parameters

compoundArchitecture	(Optional) Boolean specifying whether the object is a compound virtual document
lockOwner	(Optional) User who has checked out the object
nodeId	(Optional) Node ID of the parent for the new content.
objectId	(Required) Object ID of the parent for the new content.
ownerName	(Optional) Owner of the object, evaluated by the role precondition
vdmRootObjectId	(Optional) Object ID of the root of the virtual document.

# addprocessorauthentication

**Purpose** Sets the skill level and authorized document profiles for a new work queue processor

### Configuration File

webcomponent\config\library\queuemanagement\userprofile\_actions.xml

### Scope

All.

### Caller

processorattributes component

### Parameters

---

multiselect	(Optional) Set to true to allow multiple selection
-------------	--

---

## addrequiredgroup (type dm\_requiredgroup)

**Purpose** Adds a group of users to an ACL. A user must belong to that group in order to satisfy any security check with that ACL (Requires that Trusted Content Services be enabled in the repository.)

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

---

type	dm_requiredgroup
------	------------------

---

### Caller

aclacelist component



## Parameters

flatlist	(Optional) Boolean: True to list all objects in a flat list, false to display a drop down list.
multiselect	(Required) Boolean: True to support multiple object selection
objecttype	(Optional) Type of the selected object
permit	(Optional) Set of working and new permits. Valid values: DELETE_PERMIT   WRITE_PERMIT   VERSION_PERMIT   RELATE_PERMIT   READ_PERMIT   BROWSE_PERMIT   NONE_PERMIT
roomId	(Optional) ID of a room, in the case of a room ACL

## addrequiredgroupset (type dm\_requiredgroupset)

**Purpose** Creates a set of groups. A user must belong to one of the groups in order to satisfy any security check with that ACL (Requires that Trusted Content Services be enabled in the repository.)

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

type	dm_requiredgroupset
------	---------------------

### Caller

aclacelist component

## Parameters

flatlist	(Optional) Boolean: true to list all objects in a flat list, false to display a drop down list.
multiselect	(Required) Boolean: true to support multiple object selection
objecttype	(Optional) Type of the selected object
permit	(Optional) Set of working and new permits. Valid values: DELETE_PERMIT   WRITE_PERMIT   VERSION_PERMIT   RELATE_PERMIT   READ_PERMIT   BROWSE_PERMIT   NONE_PERMIT
roomId	(Optional) ID of the room, in the case of a room ACL

## addrrestriction (type dm\_restriction)

**Purpose** Adds a restriction to the ACL. Trusted Content Services must be enabled in the repository.

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

type	dm_restriction
------	----------------

### Caller

aclacelist component

## Parameters

flatlist	(Optional) Boolean: true to list all objects in a flat list, false to display a drop down list.
multiselect	(Required) Boolean: true to support multiple object selection
objecttype	(Optional) Type of the selected object
permit	(Optional) Set of working and new permits. Valid values: DELETE_PERMIT   WRITE_PERMIT   VERSION_PERMIT   RELATE_PERMIT   READ_PERMIT   BROWSE_PERMIT   NONE_PERMIT
roomId	(Optional) ID of the room, in the case of a room ACL

## addtoclipboard (type dm\_sysobject)

**Purpose** Adds an object to the clipboard.

### Configuration File

webcomponent\config\environment\clipboard\dm\_sysobject\_clipboard\_actions.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

menubar and toolbar components, and component pages that contain actionlinklist tag

## Parameters

---

isReference	(Optional) Boolean flag that indicates whether the object is a reference object
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the object to be added to the clipboard.
ownerName	(Optional) Name of owner of the object to be added to the clipboard
type	(Optional) (Required) The content type of the object.

---

## Elements

---

<precondition>.<role>	Value: Role that can perform move, copy, or link operations. Default = contributor
-----------------------	--

---

# addtoclipboard (type dmc\_workqueue\_category)

**Purpose** Adds a work queue to the queue management clipboard.

## Configuration File

webcomponent\config\actions\dmc\_workqueue\_category\_actions.xml

## Scope

---

type	dmc_workqueue_category
------	------------------------

---

**Caller**

menubar, toolbar, and any component containing actionlinklist tag

**Parameters**

---

objectId	(Required) ID of the work queue to add to the clipboard.
ownerName	(Optional) Name of the user who owns the work queue to be added to the clipboard.
type	(Optional) Content type of the object to be added to the clipboard

---

**addtoclipboard (type dmc\_workqueue)**

**Purpose** Adds an object to the clipboard

**Configuration File**

webcomponent\config\actions\dmc\_workqueue\_actions.xml

**Scope**

---

type	dmc_workqueue
------	---------------

---

**Caller**

Menu item in menubar component, and toolbar item in Webtop toolbar component

## Parameters

---

objectId	(Required) ID of the object to be added to the clipboard
ownerName	(Optional) Name of the user who owns the object
type	(Optional) Content type of the object to be added to the clipboard

---

# adduserorgroup (type grouplist)

**Purpose** Adds a user or group to the repository

## Configuration File

webcomponent\config\admin\group\grouplist\_actions.xml

## Scope

---

type	grouplist
------	-----------

---

## Caller

menubar component

## Parameters

---

flatlist	(Optional) Boolean: true to list all objects in a flat list, false to display a drop down list.
multiselect	(Optional) Boolean: true to support multiple object selection
objecttype	(Optional) Type of the selected object

---

## addwfattachment

**Purpose** Adds an ad hoc attachment to a workflow

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

### Caller

startwfattachment and taskattachment components

## addworkqueuemember

**Purpose** Adds a user as a valid processor for a work queue

### Configuration File

webcomponent\config\library\queuemanagement\queueusersandgroups\_actions.xml

### Scope

All.

### Caller

menubar component

## Parameters

---

addQueueMemberContext	(Required) Boolean that must have a value of true to perform action
groupName	(Optional) Must have a value of queue_processor to perform action
multiselect	(Optional) Set to true to allow multiple selection
objectId	(Optional) ID of work queue
queueName	(Required) Name of work queue

---

## adminbrowse (type dm\_group)

**Purpose** Allows the administrator to browse a list of group administrators in a repository.

### Configuration File

webcomponent\config\admin\group\dm\_group\_actions.xml

### Scope

---

type	dm_group
------	----------

---

### Caller

groupattributes component

## aliasbrowse (privilege sysadmin, type dm\_user)

**Purpose** Allows the administrator to browse a list of alias sets in the repository.



**Configuration File**

webcomponent\config\admin\user\dm\_user\_actions.xml

**Scope**

privilege	sysadmin
type	dm_user

**Caller**

groupattributes, userattributes, and userimport components

**aliasetbrowse (type dm\_group)**

**Purpose** Allows the administrator to browse a list of alias sets in the repository.

**Configuration File**

webcomponent\config\admin\group\dm\_group\_actions.xml

**Scope**

type	dm_group
------	----------

**Caller**

groupattributes, userattributes, and userimport components

## appintgnewfromtemplate

**Purpose** Creates a new document from a template, called from within a Windows application (Application Connectors)

### Configuration File

webcomponent\config\actions\appintg\_actions.xml

### Scope

All.

### Caller

Application Connectors client

### Parameters

---

contentType	(Optional) File format
-------------	------------------------

---

## appintgopenfrom

**Purpose** Locates a document in a repository to open within a Windows application (Application Connectors)

### Configuration File

webcomponent\config\actions\appintg\_actions.xml

**Scope**

All.

**Caller**

Application Connectors client

**Parameters**

---

contentType	(Optional) Type of document
folderId	(Optional) ID of folder in which the document is located
launchViewer	(Optional) Boolean flag that indicates whether to launch a viewing application
locatorcomponent	(Optional) ID of the locator component to use for locating the document, for example, myfiles_streamline

---

## appintgsaveas

**Purpose**

Performs an import or checkin of current document from within a Windows application (Application Connectors)

**Configuration File**

webcomponent\config\actions\appintg\_actions.xml

**Scope**

All.

## Caller

Application Connectors client

## Parameters

---

contentType	(Optional) Type of document
filenamewithpath	(Required) Name and path of the file to be checked in.
folderId	(Optional) Location of the current document. User can then choose a new file name to import to that folder or an existing document to check in to the folder, or navigate from the folder
locatorcomponent	(Optional) Locates the destination folder for import. Specify the component to start with (for example, myfiles_streamline)

---

# applylifecycle (type dm\_sysobject)

**Purpose** Attaches a lifecycle (business policy) to an object

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

menubar and components that have actionlinklist tag

**Parameters**

---

isReplica	(Optional) Boolean indicating whether the object is a replica; the action is invalid for replicas
lockOwner	(Optional) User who has checked out the selected object
objectId	(Required) The object to which a lifecycle will be applied.

---

**assemblylist (type dm\_sysobject)**

**Purpose** Displays the structure of a virtual document snapshot

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Parent Definition**

viewassembly:webcomponent/config/actions/dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

Webtop classic.jsp page

## assemblyliststreamline (type dm\_sysobject)

**Purpose** Displays the structure of a virtual document snapshot in drilldown (streamline) view

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Parent Definition

streamlineviewassembly:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

Called indirectly from several component classes

## assign\_queued\_task (type dm\_task)

**Purpose** Assigns a task from a work queue to a particular performer

### Configuration File

webcomponent\config\actions\dm\_queued\_task\_actions.xml

**Scope**

type	dm_task
------	---------

**Caller**

menubar component

**Parameters**

objectId	(Required)
ownerName	(Optional) Task owner
performer	(Required) Name of the task.
queueName	(Required) Name of the work queue.

**attributes (type dm\_acl)**

**Purpose** Displays the attributes of an ACL

**Configuration File**

webcomponent\config\admin\acl\acl\_actions.xml

**Scope**

type	dm_acl
------	--------

**Caller**

acclist and aclwhereused components

## Parameters

---

objectId	(Required) ID of the ACL.
----------	---------------------------

---

## attributes (type dm\_externalresult)

**Purpose** Displays the attributes page for an external result object

## Configuration File

webcomponent\config\actions\dm\_externalresult\_actions.xml

## Scope

---

type	dm_externalresult
------	-------------------

---

## Caller

menubar

## Parameters

---

entryId	(Required) ID of the result entry.
queryId	(Required) ID of the query used for searching the external source.

---

## attributes (type dm\_group)

**Purpose** Displays the attributes for a repository group or creates a new group.



**Configuration File**

webcomponent\config\admin\group\dm\_group\_actions.xml

**Scope**

type	dm_group
------	----------

**Caller**

groupelist and groupwhereusedcomponents

**Parameters**

objectId	(Required) ID of the selected group. Set to newobject to create a new group.
objectname	(Required) Name of the group.

**attributes (type dm\_member\_group)**

**Purpose** Displays the attributes for an object.

**Configuration File**

webcomponent\config\admin\group\dm\_member\_actions.xml

**Scope**

type	dm_member_group
------	-----------------

### Parameters

---

objectId	(Required) ID of the selected object.
objectname	(Required) Name of the selected object.

---

## attributes (type dm\_member\_user)

**Purpose**      Displays the attributes for an object

### Configuration File

webcomponent\config\admin\group\dm\_member\_actions.xml

### Scope

---

type	dm_member_user
------	----------------

---

### Parameters

---

objectId	(Required) ID of the selected object.
objectname	(Required) Name of the selected object.

---

## attributes (type dm\_sysobject)

**Purpose**      Displays the attributes for an object.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

**Caller**

Formatlist, menubar, myobjects\_list, relationships, renditions, doclist, searchresults, subscriptions, viewassemblies components and more

**Parameters**

enableShowAll	(Optional) Determines whether the Show All checkbox is displayed. (true false)
objectId	(Required) ID of the selected object.

**Elements**

<failoverenabled>	This component supports failover. To disable failover, enter a value of false.
-------------------	--

**attributes (type dm\_user)**

**Purpose** Displays attributes for the selected user.

**Configuration File**

webcomponent\config\admin\user\dm\_user\_actions.xml

**Scope**

type	dm_user
------	---------

### Caller

Userlist and userwhereused components

### Parameters

---

objectId	(Required) ID of the selected user. Set to newobject to create a new user.
objectname	(Required) Name of the user.

---

## attributes (type dmc\_workqueue\_category)

**Purpose** Displays the attributes of the selected object.

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_category\_actions.xml

### Scope

---

type	dmc_workqueue_category
------	------------------------

---

### Caller

queuelist component

### Parameters

---

objectId	(Required) ID of the selected object.
objectname	(Required) Name of the selected object.

---

## attributes (type dmc\_workqueue\_doc\_profile)

**Purpose** Displays the attributes page for a work queue document profile object

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_doc\_profile\_actions.xml

### Scope

type	dmc_workqueue_doc_profile
------	---------------------------

### Caller

docprofilelist component

### Parameters

objectId	(Required) ID of the work queue document profile object.
objectname	(Required) Name of the work queue document profile.

## attributes (type dmc\_workqueue\_policy)

**Purpose** Displays the attributes page for a work queue policy

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_policy\_actions.xml

### Scope

---

type	dmc_workqueue_policy
------	----------------------

---

### Caller

policylist component

### Parameters

---

objectId	(Required) ID of the work queue policy object.
objectname	(Required) Name of the work queue policy object.

---

## attributes (type dmc\_workqueue)

**Purpose** Displays the attributes of a dmc\_workqueue object.

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_actions.xml

### Scope

---

type	dmc_workqueue
------	---------------

---

### Caller

queuemonitorlist component UI

**Parameters**

objectId	(Required) ID of the selected object.
objectname	(Required) Name of the selected object

**cancelcheckout (type dm\_process)**

**Purpose** Cancels the checked out status of an object. The user must have version level permission (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

**Configuration File**

webcomponent\config\actions\dm\_process\_actions.xml

**Parent Definition**

cancelcheckout:webcomponent/config/actions/dm\_sysobject\_actions.xml

**Scope**

type	dm_process
------	------------

**Caller**

menubar and toolbar, and component pages that contain actionlinklist tag

## cancelcheckout (type dm\_sysobject)

**Purpose** Cancels the checked out status of an object. The user must have version level permission (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

isReference	(Optional) Boolean flag that indicates whether the object is a reference object
lockOwner	(Optional) User who has checked out the object
nodeId	(Optional) Node ID of the object to check out, when checking out part of a virtual document.
objectId	(Required) ID of the object to be checked in.
vdmRootObjectId	(Optional) Object ID of the root object of a virtual document.

---



## changehomedblog (type dm\_user)

**Purpose** Displays user change home repository requests and their status.

### Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

### Scope

type	dm_user
------	---------

### Caller

menubar component

## changehomedocbase (privilege sysadmin, type dm\_user)

**Purpose** Changes the user's home repository.

### Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

### Scope

privilege	sysadmin
type	dm_user

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the selected object.
objectname	(Required) Name of the selected object.

---

## changehomedocbase (type dm\_member\_user)

**Purpose** Changes the user's home repository.

### Configuration File

webcomponent\config\admin\group\dm\_member\_actions.xml

### Scope

---

type	dm_member_user
------	----------------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the selected object.
objectname	(Required) Name of the selected object.

---

## changeperformer (type dm\_task)

**Purpose** Changes the performer of a future task in a workflow

### Configuration File

webcomponent\config\actions\dm\_task\_actions.xml

### Scope

type	dm_task
------	---------

### Caller

reportdetailssummarylist component

### Parameters

objectName	(Required) Activity name of the future task.
supervisor	(Optional) Workflow supervisor
taskRuntimeFlag	(Required) State of the workflow task. Valid values: 0=dormant   1=acquired   2=finished   3-6 =paused   7=ready
workflowId	(Required) ID of the workflow.
workflowRuntimeState	(Required) State of the workflow activity. Valid values: 0=dormant   1=running   2=finished   3=halted   4=terminated

## changesupervisor (type dm\_workflow)

**Purpose** Changes the supervisor of a workflow.

### Configuration File

webcomponent\config\actions\dm\_workflow\_actions.xml

### Scope

---

type	dm_workflow
------	-------------

---

### Caller

reportmainlist component

### Parameters

---

objectId	(Required) ID of the workflow.
workflowRuntimeState	(Required) State of the workflow activity. Valid values: 0=dormant   1=running   2=finished   3=halted   4=terminated

---

## checkin (type dm\_sysobject)

**Purpose** Checks in the selected object. The user must have version\_permit access to the object (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT). The user must be assigned to a contributor role defined in the repository.

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

menubar component

## Parameters

---

filenamewithpath	(Optional) Name and path of the file to be checked in.
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
lockOwner	(Optional) User who has checked out the object
nodeId	(Optional) ID of the repository node
objectId	(Required) ID of the object to be checked in.
ownerName	(Optional) User who owns the checked in object
vdmRootObjectId	(Optional) Object ID of the root for a virtual document, when checking in a virtual document

---

## Elements

The configurable elements for dynamic filters are described in [Dynamic component launching, page 9](#).

## checkout (type dm\_process)

**Purpose** Checks out the selected workflow template object. The user must have version level permission (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_process\_actions.xml

### Scope

---

type	dm_process
------	------------

---

### Caller

menubar and toolbar components and component pages that contain actionlinklist tag

### Parameters

---

isReference	(Optional) Boolean flag that indicates whether the object is a reference object
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the object to be checked out.
ownerName	(Optional) User who owns the object

---

## checkout (type dm\_sysobject)

**Purpose** Checks out the selected object. The user must have version level permission (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

menubar component

### Parameters

isIgnoreDescendants	(Optional) Boolean value that indicates whether or not to check out descendants in a virtual document.
isLockOnly	(Optional) Boolean flag that indicates whether the object should be locked but not copied to the checkout location
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
lockOwner	(Optional) User who has checked out the object
nodeId	(Optional) Node ID of the object to check out, when checking out part of a virtual document.

objectId	(Required) ID of the object to be checked out.
ownerName	(Optional) User who owns the checked in object
vdmRootObjectId	(Optional) Object ID of the root of a virtual document.

---

## Elements

See [Dynamic component launching, page 9](#). The following additional element is defined:

---

<execution>	Set enabled to true if your action class can process OLE
<olecompound>	compound documents

---

# comment (type dm\_process)

**Purpose** Adds a comment to a workflow template

## Configuration File

webcomponent\config\actions\dm\_process\_actions.xml

## Scope

---

type	dm_process
------	------------

---

## Caller

None



**Parameters**

objectId	(Required) ID of the template
startworkflowId	(Required) Component ID of the task manager component to be displayed as the first tab in the container.

**comment (type dm\_sysobject)**

**Purpose** Checks for a rendition in the allowable formats and generates JavaScript to invoke Adobe Acrobat 6 in a new browser window. The allowable formats for comments are specified in /webcomponent/app.xml. The Adobe Comment Connector feature must be enabled for the repository. This action does not check for the presence of Adobe Acrobat on the client machine.

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

**Caller**

PreferredRenditionsService

**Parameters**

contentType	(Required) Type of the selected object.
objectId	(Required) ID of the template

## comment (type dm\_task)

**Purpose** Adds a comment to a workflow task.

### Configuration File

webcomponent\config\actions\dm\_task\_actions.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

Not called

### Parameters

---

objectId	(Required) ID of the task
taskmanagerId	(Required) Component ID of the task manager component.

---

## commitchanges (type dm\_sysobject)

**Purpose** Asks the user whether to save changes to a virtual document before displaying it. Called by the vdmviewcomponent when a user views a virtual document that has been checked out.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**


---

type	dm_sysobject
------	--------------

---

**Caller**

Not called

**Parameters**


---

lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the template
ownerName	(Optional) Name of the person who created the document

---

**completefailedautoworkflowtask (type dm\_task)****Purpose** Completes a failed automatic workflow task.**Configuration File**

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

**Scope**


---

type	dm_task
------	---------

---

**Caller**

taskmanagercontainer UI

## Parameters

---

canCompleteTask	(Optional) Specifies that the task is active and can be completed.
isFailedAutoTask	(Optional) Specifies that an automatic task has failed and can be re-run.
objectId	(Required) ID of the task.
userCost	(Optional) Customer-defined cost data, stored as double in Server. Can be audited and summed up for completed workflows.
userTime	(Optional) Customer-defined time data, stored as integer in Server. Can be audited and summed up for completed workflows.

---

# configuredocbroker

**Purpose** Launches the configuredocbroker component in the dialogcontainer.

## Configuration File

wdk\config\preferences\_actions.xml

## Scope

All.

## Caller

Called from the visiblerepository\_preferences component UI as well as actionlink in the login page

---

## copy (type dm\_sysobject)

**Purpose** Copies a clipboard object to a new location

### Configuration File

webcomponent\config\environment\clipboard\dm\_sysobject\_clipboard\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Usage

Copy behavior can be configured in the <copy\_operation> element of app.xml. Set the value of the child element <retainstorageareas> to true to retain storage areas.

The abstract clipboard class ignores auxiliary files for the linked or copied object if they are not in the clipboard. The link or copy operation proceeds without the auxiliary files. Whether a linked file is auxiliary can be determined by calling isAuxiliaryObject(objectId).

## createhtmlrendition (type dm\_sysobject)

**Purpose** Creates an HTML rendition of the object.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

contentType	(Optional) File format to use when opening the object (passed to the operating system)
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the object.

---

## createhtmlrendition (type dmr\_content)

**Purpose**      Creates an HTML rendition of a document

### Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

### Scope

---

type	dmr_content
------	-------------

---

### Caller

menubar and component pages containing actionlinklist tag

**Parameters**

contentType	(Required) File format to use when creating the object (passed to the operating system)
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the selected object.

**createhtmlrenditiongeneric (type dm\_sysobject)**

**Purpose** Creates an HTML rendition of the object

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Parent Definition**

createhtmlrendition:application='webcomponent' type='dm\_sysobject'

**Scope**

type	dm_sysobject
------	--------------

**Caller**

Not called

## createhtmlrenditiongeneric (type dmr\_content)

**Purpose**        Creates an HTML rendition of a document

### Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

### Parent Definition

createhtmlrendition:application='webcomponent' type='dm\_sysobject'

### Scope

---

type	dmr_content
------	-------------

---

### Caller

None

## createpdfrendition (type dm\_sysobject)

**Purpose**        Creates a PDF rendition of the object.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---



**Caller**

menubar and renditions components

**Parameters**

contentType	(Optional) File format to use when opening the object (passed to the operating system)
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the object.

**createpdfrendition (type dmr\_content)**

**Purpose** Creates a PDF rendition of a document.

**Configuration File**

webcomponent\config\actions\dmr\_content\_actions.xml

**Scope**

type	dmr_content
------	-------------

**Caller**

menubar and component pages containing actionlinklist tag

## Parameters

---

contentType	(Required) File format to use when creating the object (passed to the operating system)
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the selected object.

---

# createpdfrenditiongeneric (type dm\_sysobject)

**Purpose**        Creates a PDF rendition of the object.

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Parent Definition

createpdfrendition:application='webcomponent' type='dm\_sysobject'

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

Not called

## createpdfrenditiongeneric (type dmr\_content)

**Purpose** Creates a PDF rendition of a document

### Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

### Parent Definition

createpdfrendition:application='webcomponent' type='dm\_sysobject'

### Scope

type	dmr_content
------	-------------

### Caller

None

## createrecordrelation (entitlement recordsmanager, isPolicyManaged true, type dm\_sysobject not dm\_folder)

**Purpose** Creates a relationship between formal records. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\records\relationships\dm\_sysobject\_actions.xml

## Scope

---

entitlement	recordsmanager
isPolicyManaged	true
type	dm_sysobject,not dm_folder

---

## Parameters

---

objectId	(Required) Object ID of the source formal record object.
----------	--

---

# **declareformalrecord (entitlement recordsmanager, role dmc\_rm\_recordscontributor, type dm\_sysobject not dm\_folder)**

**Purpose**      Creates formal records. Requires a Records Manager license.

## Configuration File

`webcomponent\config\library\records\declarerecord\dmc_rm_record_actions.xml`

## Scope

---

entitlement	recordsmanager
role	dmc_rm_recordscontributor
type	dm_sysobject, not dm_folder

---

## Caller

menubar component

**Parameters**

folderId	(Optional) ID of the folder in which to create the formal record object.
objectId	(Required) Object ID of the formal record object to create.

**Elements**

<precondition>. <role>	The role whose members can perform the action. Default = form_user.
---------------------------	---

**delegateworkflowtask (type dm\_task)**

**Purpose** Delegates a task in a workflow.

**Configuration File**

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

**Scope**

type	dm_task
------	---------

**Caller**

taskmanagercontainer component

## Parameters

---

canCompleteTask	(Optional) Specifies that the task is active and can be completed.
isDelegable	(Optional) True to specify that the task can be delegated
isFailedAutoTask	(Optional) Specifies that an automatic task has failed and can be re-run.
objectId	(Required) ID of the task.

---

# delete (entitlement collaboration, type dmc\_notepage)

**Purpose** Deletes a dmc\_notepage object. Requires a Collaborative Edition license.

## Configuration File

webcomponent\config\library\notepage\dmc\_notepage\_actions.xml

## Scope

---

entitlement	collaboration
type	dmc_notepage

---

## Parameters

---

folderId	(Optional) ID of the folder in which the dmc_notepage object to delete resides.
lockOwner	(Optional) Lock owner of the dmc_notepage object to delete.

---

objectId	(Required) Object ID of the dmc_notepage object to delete.
ownerName	(Optional) Owner name of the dmc_notepage object to delete.

---

## delete (privilege sysadmin, type dm\_user)

**Purpose** Deletes a user from the repository.

### Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

### Scope

---

privilege	sysadmin
type	dm_user

---

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the selected user.
objectname	(Required) Name of the user.

---

## delete (type dm\_acl)

**Purpose** Deletes an ACL.

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

---

type	dm_acl
------	--------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) The ID of the object to be deleted.
objectname	(Required) Name of the ACL.
type	(Required) Object type of the ACL.

---

## delete (type dm\_activity)

**Purpose** Deletes one or more activity objects. The user must have a role of contributor.

### Configuration File

webcomponent\config\actions\dm\_activity\_actions.xml



**Scope**

type	dm_activity
------	-------------

**Caller**

menubar and any component page that contains actionlinklist tag

**Parameters**

folderId	(Optional) The ID of the folder in which the object is located
lockOwner	(Optional) User who has checked out the object
objectId	(Required) The ID of the object to be deleted.
ownerName	(Optional) The name of the user who owns the object to be deleted

**delete (type dm\_folder)**

**Purpose** Deletes one or more folder objects. The user must have a role of contributor.

**Configuration File**

webcomponent\config\actions\dm\_folder\_actions.xml

**Scope**

type	dm_folder
------	-----------

## Caller

menubar and component pages that contain actionlinklist tag

## Parameters

---

folderId	(Optional) The ID of the folder in which the object is located
objectId	(Required) The ID of the object to be deleted.
ownerName	(Optional) The name of the user who owns the object to be deleted

---

# delete (type dm\_group)

**Purpose** Deletes a group from the repository.

## Configuration File

webcomponent\config\admin\group\dm\_group\_actions.xml

## Scope

---

type	dm_group
------	----------

---

## Caller

menubar component

**Parameters**

---

objectId	(Required) The ID of the object to be deleted.
objectname	(Required) Name of the group.

---

**delete (type dm\_notification)****Purpose** Deletes a notification.**Configuration File**

webcomponent\config\actions\dm\_notification\_actions.xml

**Scope**

---

type	dm_notification
------	-----------------

---

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**

---

objectId	(Required) The ID of the object to be deleted.
objectName	(Optional) The name of the notification object

---

## delete (type dm\_process)

**Purpose** Deletes one or more workflow templates. The user must have a role of contributor.

### Configuration File

webcomponent\config\actions\dm\_process\_actions.xml

### Scope

---

type	dm_process
------	------------

---

### Caller

menubar and component pages that contain actionlinklist tag

### Parameters

---

folderId	(Optional) The ID of the folder in which the workflow template is located
lockOwner	(Optional) User who has checked out the workflow template
objectId	(Required) The ID of the workflow template to be deleted.
ownerName	(Optional) User who owns the workflow template to be deleted

---

## delete (type dm\_queued)

**Purpose** Deletes a queue item in the inbox.

**Configuration File**

webcomponent\config\actions\dm\_queued\_actions.xml

**Scope**

type	dm_queued
------	-----------

**Caller**

menubar component

**Parameters**

objectId	(Required) The ID of the object to be deleted.
objectName	(Optional) The name of the queue item

**delete (type dm\_sysobject)****Purpose**

Deletes one or more objects. The user must have a role of contributor. In addition, the user must have the DELETE\_PERMIT access (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

## Caller

menubar component

## Parameters

---

folderId	(Optional) The ID of the folder in which the object is located
lockOwner	(Optional) User who has checked out the object
objectId	(Required) The ID of the object to be deleted.
ownerName	(Optional) The name of the user who owns the object to be deleted

---

# delete (type dm\_workflow)

**Purpose** Deletes a workflow object.

## Configuration File

webcomponent\config\actions\dm\_workflow\_actions.xml

## Scope

---

type	dm_workflow
------	-------------

---

## Caller

menubar component

**Parameters**

objectId	(Required) The ID of the object to be deleted.
workflowRuntimeState	(Required) State of the workflow. 0=dormant   1=running   2=finished   3=halted   4=terminated

**delete (type dmc\_comment)**

**Purpose** Deletes one or more comment objects. The user must have a role of contributor.

**Configuration File**

webcomponent\config\actions\dmc\_comment\_actions.xml

**Scope**

type	dmc_comment
------	-------------

**Caller**

embeddedtopic component UI

**Parameters**

objectId	(Required) ID of the selected object.
ownerName	(Optional) Name of the user who owns the object to be deleted

## delete (type dmc\_workqueue\_category)

**Purpose** Deletes one or more objects.

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_category\_actions.xml

### Scope

---

type	dmc_workqueue_category
------	------------------------

---

### Caller

menubar component and any component containing actionlinklist

### Parameters

---

objectId	(Required) ID of the object to be deleted.
objectname	(Optional) Name of the selected object
ownerName	(Optional) Name of the user who owns the object to be deleted

---

## delete (type dmc\_workqueue\_doc\_profile)

**Purpose** Deletes a work queue document profile object. This action is available only when the document profile is not used.

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_doc\_profile\_actions.xml



**Scope**

type	dmc_workqueue_doc_profile
------	---------------------------

**Caller**

menubar component and any component containing actionlinklist

**Parameters**

objectId	(Required) ID of the work queue document profile object.
objectname	(Optional) Name of the work queue document profile
ownerName	(Optional) Name of the user who owns the work queue document profile object

**delete (type dmc\_workqueue\_policy)**

**Purpose** Deletes a work queue policy. This action is available only when the policy is not being used.

**Configuration File**

webcomponent\config\actions\dmc\_workqueue\_policy\_actions.xml

**Scope**

type	dmc_workqueue_policy
------	----------------------

**Caller**

menubar component and any component containing actionlinklist

**Parameters**

---

objectId	(Required) ID of the work queue policy object.
objectname	(Optional) Name of the work queue policy object
ownerName	(Optional) Name of the user who owns the work queue policy object

---

## delete (type dmc\_workqueue)

**Purpose** Deletes one or more workqueue objects. The user must have a role of contributor.

**Configuration File**

webcomponent\config\actions\dmc\_workqueue\_actions.xml

**Scope**

---

type	dmc_workqueue
------	---------------

---

**Caller**

menubar and all component pages containing actionlinklist tag

**Parameters**

---

objectId	(Required) ID of the selected object.
objectname	(Optional) Name of the selected object to be deleted
ownerName	(Optional) Name of the user who owns the object to be deleted

---

---

## delete (type dmr\_content)

**Purpose** Deletes the selected rendition. The user must have write permission on the base object. If the selected object is a replica, it cannot be deleted.

### Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

### Scope

---

type	dmr_content
------	-------------

---

### Caller

menubar and any component page that contains actionlinklist tag

### Parameters

---

contentType	(Required) Type of the selected rendition.
objectId	(Required) The ID of the object to be deleted.

---

## delete (type job\_request)

**Purpose** Deletes a job request from the repository.

### Configuration File

webcomponent\config\admin\renamelog\renamelog\_actions.xml

### Scope

---

type	job_request
------	-------------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) The ID of the object to be deleted.
----------	--

---

## deletenotification (type dm\_notification)

**Purpose** Deletes a notification from a router.

### Configuration File

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

### Scope

---

type	dm_notification
------	-----------------

---

### Caller

taskmanagercontainer component

**Parameters**

objectId	(Required) The ID of the notification.
routerId	(Required) ID of the router.

**deletenotification (type dm\_queued)**

**Purpose** Deletes a notification from a router.

**Configuration File**

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

**Scope**

type	dm_queued
------	-----------

**Caller**

taskmanagercontainer component

**Parameters**

objectId	(Required) The ID of the notification.
routerId	(Required) ID of the router.

**deletesmartlist (type dm\_smart\_list)**

**Purpose** Deletes a smart list, even if it searches multiple repositories.

### Configuration File

webcomponent\config\actions\dm\_smart\_list\_actions.xml

### Scope

---

type	dm_smart_list
------	---------------

---

### Caller

allsavedsearches and mysavedsearches components

### Parameters

---

folderId	(Optional) ID of the folder that contains the smart list
lockOwner	(Optional) User who has checked out the smart list
objectId	(Required) ID of the selected smart list.
ownerName	(Optional) Owner of the smart list

---

## demote (type dm\_sysobject)

**Purpose** Demotes an object to the previous lifecycle state.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar component

**Parameters**

lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the object to be demoted.

**detachlifecycle (type dm\_sysobject)****Purpose** Detaches a lifecycle from the selected object.**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

**Caller**

menubar component

**Parameters**

lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the object to be demoted.

## docprofileadd

**Purpose** Adds a work queue document profile object

### Configuration File

webcomponent\config\library\queuemanagement\docprofileadd\_actions.xml

### Scope

All.

### Caller

docprofileattributes component

### Parameters

---

objectId	(Optional) Work queue ID
----------	--------------------------

---

## docprofileremove

**Purpose** Removes a work queue document profile object. This action is available only if the document profile is not being used.

### Configuration File

webcomponent\config\library\queuemanagement\docprofileadd\_actions.xml



**Scope**

All.

**Caller**

docprofileattributes component

## downloadcontent (type dm\_sysobject)

**Purpose**

Downloads the selected object to the disk.

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

appintgopen component class

**Parameters**

---

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system)
isIgnoreDescendants	(Optional) Boolean value that indicates whether or not to download descendants in a virtual document.

isReference	(Optional) Boolean flag that indicates whether the object is a reference object
nodeId	(Optional) Node ID of the object to download, when checking out part of a virtual document.
objectId	(Required) ID of the object to be viewed.
type	(Optional) Object type
vdmRootObjectId	(Optional) Object ID of the root of a virtual document (when operating on a child node).

---

## drledit (type dm\_sysobject)

**Purpose** Extends the action [editfile \(type dm\\_sysobject\)](#), [page 292](#) and opens a DRL for editing. This action is not defined for folders.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Parent Definition

editfile:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

This action is called by the **Edit** button event handler in the [drl](#), [page 698](#) component.

## drlview (entitlement collaboration, type dmc\_notepage)

**Purpose** Navigates to the parent folder or object of the DRL and opens the dmc\_notepage object for viewing. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\notepage\dmc\_notepage\_actions.xml

### Scope

---

entitlement	collaboration
type	dmc_notepage

---

### Parameters

---

objectId	(Required) Object ID of the dmc_notepage object to be opened.
----------	---

---

## drlview (type dm\_process)

**Purpose** Navigates to the parent folder or object of the DRL and opens the workflow template for viewing.

### Configuration File

webcomponent\config\actions\dm\_process\_actions.xml

### Parent Definition

view:webcomponent/config/actions/dm\_process\_actions.xml

### Scope

---

type	dm_process
------	------------

---

### Caller

Not called

## drlview (type dm\_sysobject)

**Purpose**      Opens a DRL for viewing.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Parent Definition

view:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

This action is called by the **View** button event handler in the [drl, page 698](#) component.

## edit (type dm\_smart\_list)

**Purpose** Launches the editing application so that the saved search can be revised. The user must have read\_permit access to the object (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_smart\_list\_actions.xml

### Scope

type	dm_smart_list
------	---------------

### Caller

allsavedsearches and mysavedsearches components

### Parameters

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system).
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
lockOwner	(Optional) User who has checked out the object
navigation	(Optional) Type of navigation upon execution
objectId	(Required) ID of the object to be viewed.
ownerName	(Optional) Name of the object owner
type	(Optional) Object type

## editaction (type dm\_ace)

**Purpose** Edits an entry in an ACL.

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

---

type	dm_ace
------	--------

---

### Caller

aclacelist and permissions components

### Parameters

---

accessorName	(Required) Name of user or group to add to ACL.
basicpermit	(Optional) The name of the basic permit for the accessor. Valid values are 1 (none), 2 (browse), 3 (read), 4 (relate), 5 (version), 6 (write), 7 (delete). Default = 3 (read)
extpermit	(Optional) The extended permission for the accessor. Valid values are EXECUTE_PROCEDURE_BIT, CHANGE_LOCATION_BIT, CHANGE_STATE_BIT, CHANGE_PERMISSIONS_BIT, TAKE_OWNERSHIP_BIT. Default = 0 (no extended permissions)
extpermit_readonly	(Optional) Set to true to disable extended permission controls
objectId	(Optional) ID of the ACL

---

objectName	(Optional) Name of ACL
permit	(Optional) String representation of the permission that is being applied to the accessor
permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT=0DF_ EXTENDED_PERMIT = 1DF_ APPLICATION_PERMIT = 2DF_ACCESS_RESTRICTION = 3DF_EXTENDED_RESTRICTION = 4DF_APPLICATION_RESTRICTION = 5DF_REQUIRED_GROUP = 6DF_REQUIRED_GROUP_SET = 7

---

## editacepermitaction (type dm\_ace)

**Purpose** Sets extended permissions in an ACL including Trusted Content Services support if it is configured for the repository

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

---

type	dm_ace
------	--------

---

### Caller

Not called

---

## Parameters

---

accessorName	(Required) Name of accessor (user or group)
basicpermit	(Optional) The name of the basic permit for the accessor. Valid values are 1 (none), 2 (browse), 3 (read), 4 (relate), 5 (version), 6 (write), 7 (delete). Default = 3 (read)
extpermit	(Optional) The extended permission for the accessor. Valid values are EXECUTE_PROCEDURE_BIT, CHANGE_LOCATION_BIT, CHANGE_STATE_BIT, CHANGE_PERMISSIONS_BIT, TAKE_OWNERSHIP_BIT. Default = 0 (no extended permissions)
extpermit_readonly	(Optional) Set to true to disable extended permission controls
objectId	(Optional) ID of the ACL
objectName	(Optional) Name of ACL
permit	(Optional) String representation of the permission that is being applied to the accessor
permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT=0DF_ EXTENDED_PERMIT = 1DF_ APPLICATION_PERMIT = 2DF_ACCESS_RESTRICTION = 3DF_EXTENDED_RESTRICTION = 4DF_APPLICATION_RESTRICTION = 5DF_REQUIRED_GROUP = 6DF_REQUIRED_GROUP_SET = 7

---



## editafternew (type dm\_sysobject)

**Purpose** Launches the editing application, as recognized by the operating system for the document extension, when a new object is created. The user must have a contributor role.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Parent Definition

editfile:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

newdoccontainer component

### Elements

<precondition>. <role>	The role whose members can perform the action. Default = contributor
<execution>. <navigation>	Type of navigation upon execution. Valid values: jump   returnjump   nested (default).

## editfile (entitlement collaboration, type dmc\_notepage)

**Purpose** Launches the editing application for dmc\_notepage objects as recognized by the operating system for the document extension. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\notepage\dmc\_notepage\_actions.xml

### Scope

---

entitlement	collaboration
type	dmc_notepage

---

### Parameters

---

objectId	(Required) Object ID of the dmc_notepage object to be opened.
ownerName	(Optional) User name of owner of the dmc_notepage object to be opened.

---

## editfile (type dm\_process)

**Purpose** Launches the Web Workflow Manager to allow editing of a workflow template. The user must have version level permission (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT). The user must also have a contributor role.

---

**Configuration File**

webcomponent\config\actions\dm\_process\_actions.xml

**Scope**

---

type	dm_process
------	------------

---

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**

---

objectId	(Required) ID of the workflow template. (This is the only parameter defined for templates.)
ownerName	(Optional) User who owns the workflow template
setError	(Optional) Set to true to pass the precondition error message to the WebcomponentErrorService as a non-fatal message

---

**editfile (type dm\_smart\_list)**

**Purpose** Edits a smart list (saved query)

**Configuration File**

webcomponent\config\actions\dm\_smart\_list\_actions.xml

**Scope**

---

type	dm_smart_list
------	---------------

---

**Caller**

5.3 search component

**Parameters**

---

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system).
isReference	(Optional) Boolean flag that indicates whether the object is a reference object (pointer to smart list in another repository)
lockOwner	(Optional) User who has checked out the smart list
navigation	(Optional) Type of navigation upon execution
objectId	(Required) ID of the smart list
ownerName	(Optional) Name of the smart list owner
type	(Optional) Object type of smartlist (dm_smartlist or subtype)

---

**editfile (type dm\_sysobject)****Purpose**

Launches the editing application, as recognized by the operating system for the document extension. The user must have version level permission (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT). The user must also have a contributor role.

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

appintgopen, doclist, drilldown, objectgrid, vdmlist, vdmliststreamline, menubar components

## Parameters

---

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system)
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
lockOwner	(Optional) User who has checked out the object (dm_query only)
nodeId	(Optional) Node ID of the object to open, when opening part of a virtual document.
objectId	(Required) ID of the object to be viewed. (This is the only parameter defined for templates.)
ownerName	(Optional) Name of the user who owns the object
type	(Optional) Object type
vdmRootObjectId	(Optional) Object ID of the root of a virtual document.

---

## Elements

The configurable elements are the same as those for [editafternew](#) (type `dm_sysobject`), [page 289](#). The following additional element is defined:

---

<execution>.	Set enabled to true if your action class can process OLE
<olecompound>	compound documents

---

# editprocessorauthentication

**Purpose** Sets the skill level and authorized document profiles for an existing work queue processor.

## Configuration File

webcomponent\config\library\queuemanagement\userprofile\_actions.xml

## Scope

All.

## Caller

processorattributes component

## Parameters

---

docprofile_enable	(Optional) Set to true to enable setting a profile for the user
docprofile_name	(Optional) Name to assign to the profile
skill	(Optional) Adds a value to the dropdown list of skills in the form of an NLS ID

---

username	(Required) Sets a label for the User Name, in the form of an NLS ID
wq_name	(Required) Work queue name

---

## editrestriction (type dm\_restriction)

**Purpose** Edits the restricted access of selected users or groups. Trusted Content Services must be enabled in the repository.

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

---

type	dm_restriction
------	----------------

---

### Caller

aclacelist component

### Parameters

---

accessorName	(Required) Name of user or group to add to ACL.
basicpermit	(Optional) The name of the basic permit for the accessor. Valid values are 1 (none), 2 (browse), 3 (read), 4 (relate), 5 (version), 6 (write), 7 (delete). Default = 3 (read)

extpermit	(Optional) The extended permission for the accessor. Valid values are EXECUTE_PROCEDURE_BIT, CHANGE_LOCATION_BIT, CHANGE_STATE_BIT, CHANGE_PERMISSIONS_BIT, TAKE_OWNERSHIP_BIT. Default = 0 (no extended permissions)
objectName	(Optional) Name of ACL
permit	(Optional) String representation of the permission that is being applied to the accessor
permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT=0DF_ EXTENDED_PERMIT = 1DF_ APPLICATION_PERMIT = 2DF_ACCESS_RESTRICTION = 3DF_EXTENDED_RESTRICTION = 4DF_APPLICATION_RESTRICTION = 5DF_REQUIRED_GROUP = 6DF_REQUIRED_GROUP_SET = 7

---

## editstartwfnote (type dmi\_package)

**Purpose** Allows the user to add or edit a task note at the start of a workflow

### Configuration File

webcomponent\config\actions\dmi\_package\_actions.xml

### Scope

---

type	dmi_package
------	-------------

---



**Caller**

startwfcomments component UI

**Parameters**

---

operationType	(Required) Type of operations to perform on a note. Valid values: add   edit   remove.
packageName	(Required) Name of the workflow package.
processId	(Required) ID of the workflow template

---

## edittasknote (type dmi\_package)

**Purpose** Allows the user to add, remove, or edit a task note

**Configuration File**

webcomponent\config\actions\dmi\_package\_actions.xml

**Scope**

---

type	dmi_package
------	-------------

---

**Caller**

taskcomments component UI

## Parameters

---

operationType	(Required) Type of edit operation: add   edit   remove
packageName	(Required) Name of the workflow package.
queueItemId	(Required) ID of the workflow task, which is passed as the objectId value to the edittasknote component.
queueItemState	(Required) State of queue item. Valid values: dormant   acquired   paused   ready   null.

---

## export (type dm\_sysobject)

**Purpose** Exports the object to the local file system.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

## Parameters

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format of the object (passed to the operating system)
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
nodeId	(Optional) Node ID of the object to export, when exporting part of a virtual document.
objectId	(Required) ID of the object to be exported.
vdmRootObjectId	(Optional) Object ID of the root of a virtual document.

## Elements

See [Dynamic component launching, page 9](#). The following additional element is defined:

<execution>.	Set enabled to true if your action class can process OLE
<olecompound>	compound documents

# exportrendition (type dmr\_content)

**Purpose** Exports a rendition to the local file system and launches the viewing application for the selected rendition

## Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

## Scope

type	dmr_content
------	-------------

## Caller

menubar and component pages containing actionlinklist tag

## Parameters

---

contentType	(Required) File format to use when opening the object (passed to the operating system)
objectId	(Required) ID of the selected object.
pageModifier	(Optional) A valid value for the page_modifier attribute of the dmr_content object. Do not pass in your own value for PDF renditions.

---

## Elements

See [Dynamic component launching, page 9](#).

# findtarget (type dm\_sysobject)

**Purpose** Allows the user to click on a reference object and navigate to the object or folder in the remote repository

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar component

**Parameters**


---

isReference	(Optional) Set to true if the selected object is a reference object
objectId	(Required) ID of the reference object.
openFolder	(Optional) Set to true if the reference object is a folder.

---

**finishworkflowtask (type dm\_task)****Purpose** Finishes a workflow task.**Configuration File**

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

**Scope**


---

type	dm_task
------	---------

---

**Caller**

taskmanagercontainer component

## Parameters

---

canCompleteTask	(Optional) Specifies that the task is active and can be completed.
hasEmptyMandatoryPkg	(Optional) The task contains an empty package that must be passed with the task
isFailedAutoTask	(Optional) Specifies that an automatic task has failed and can be re-run.
isManual	(Optional) Specifies whether transition is manual or automated.
objectId	(Required) ID of the task.
userCost	(Optional) Customer-defined cost data, stored as double in Server. Can be audited and summed up for completed workflows.
userTime	(Optional) Customer-defined time data, stored as integer in Server. Can be audited and summed up for completed workflows.

---

## folderbrowse (privilege sysadmin, type dm\_user)

**Purpose** Allows the administrator to browse a list of folders in a repository.

### Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

### Scope

---

privilege	sysadmin
type	dm_user

---

**Caller**

userattributes and userimport components

**forwardroutertask (type dm\_router\_task)****Purpose**

Forwards a router task.

**Configuration File**

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

**Scope**


---

type	dm_router_task
------	----------------

---

**Caller**

Not called

**Parameters**


---

canCompleteTask	(Optional) True to specify that the user can complete the task
objectId	(Required) ID of the router task.

---

**forwardworkflowtask (type dm\_task)****Purpose**

Forwards a task in a workflow.

## Configuration File

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

## Scope

---

type	dm_task
------	---------

---

## Caller

taskmanagercontainer component

## Parameters

---

canCompleteTask	(Optional) Specifies that the task is active and can be completed.
hasEmptyMandatoryPkg	(Optional) The task contains an empty package that must be passed with the task
hasForwardPaths	(Optional) Boolean flag that specifies whether there are any remaining paths in the workflow
isManual	(Optional) Specifies whether transition is manual or automated.
objectId	(Required) ID of the task.
userCost	(Optional) Customer-defined cost data, stored as double in Server. Can be audited and summed up for completed workflows.
userTime	(Optional) Customer-defined time data, stored as integer in Server. Can be audited and summed up for completed workflows.

---



## freezeassembly (type dm\_sysobject)

**Purpose** Freezes a virtual document snapshot. The user must have a contributor role.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

menubar component

### Parameters

assembledFromId	(Optional) ID of the parent virtual document, if any
isFrozenAssembly	(Optional) Boolean parameter that specifies whether the snapshot to be frozen is already frozen
lockOwner	(Optional) User who has checked out the snapshot
objectId	(Required) ID of the virtual document.
ownerName	(Optional) User who owns the snapshot

## get\_work

**Purpose** Retrieves the highest priority unassigned task from a work queue

## Configuration File

webcomponent\config\library\queuemanagement\queue\_actions.xml

## Scope

All.

## Caller

inboxdrilldown and inboxlist components, menubar

## Parameters

---

allowPullEmptyOnly	(Optional) Set to false to allow the user to get the next task even if the user's inbox contains other active tasks.
getWorkContext	(Required) Set to true to allow this action within the context of the calling component (set by inboxdrilldown and inboxlist). The action will be disabled within components that do not set the value to true.
ownerName	(Optional) Owner of the task
userName	(Optional) Current user

---

# getowner (type dm\_acl)

**Purpose** Gets the ACL owner name

## Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

**Scope**


---

type	dm_acl
------	--------

---

**Caller**

aclinfo component

**Parameters**


---

flatlist	(Optional) Boolean: true to list all objects in a flat list, false to display a drop down list.
multiselect	(Optional) Boolean: true to support multiple object selection
objectId	(Required) ID of ACL object.
objecttype	(Optional) Type of the selected object

---

**govern (entitlement collaboration)****Purpose**

Causes an object to become governed by a room. Requires Collaborative Edition. The object's access is restricted to a room's membership, and object can be governed by only one room.

**Configuration File**

webcomponent\config\library\rooms\room\_actions.xml

**Scope**


---

entitlement	collaboration
-------------	---------------

---

**Parameters**

---

folderId	(Required) ID of the governed folder that will contain the selected object..
objectId	(Required) ID of the selected object.
roomId	(Required) ID of the room that will govern the object.

---

## governfolder (entitlement collaboration)

**Purpose** Launches the govern component to govern a folder. The UI asks the user to specify whether child objects should also become governed. Requires Collaborative Edition.

**Configuration File**

webcomponent\config\library\rooms\room\_actions.xml

**Scope**

---

entitlement	collaboration
-------------	---------------

---

**Parameters**

---

folderId	(Required) ID of the governed folder that will contain the new governed folder.
objectId	(Required) ID of the selected object.
ownerName	(Optional) Name of the folder owner
roomId	(Required) ID of the room that will govern the folder
sourceFolderId	(Optional) ID of the folder that will be added to the governed folder or room

---

---

## groupbrowse (privilege sysadmin, type dm\_user)

**Purpose** Allows the administrator to browse a list of groups in a repository.

### Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

### Scope

privilege	sysadmin
type	dm_user

### Caller

grouprename, userattributes, userimport components

## groupbrowse (type dm\_group)

**Purpose** Allows the administrator to browse a list of groups in a repository.

### Configuration File

webcomponent\config\admin\group\dm\_group\_actions.xml

### Scope

type	dm_group
------	----------

### Caller

grouprename, userattributes, userimport components

## groupreassign (type dm\_group)

**Purpose** Renames a group

### Configuration File

webcomponent\config\admin\group\dm\_group\_actions.xml

### Scope

---

type	dm_group
------	----------

---

### Caller

menubar

### Parameters

---

objectId	(Required) ID of the selected group.
objectname	(Required) Name of the group.

---

## groupreassign (type dm\_member\_group)

**Purpose** Renames a group.

**Configuration File**

webcomponent\config\admin\group\dm\_member\_actions.xml

**Scope**

type	dm_member_group
------	-----------------

**Caller**

menubar component

**Parameters**

objectId	(Required) ID of the selected group.
objectname	(Required) Name of the group.

**grouprenamelog (type dm\_group)****Purpose** Displays requests for renaming groups.**Configuration File**

webcomponent\config\admin\group\dm\_group\_actions.xml

**Scope**

type	dm_group
------	----------

**Caller**

menubar component

## Parameters

---

renametarget	(Optional) Type of objects requested for rename. Valid values: USER   GROUP
--------------	---

---

# halttask (type dm\_task)

**Purpose** Halts (pauses) a task in a workflow.

## Configuration File

webcomponent\config\actions\dm\_task\_actions.xml

## Scope

---

type	dm_task
------	---------

---

## Caller

reportdetailsummarylist component

## Parameters

---

objectName	(Required) Task name.
taskRuntimeFlag	(Required) State of the workflow task. Valid values: 0=dormant   1=acquired   2=finished   3 – 6 =paused   7=ready
workitemId	(Required) ID of the work item.

---



## haltworkflow (type dm\_workflow)

**Purpose** Halts a workflow

### Configuration File

webcomponent\config\actions\dm\_workflow\_actions.xml

### Scope

type	dm_workflow
------	-------------

### Caller

menubar, reportdetailsummarydrilldown, reportdetailsummarylist, reportmainlist components

### Parameters

objectId	(Required) ID of the workflow.
supervisor	(Optional) User name of workflow supervisor
workflowRuntimeState	(Required) State of the workflow. 0=dormant   1=running   2=finished   3=halted   4=terminated

## historicalprocessreport

**Purpose** Creates a report showing historical workflow data.

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

### Caller

menubar component

### Parameters

---

ownerName	(Optional) Not used
-----------	---------------------

---

## historicaluserreport

**Purpose**      Creates a report showing historical workflow data organized by user

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

### Caller

menubar component

**Parameters**

---

ownerName	(Optional) Not used
-----------	---------------------

---

## history (type dm\_sysobject)

**Purpose**        Displays the audit trail for an object.

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar component

**Parameters**

---

objectId	(Required) ID of the selected object.
----------	---------------------------------------

---

**Elements**

---

<failoverenabled>	This component supports failover. To disable failover, enter a value of false.
-------------------	--

---

## history (type dm\_task)

**Purpose** Displays the history of a workflow task.

### Configuration File

webcomponent\config\actions\dm\_task\_actions.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the selected object.
taskmanagerId	(Required) Component ID of the task manager component.

---

## import (type dm\_folder)

**Purpose** Launches import of an object from the local file system to the repository. The user must have a contributor role. The permit value folder\_link\_permit performs a check for write permissions on the folder when the folder security is enabled.

### Configuration File

webcomponent\config\actions\dm\_folder\_actions.xml

**Scope**


---

type	dm_folder
------	-----------

---

**Caller**

menubar

**Parameters**


---

baseDocbaseType	(Optional) Base object type for import
docbaseType	(Optional) Object type of imported object
filePath	(Optional) Path to the file on the local machine
format	(Optional) Format of the object to be imported
isDirectory	(Optional) Boolean that indicates whether the object being imported is a directory
objectId	(Required) ID of the destination folder.
objectName	(Optional) Name of object to be imported
ownerName	(Optional) Name of the owner of the destination folder.
parentPath	(Optional) Repository path into which the file is imported

---

**importexternalresult (type dm\_externalresult)****Purpose** Imports a file listed after searching an external source**Configuration File**

webcomponent\config\actions\dm\_externalresult\_actions.xml

### Scope

---

type	dm_externalresult
------	-------------------

---

### Caller

Search results page (5.3 search component) and viewexternalresult component page

### Parameters

---

entryId	(Required) ID of the result entry.
queryId	(Required) Internal ID of the query used for searching the external source.

---

## importrendition (type dm\_sysobject)

### Purpose

Imports an HTML or PDF rendition of the selected object. The user must have version level permission on the object. The user must have read\_permit access to the object (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar and renditions components

## Parameters

contentType	(Optional) Type of rendition to import
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the selected object.
ownerName	(Optional) Name of the user who owns the object

## Elements

See [Dynamic component launching, page 9](#).

# importrendition (type dmr\_content)

**Purpose** Imports a file to create a new rendition

## Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

## Scope

type	dmr_content
------	-------------

## Caller

menubar and component pages that contain actionlinklist tag

**Parameters**

---

contentType	(Optional) File format to use when importing the object (passed to the operating system).
isReference	(Optional) Boolean flag that specifies whether the object is a reference
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the selected object.

---

**importrenditiongeneric (type dm\_sysobject)**

**Purpose** Imports an HTML or PDF rendition of the selected object. The user must have version level permission on the object.

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Parent Definition**

importrendition:application='webcomponent' type='dm\_sysobject'

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

Not called



## importrenditiongeneric (type dmr\_content)

**Purpose** Imports a file to create a new rendition

### Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

### Parent Definition

importrendition:application='webcomponent' type='dm\_sysobject'

### Scope

type	dmr_content
------	-------------

### Caller

None

## launchauthenticate

**Purpose** Launches the authenticate component

### Configuration File

wdk\config\session\_actions.xml

### Scope

All.

**Caller**

changesearchsources, savedsearches, savesearch, search components.

**Parameters**

---

docbase	(Required) Repository for which to display the login dialog
docbroker	(Optional) Name of the connection broker to which the specified repository is projected
docbrokerPort	(Optional) Port that needs to be used for the specified connection broker
suppressLoginEvent	(Optional) Specifies whether to fire the "loginSuccess" or "loginCancel" client events after the login dialog has been dismissed

---

## launchauthenticateexternalsource

**Purpose**      Displayed when a user is required to provide credential to an external source

**Configuration File**

wdk\config\session\_actions.xml

**Scope**

All.

**Caller**

Called by the SearchEx class of the 5.3 search component

## Parameters

---

externalSource	(Required) Source that is set up in the ECI server. It is an internal name for the source, and is used to obtain a suitable source name. For example, for an internal source name 'general/GoogleBackends/Google-French,' it returns 'Google-French.'
----------------	---

---

## link (type dm\_sysobject)

**Purpose** Links a clipboard object to a new location

### Configuration File

webcomponent\config\environment\clipboard\dm\_sysobject\_clipboard\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Usage

The abstract clipboard class ignores auxiliary files for the linked or copied object if they are not in the clipboard. The link or copy operation proceeds without the auxiliary files. Whether a linked file is auxiliary can be determined by calling `isAuxiliaryObject(objectId)`.

## locations (type dm\_acl)

**Purpose** Displays the locations where an ACL is used.

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

---

type	dm_acl
------	--------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the selected object.
----------	---------------------------------------

---

## locations (type dm\_cabinet)

**Purpose** Specifies the locations in the repository of a given object (links). This action is disabled for dm\_cabinet objects.

### Configuration File

webcomponent\config\actions\dm\_cabinet\_actions.xml

**Scope**

---

type	dm_cabinet
------	------------

---

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**

---

objectId	(Required) ID of the selected object.
----------	---------------------------------------

---

## locations (type dm\_group)

**Purpose**

Allows an administrator to get locations where the group owns objects, ACLS, or workflows.

**Configuration File**

webcomponent\config\admin\group\dm\_group\_actions.xml

**Scope**

---

type	dm_group
------	----------

---

**Caller**

menubar component

## Parameters

---

objectId	(Required) ID of the selected object.
objectname	(Required) Name of the group.

---

## locations (type dm\_member\_group)

**Purpose** Specifies the locations in the repository of a given object (links).

### Configuration File

webcomponent\config\admin\group\dm\_member\_actions.xml

### Scope

---

type	dm_member_group
------	-----------------

---

### Caller

menubar component

## Parameters

---

objectId	(Required) ID of the selected object.
objectname	(Required) The name of the selected object.

---

## locations (type dm\_member\_user)

**Purpose** Specifies the locations in the repository of a given object (links).

**Configuration File**

webcomponent\config\admin\group\dm\_member\_actions.xml

**Scope**

---

type	dm_member_user
------	----------------

---

**Caller**

menubar component

**Parameters**

---

objectname	(Required) The name of the selected object.
------------	---

---

## locations (type dm\_sysobject)

**Purpose** Specifies the locations in the repository of a given object (links).

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar component

## Parameters

---

folderId	(Optional) ID of the folder that contains the selected object.
folderPath	(Optional) Path to the folder that contains the selected object. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder
objectId	(Required) ID of the selected object.

---

## Elements

---

<failoverenabled>	This component supports failover. To disable failover, enter a value of false.
<navigation>	Type of navigation upon execution. Valid values: jump   returnjump   nested (default).

---

# locations (type dm\_user)

**Purpose** Allows the administrator to view all of the groups, ACLs, alias sets, objects, and workflows that the specified user owns or belongs to.

## Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

## Scope

---

type	dm_user
------	---------

---



**Caller**

menubar component

**Parameters**


---

objectname	(Required) The name of the selected user
------------	--

---

# logout

**Purpose**

Logs the user out of the application

**Configuration File**

wdk\config\session\_actions.xml

**Scope**

All.

**Caller**

Not called directly. The webcomponent layer logout action extends this action and is called by the menubar component

**Parameters**


---

homeurl	(Optional) URL to a JSP page or component that will be displayed when the user logs out of the application or when a DRL action has been completed by a default account (anonymous) user
---------	--

---

## Elements

---

<code>&lt;execution&gt;.&lt;homeurl&gt;</code>	URL to a JSP page or component that will be displayed when the user logs out of the application
--	---

---

# logout

**Purpose**      Logs the user out of the application

## Configuration File

`webcomponent\config\session_actions.xml`

## Parent Definition

`/wdk/config/session_actions.xml`

## Scope

All.

## Caller

Not called directly. The webcomponent layer logout action extends this action and is called by the menubar component

## Elements

---

<code>&lt;execution&gt;.&lt;homeurl&gt;</code>	URL to a JSP page or component that will be displayed when the user logs out of the application
--	---

---

## makesimple (type dm\_sysobject)

**Purpose** Converts a virtual document to a simple document if the document has no child nodes.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

isReference	(Optional) Boolean flag that indicates whether the object is a reference object
isVirtualDoc	(Optional) Boolean flag that specifies whether the object is a virtual document (required for pre-4.1 repository virtual documents)
linkCount	(Optional) Number of links (child nodes) in a document. If greater than 0, the document is a virtual document.
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the object to be converted into a virtual document.

---

## makevirtual (type dm\_sysobject)

**Purpose** Converts a simple object to a virtual document.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

addnewvirtualdocumentnode and menubar components

### Parameters

---

isReference	(Optional) Boolean flag that indicates whether the object is a reference object
isVirtualDoc	(Optional) Boolean flag that specifies whether the object is a virtual document (required for pre-4.1 repository virtual documents)
linkCount	(Optional) Number of links (child nodes) in a document. If greater than 0, the document is a virtual document.
lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the object to be converted into a virtual document.

---

## managequeueinbox (type dmc\_workqueue)

**Purpose** Manages the queue inbox containing the tasks of the selected queue.

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_actions.xml

### Scope

type	dmc_workqueue
------	---------------

### Caller

queuemonitorlist component UI

### Parameters

categoryPath	(Optional) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs.
queuename	(Required) Name of the selected queue.

## markread (entitlement collaboration, type dm\_sysobject)

**Purpose** Marks a topic as having been read. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\discussion\topic\_actions.xml

## Scope

---

entitlement	collaboration
type	dm_sysobject

---

## Caller

menubar component

## Parameters

---

objectId	(Required) ID of the topic to mark as read.
topicStatus	(Optional) Valid values of TopicStatusAttributeDataHandler: 0 (STATUS_DISABLED)   1 (STATUS_NONE)   2 (STATUS_NO_COMMENTS)   3 (STATUS_READ)   4 (STATUS_UNREAD)

---

# markunread (entitlement collaboration, type dm\_sysobject)

**Purpose** Marks a selected topic as unread. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\discussion\topic\_actions.xml

## Scope

---

entitlement	collaboration
type	dm_sysobject

---

**Caller**

menubar component

**Parameters**

---

objectId	(Required) ID of the topic to mark as unread.
topicStatus	(Optional) Valid values of TopicStatusAttributeDataHandler: 0 (STATUS_DISABLED)   1 (STATUS_NONE)   2 (STATUS_NO_COMMENTS)   3 (STATUS_READ)   4 (STATUS_UNREAD)

---

**modifyversionlabels (type dm\_sysobject)**

**Purpose**        Modifies the version labels on an object or one or more nodes in a virtual document.

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar component

## Parameters

---

nodeId	(Optional) Node ID of the virtual document node on which to change the version label.
objectId	(Required) Object ID of the object on which to change version labels.

---

# move\_queued\_task (type dm\_task)

**Purpose** Moves a task from one work queue to another.

## Configuration File

webcomponent\config\actions\dm\_queued\_task\_actions.xml

## Scope

---

type	dm_task
------	---------

---

## Caller

menubar component

## Parameters

---

objectId	(Required) ID of task
ownerName	(Optional) Owner of task
performer	(Required) Task performer
queueName	(Required) Name of work queue

---



## move (type dm\_sysobject)

**Purpose** Moves a clipboard object to a new location

### Configuration File

webcomponent\config\environment\clipboard\dm\_sysobject\_clipboard\_actions.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

menubar component

### Usage

The move operation can be globally configured as the value of <move\_operations>. <all\_versions> in app.xml to move the selected version (false) or to move all versions of the selected object (true). With the value of true, if the object has versions linked to other folders, only the versions linked to the folder containing the current version will be moved. When a virtual document is selected and move, only the root document is moved with all its versions, not the descendants.

## move (type dmc\_workqueue\_category)

**Purpose** Moves a work queue or work queue category to a new location

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_category\_actions.xml

### Scope

---

type	dmc_workqueue_category
------	------------------------

---

### Caller

menubar component

### Parameters

---

ownerName	(Optional) Name of the user who owns the work queue to be moved
-----------	---

---

## move (type dmc\_workqueue)

**Purpose** Moves a clipboard object to a new location.

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_actions.xml

### Scope

---

type	dmc_workqueue
------	---------------

---

### Caller

menubar component

# myworkqueuecategories

**Purpose** Displays a list of the work queue categories of which the current user is a manager.

## Configuration File

webcomponent\config\actions\queue\_management\_actions.xml

## Scope

All.

## Caller

queuemonitorlist component

## Parameters

arrangeselect	(Optional) Boolean: true to allow rearrangement of order
multiselect	(Optional) Boolean: true to support multiple object selection
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

# newacl (type dm\_acl)

**Purpose** Creates a new ACL

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

---

type	dm_acl
------	--------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the ACL.
----------	---------------------------

---

## newassembly (type dm\_sysobject)

**Purpose** Creates a new virtual document snapshot. The user must have a contributor role assigned in the repository.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

**Parameters**


---

folderId	(Optional) ID of the folder in which to create the assembly
linkCount	(Optional) Number of links (child nodes) in a document. If greater than 0, the document is a virtual document.
objectId	(Required) ID of the virtual document.
ownerName	(Optional) Owner of virtual document. Used by role precondition to determine whether user can create an assembly.

---

**newcabinet (type dm\_docbase)**

**Purpose**        Creates a new cabinet in a repository

**Configuration File**

webcomponent\config\actions\dm\_cabinet\_actions.xml

**Scope**


---

type	dm_docbase
------	------------

---

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**


---

ownerName	(Optional) Name of the user having the rights to create a new cabinet
-----------	---

---

## newdocument (type dm\_folder)

**Purpose** Creates a document within a folder. The user must have a contributor role. The permit value folder\_link\_permit performs a check for write permissions on the folder when folder security is enabled.

### Configuration File

webcomponent\config\actions\dm\_folder\_actions.xml

### Scope

---

type	dm_folder
------	-----------

---

### Caller

menubar and component pages that contain actionlinklist tag

### Parameters

---

contentType	(Optional) Document format
editAfterCreate	(Optional) Specifies whether the document can be edited after creation
objectId	(Required) ID of the folder or cabinet that will contain the new object.
ownerName	(Optional) Name of the owner of the folder or cabinet. Used by the NewObjectPrecondition class to determine whether the action can be performed.

---

## newfolder (type dm\_folder)

**Purpose** Creates a new folder. The user must have a contributor role. The permit value folder\_link\_permit performs a check for write permissions on the folder when the folder security is enabled.

### Configuration File

webcomponent\config\actions\dm\_folder\_actions.xml

### Scope

type	dm_folder
------	-----------

### Caller

menubar and component pages that contain actionlinklist tag

### Parameters

objectId	(Required) ID of the folder or cabinet that will contain the new object.
ownerName	(Optional) Name of the owner of the folder or cabinet. Used by the NewObjectPrecondition class to determine whether the action can be performed.

## newformat (type dm\_format)

**Purpose** Creates a new Documentum format

### Configuration File

webcomponent\config\admin\format\dm\_format\_actions.xml

### Scope

---

type	dm_format
------	-----------

---

### Caller

Not called

### Parameters

---

objectId	(Required) Set value to newobject to create new format.
----------	---

---

## newgroup (privilege creategroup, type dm\_group)

**Purpose**      Creates a new group in the repository.

### Configuration File

webcomponent\config\admin\group\dm\_group\_actions.xml

### Scope

---

privilege	creategroup
type	dm_group

---

### Caller

menubar component



**Parameters**

objectId	(Required) Must have a value of newobject to create a new group.
objectname	(Required) Name of the group.

**newnotepage (entitlement collaboration, type dm\_folder)**

**Purpose** Creates a new note object. A note is a content file that can be displayed directly in the browser. Requires Collaborative Edition.

**Configuration File**

webcomponent\config\actions\dm\_folder\_actions.xml

**Scope**

entitlement	collaboration
type	dm_folder

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**

objectId	(Required) ID of the folder or cabinet that will contain the new object.
ownerName	(Optional) Name of the owner of the folder or cabinet. Used by the NewObjectPrecondition class to

determine whether the action can be performed.

---

## newprocess (type dm\_folder)

**Purpose** Creates a new workflow template by launching the Web workflow manager. The user must have a contributor role. The permit value folder\_link\_permit performs a check for write permissions on the folder when folder security is enabled.

### Configuration File

webcomponent\config\actions\dm\_folder\_actions.xml

### Scope

---

type	dm_folder
------	-----------

---

### Caller

menubar and component pages that contain actionlinklist tag

### Parameters

---

objectId	(Required) ID of the folder in which to create the template.
ownerName	(Optional) Name of the owner of the folder or cabinet. Used by the NewObjectPrecondition class to determine whether the action can be performed.

---

## newrole (privilege creategroup, type dm\_role)

**Purpose** Creates a new role in the repository (dm\_group object with attributes group\_class="role" and group\_name=" *role\_name*").

### Configuration File

webcomponent\config\admin\role\dm\_role\_actions.xml

### Scope

privilege	creategroup
type	dm_role

### Caller

menubar component

### Parameters

objectId	(Required) ID of the selected role. Set to newobject to create a new role.
objectname	(Required) Name of the role.

## newroom (entitlement collaboration, privilege creategroup, type dm\_folder)

**Purpose** Creates a new room object (subtype of dm\_folder). Requires Collaborative Edition

### Configuration File

webcomponent\config\library\rooms\room\_actions.xml

### Scope

---

entitlement	collaboration
privilege	creategroup
type	dm_folder

---

### Caller

menubar and component pages that contain actionlinklist tag

### Parameters

---

objectId	(Required) ID of the cabinet in which to create a room.
ownerName	(Optional) Name of the object owner

---

## newuser (privilege sysadmin, type dm\_user)

**Purpose**      Creates a new user in the repository

### Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

### Scope

---

privilege	sysadmin
type	dm_user

---

**Caller**

menubar component

**Parameters**

---

objectId	(Required) ID of the selected user. Set to newobject to create a new user.
objectname	(Required) Name of the user.

---

## newvirtualdocumentnode (type dm\_sysobject)

**Purpose** Adds content to a virtual document**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar component

## Parameters

---

editAfterCreate	(Optional) Set to true to opening editing application after creating the new node
objectId	(Required) Object ID of the object to add to the virtual document.
ownerName	(Optional) Owner of virtual document. Used by role precondition to determine whether user can add a node.

---

## newwindow

**Purpose** Launches the newwindow component to open a second WDK application browser window

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

### Caller

menubar component

## newWorkQueue (type dmc\_workqueue)

**Purpose** Creates a new work queue object.

**Configuration File**

webcomponent\config\actions\dmc\_workqueue\_actions.xml

**Scope**


---

type	dmc_workqueue
------	---------------

---

**Caller**

Menu item in menubar component

**Parameters**


---

categoryPath	(Required) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs.
objectId	(Required) ID of the work queue object.
objectname	(Required) Name of the work queue object.

---

## newWorkQueueCategory

**Purpose**      Creates a new work queue category object

**Configuration File**

webcomponent\config\actions\dmc\_workqueue\_category\_actions.xml

**Scope**

All.

## Caller

Menubar component

## Parameters

---

categoryPath	(Required) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs.
objectId	(Required) ID of the work queue category object.
objectname	(Required) Name of the work queue category.

---

# newWorkQueueDocProfile (type dmc\_workqueue\_doc\_profile)

**Purpose**      Creates a new work queue document profile object

## Configuration File

webcomponent\config\actions\dmc\_workqueue\_doc\_profile\_actions.xml

## Scope

---

type	dmc_workqueue_doc_profile
------	---------------------------

---

## Caller

menubar component



**Parameters**

objectId	(Required) ID of the work queue document profile object.
objectname	(Required) Name of the work queue document profile.

**newWorkQueuePolicy (type dmc\_workqueue\_policy)**

**Purpose**      Creates a new work queue policy object

**Configuration File**

webcomponent\config\actions\dmc\_workqueue\_policy\_actions.xml

**Scope**

type	dmc_workqueue_policy
------	----------------------

**Caller**

menubar component

**Parameters**

objectId	(Required) ID of the work queue policy object.
objectname	(Required) Name of the work queue policy object.

## newxforms (type dm\_folder)

**Purpose** Creates a new form. The user must have the role form\_user and folder\_link\_permit permission. Customization is not supported.

### Configuration File

webcomponent\config\actions\dm\_folder\_actions.xml

### Scope

---

type	dm_folder
------	-----------

---

### Caller

menubar and component pages that contain actionlinklist tag

### Parameters

---

objectId	(Required) Object ID of the current folder in which the form will be created.
ownerName	(Optional) Name of the owner of the folder or cabinet. Used by the NewObjectPrecondition class to determine whether the action can be performed.

---

## openassembly (type dm\_sysobject)

**Purpose** Opens a virtual document assembly (snapshot) in the virtual document editor

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

mybojects, search, subscriptions, doclist, drilldown, objectgrid components

## Parameters

---

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system)
home_component	(Optional) In Portal application, specifies portal component that calls this component. Used to construct a breadcrumb back to the portal.
home_component_args	(Optional) In Portal application, specifies arguments that are passed to the portal component that calls this component when the user clicks the breadcrumb to go back.
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
launchViewer	(Optional) Boolean flag that indicates whether to launch a viewing application.
nodeId	(Optional) ID of the node in a virtual document
objectId	(Required) ID of the object to be viewed.
pathInfo	(Optional) Path to be displayed in the breadcrumb

selectedNodeIds	(Optional) Comma-separated list of nodeIds that should be initially selected when the vdmList component is first displayed
structureComponent	(Optional) Name of the component that displays virtual document structure

---

## openvirtualdocument (type dm\_sysobject)

**Purpose** Opens a virtual document in the virtual document structure viewing component.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

myobjects, search, subscriptions, doclist, drilldown, objectgrid components

### Parameters

---

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system)
home_component	(Optional) In Portal application, specifies portal component that calls this component. Used to construct a breadcrumb back to the portal.

---

home_component_args	(Optional) In Portal application, specifies arguments that are passed to the portal component that calls this component when the user clicks the breadcrumb to go back.
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
launchViewer	(Optional) Boolean flag that indicates whether to launch a viewing application.
nodeId	(Optional) ID of the node in a virtual document
objectId	(Required) ID of the object to be viewed.
pathInfo	(Optional) Path to be displayed in the breadcrumb
selectedNodeIds	(Optional) Comma-separated list of nodeIds that should be initially selected when the vdmList component is first displayed
structureComponent	(Optional) Name of the component that displays virtual document structure

---

## Elements

The configurable elements are the same as those in [doclist](#), page 684.

# options (type dmc\_room)

**Purpose** Launches the roomoptions component with room options such as setting a banner and default permissions

## Configuration File

webcomponent\config\library\rooms\room\_actions.xml

### Scope

---

type	dmc_room
------	----------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) Room ID
----------	--------------------

---

## ownerbrowse (type dm\_group)

**Purpose** Allows the administrator to browse a list of group owners in the repository.

### Configuration File

webcomponent\config\admin\group\dm\_group\_actions.xml

### Scope

---

type	dm_group
------	----------

---

### Caller

This action is called by the UI of the component [groupattributes](#), page 746.

## permissions (type dm\_sysobject)

**Purpose** Displays the permissions for a selected object.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

menubar component

### Parameters

objectId	(Required) ID of the selected object.
----------	---------------------------------------

### Elements

<failoverenabled>	This component supports failover. To disable failover, enter a value of false.
-------------------	--

## preferences

**Purpose** Allows a user to set preferences.

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

### Caller

Not called

## processorattributes (type dm\_group)

**Purpose** Displays the attributes page for a group assigned to a work queue.

### Configuration File

webcomponent\config\library\queuemangement\QueueUsersandGroups\_actions.xml

### Scope

---

type	dm_group
------	----------

---

### Caller

QueueUsersandGroups component



**Parameters**

objectId	(Required) ID of group for which to display attributes.
objectname	(Required) Name of group for which to display attributes.

**processorattributes (type dm\_user)**

**Purpose** Displays the attributes page for a work queue processor.

**Configuration File**

webcomponent\config\library\queuemanagement\queueusersandgroups\_actions.xml

**Scope**

type	dm_user
------	---------

**Caller**

queueusersandgroups component

**Parameters**

objectId	(Required) ID of user for which to display attributes.
objectname	(Required) Name of user for which to display attributes.

## processorproperties (type dm\_group)

**Purpose** Displays the properties component for a group assigned to a work queue.

### Configuration File

webcomponent\config\library\queuemanagement\queueusersandgroups\_actions.xml

### Scope

---

type	dm_group
------	----------

---

### Caller

queueusersandgroups component

### Parameters

---

objectId	(Required) ID of group for which to display properties.
objectname	(Required) Name of group for which to display properties.

---

## processorproperties (type dm\_user)

**Purpose** Displays the properties component for a work queue processor.

### Configuration File

webcomponent\config\library\queuemanagement\queueusersandgroups\_actions.xml

**Scope**

type	dm_user
------	---------

**Caller**

queueusersandgroups component

**Parameters**

objectId	(Required) ID of user for which to display properties.
objectname	(Required) Name of user for which to display properties.

## processorworkloadinbox

**Purpose** Displays the work queue items assigned to a particular work queue processor.

**Configuration File**

webcomponent\config\actions\queue\_member\_actions.xml

**Scope**

All.

**Caller**

menubar component

## Parameters

---

categoryPath	(Optional) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs.
objectname	(Required) Name of group to which user or group must belong to process inbox queue items.
ownerName	(Optional) Owner of group
queueName	(Required) Name of the workload queue

---

# promote (type dm\_sysobject)

**Purpose** Promotes an object to the next lifecycle state.

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

menubar component

## Parameters

---

lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the object to be promoted.

---

## properties (type dm\_acl)

**Purpose** Displays the properties of an ACL.

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

type	dm_acl
------	--------

### Caller

acllist, aclwhereused components

### Parameters

objectId	(Required) ID of the selected object.
----------	---------------------------------------

## properties (type dm\_externalresult)

**Purpose** Displays the properties component for an external result object

### Configuration File

webcomponent\config\actions\dm\_externalresult\_actions.xml

### Scope

---

type	dm_externalresult
------	-------------------

---

### Caller

Search results page (5.3 search component) and viewexternalresult component page

### Parameters

---

entryId	(Required) ID of the result entry.
queryId	(Required) ID of the query used for searching the external source.

---

## properties (type dm\_format)

**Purpose** Displays the properties of a selected format.

### Configuration File

webcomponent\config\admin\format\dm\_format\_actions.xml

### Scope

---

type	dm_format
------	-----------

---

### Caller

formatlist component

**Parameters**

---

objectId	(Required) ID of the selected object.
----------	---------------------------------------

---

## properties (type dm\_group)

**Purpose** Displays the properties for the repository group or creates a new group.

**Configuration File**

webcomponent\config\admin\group\dm\_group\_actions.xml

**Scope**

---

type	dm_group
------	----------

---

**Caller**

groupelist, groupwhereused components

**Parameters**

---

objectId	(Required) ID of the selected group. Set to newobject to create a new group.
objectname	(Required) Name of the group.

---

## properties (type dm\_member\_group)

**Purpose** Displays the properties of a dm\_member\_group object.

### Configuration File

webcomponent\config\admin\group\dm\_member\_actions.xml

### Scope

---

type	dm_member_group
------	-----------------

---

### Caller

roommembership component

### Parameters

---

objectId	(Required) ID of the group for which to display properties
objectname	(Required) Name of group

---

## properties (type dm\_member\_user)

**Purpose** Displays the properties of a dm\_member\_user object.

### Configuration File

webcomponent\config\admin\group\dm\_member\_actions.xml

### Scope

---

type	dm_member_user
------	----------------

---

### Caller

userlist, userwhereused component



**Parameters**

---

objectId	(Required) ID of the user.
objectname	(Required) Name of user

---

## properties (type dm\_sysobject)

**Purpose** Displays the properties of an object (dm\_sysobject).

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

myobjects\_drilldown, myobjects\_list, relationships, renditions, subscriptions, viewassemblies components

**Parameters**

---

enableShowAll	(Optional) Determines whether the Show All checkbox is displayed. (true false)
objectId	(Required) ID of the selected object.
preferForms	(Optional) Set to true to launch the forms properties component
templateId	(Optional) ID of associated Xforms template, if any

---

## Elements

---

<failoverenabled>	This component supports failover. To disable failover, enter a value of false.
-------------------	--

---

## properties (type dm\_user)

**Purpose** Displays properties for a selected user.

### Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

### Scope

---

type	dm_user
------	---------

---

### Caller

userlist, userwhereused component

### Parameters

---

objectId	(Required) ID of the selected user. Set to newobject to create a new user.
objectname	(Required) Name of the user.

---

## properties (type dmc\_workqueue\_category)

**Purpose** Displays the properties of an object (dmc\_workqueue\_category).

**Configuration File**

webcomponent\config\actions\dmc\_workqueue\_category\_actions.xml

**Scope**

type	dmc_workqueue_category
------	------------------------

**Caller**

queuemonitorlist component UI

**Parameters**

objectId	(Required) ID of the selected object.
objectname	(Required) Name of the selected object.

**properties (type dmc\_workqueue\_doc\_profile)****Purpose** Displays the properties of a work queue document profile object**Configuration File**

webcomponent\config\actions\dmc\_workqueue\_doc\_profile\_actions.xml

**Scope**

type	dmc_workqueue_doc_profile
------	---------------------------

**Caller**

docprofilelist component

## Parameters

---

objectId	(Required) ID of the work queue document profile object.
objectname	(Required) Name of the work queue document profile.

---

# properties (type dmc\_workqueue\_policy)

**Purpose** Displays the properties component for a work queue policy

## Configuration File

webcomponent\config\actions\dmc\_workqueue\_policy\_actions.xml

## Scope

---

type	dmc_workqueue_policy
------	----------------------

---

## Caller

policylist component

## Parameters

---

objectId	(Required) ID of the work queue policy object.
objectname	(Required) Name of the work queue policy object.

---

## properties (type dmc\_workqueue)

**Purpose** Displays the properties of a dmc\_workqueue object

### Configuration File

webcomponent\config\actions\dmc\_workqueue\_actions.xml

### Scope

type	dmc_workqueue
------	---------------

### Caller

queuelist component UI

### Parameters

objectId	(Required) ID of the selected object.
objectname	(Required) Name of the selected object

## queuelist

**Purpose** Displays a list of the available work queues and work queue categories.

### Configuration File

webcomponent\config\library\queuemanagement\list\queuelist\_actions.xml

**Scope**

All.

**Caller**

docprofileadd component

**Parameters**

---

category_path	(Optional) Work queue category to display
object_name	(Optional) Not used
objectId	(Optional) Queue ID

---

## queuemonitorlist

**Purpose** Displays an overview screen showing the status of a manager’s work queues.

**Configuration File**

webcomponent\config\library\queuemanagement\queue\_actions.xml

**Scope**

All.

**Caller**

menubar component

**Parameters**

ownerName	(Optional) Not used
-----------	---------------------

**queueusersandgroups (type dmc\_workqueue)**

**Purpose** Displays a list of the users and groups assigned to a work queue

**Configuration File**

webcomponent\config\actions\dmc\_workqueue\_actions.xml

**Scope**

type	dmc_workqueue
------	---------------

**Caller**

queuemonitorlist and queuelist components UI

**Parameters**

categoryPath	(Optional) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs.
objectId	(Required) ID of the task to assign.
queuename	(Required) Name of the work queue.

## reassign\_queued\_task (type dm\_task)

**Purpose** Changes the user assigned to work on a work queue task.

### Configuration File

webcomponent\config\actions\dm\_queued\_task\_actions.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

menubar, taskmanagercontainer components

### Parameters

---

objectId	(Required)
ownerName	(Optional) Not used
performer	(Required) Name of the performer to which to reassign the task.
queueName	(Required) Name of the work queue for which to assign a task.

---

## reassignuser (privilege superuser, type dm\_user)

**Purpose** Renames a user.



**Configuration File**

webcomponent\config\admin\user\dm\_user\_actions.xml

**Scope**

privilege	superuser
type	dm_user

**Caller**

menubar component

**Parameters**

objectId	(Required) Object ID of the user to reassign.
objectname	(Required) The name of the user.

**reassignuser (type dm\_member\_user)****Purpose** Renames a user.**Configuration File**

webcomponent\config\admin\group\dm\_member\_actions.xml

**Scope**

type	dm_member_user
------	----------------

## Caller

menubar component

## Parameters

---

objectId	(Required) ID of the selected user. Set to newobject to create a new user.
objectname	(Required) Name of the user.

---

# refresh (type dm\_sysobject)

**Purpose** Refreshes a reference or replica object to reflect changes in the object source

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

menubar component

## Parameters

---

isReplica	(Optional) Boolean parameter that specifies whether the object is a replica.
objectId	(Required) ID of the object to refresh (not ID of the source object.)

---

## registerevents (type dm\_sysobject)

**Purpose** Registers for notification when certain events occur on an object. Replica and reference objects cannot be registered.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

events	(Optional) List of the specific events for which to register.
objectId	(Required) Object ID of the object whose events should be registered for notification

---

## rejectroutertask (type dm\_router\_task)

**Purpose** Rejects a router.

### Configuration File

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

### Scope

---

type	dm_router_task
------	----------------

---

### Caller

taskmanagercontainer component.

### Parameters

---

canCompleteTask	(Optional) True to specify that the user can complete the task
objectId	(Required) ID of the router task.

---

## rejectworkflowtask (type dm\_task)

**Purpose**      Rejects a task in a workflow

### Configuration File

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

taskmanagercontainer component

## Parameters

---

canCompleteTask	(Optional) Specifies that the task is active and can be completed.
hasEmptyMandatoryPkg	(Optional) The task contains an empty package that must be passed with the task
hasRejectPaths	(Optional) Boolean flag that specifies whether there are any reject paths in the workflow
isManual	(Optional) Specifies whether transition is manual or automated.
objectId	(Required) ID of the task.
userCost	(Optional) Customer-defined cost data, stored as double in Server. Can be audited and summed up for completed workflows.
userTime	(Optional) Customer-defined time data, stored as integer in Server. Can be audited and summed up for completed workflows.

---

## relationships (type dm\_sysobject)

**Purpose** Shows relationships between objects.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar component

**Parameters**

---

folderId	(Optional) ID of the folder that contains the selected object.
folderPath	(Optional) Path to the folder that contains the selected object. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder
objectId	(Required) ID of the selected object.

---

**Elements**

---

<failoverenabled>	This component supports failover. To disable failover, enter a value of false.
-------------------	--

---

This action provides options for launching different components from the same action using the <dynamicfilter> element. For information on the <dynamicfilter> element and its sub-elements, see *WDK and Client Applications Development Guide*.

**Elements**

The <execution> element contains the following configurable element:

---

<navigation>	Type of navigation upon execution. Valid values: jump   returnjump   nested (default).
--------------	--

---

## removeaccessor (type dm\_ace)

**Purpose** Removes an accessor from an ACL

**Configuration File**

webcomponent\config\admin\acl\acl\_actions.xml

**Scope**


---

type	dm_ace
------	--------

---

**Caller**

Called by the Permissions component class

**Parameters**


---

accessorName	(Required) Name of accesor (user or group) to be removed
permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT=0DF_ EXTENDED_PERMIT = 1DF_ APPLICATION_PERMIT = 2DF_ACCESS_RESTRICTION = 3DF_EXTENDED_RESTRICTION = 4DF_APPLICATION_RESTRICTION = 5DF_REQUIRED_GROUP = 6DF_REQUIRED_GROUP_SET = 7

---

**removeattachment (type dmi\_package)****Purpose** Removes an attachment from a workflow package**Configuration File**

webcomponent\config\actions\dmi\_package\_actions.xml

### Scope

---

type	dmi_package
------	-------------

---

### Caller

taskattachment

### Parameters

---

attachmentName	(Optional) Name of the attachment
isPackageManufactured	(Optional) Boolean flag that specifies whether the package was added in this task or previously
objectId	(Required) ID of the workflow task.
packageName	(Required) Name of the workflow package.

---

## removecomponent (type dm\_sysobject)

**Purpose** Removes a child node from a virtual document.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

Not called



---

## Parameters

---

isCurrentVDMNode	(Optional) Boolean flag that specifies whether the node is the root node of the current virtual document. The menu item to remove a node is disabled if the node is the root node.
nodeId	(Optional) ID of the child node to be removed.
objectId	(Required) ID of the root node.

---

# removeprocessorauthentication

**Purpose** Removes the skill level and authorized document profile settings for a work queue processor.

## Configuration File

webcomponent\config\library\queuemanagement\userprofile\_actions.xml

## Scope

All.

## Caller

ProcessorAttributes component class

## removerecordrelationships (entitlement recordsmanager)

**Purpose** Deletes the relationship between formal records. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\records\relationships\dm\_relation\_actions.xml

### Scope

---

entitlement	recordsmanager
-------------	----------------

---

### Caller

menubar component

### Parameters

---

isRecordRelationship	(Required) Boolean: True indicates that the relationship is between formal records.
objectId	(Required) Object ID of the source formal record object.

---

## removerequredgroup (type dm\_requiredgroup)

**Purpose** Removes a group requirement from an ACL. (Requires that Trusted Content Services be enabled in the repository.)

---

**Configuration File**

webcomponent\config\admin\acl\acl\_actions.xml

**Scope**

---

type	dm_requiredgroup
------	------------------

---

**Caller**

aclacelist component

**Parameters**

---

accessorName	(Required) Group to be removed from ACL
permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT DF_EXTENDED_PERMIT DF_APPLICATION_PERMIT DF_ACCESS_RESTRICTION DF_EXTENDED_RESTRICTION DF_APPLICATION_RESTRICTION DF_REQUIRED_GROUP DF_REQUIRED_GROUP_SET

---

## removerequiredgroupset (type dm\_requiredgroupset)

**Purpose**

Removes a set of required groups from an ACL. (Requires that Trusted Content Services be enabled in the repository.)

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

---

type	dm_requiredgroupset
------	---------------------

---

### Caller

aclacelist component

### Parameters

---

accessorName	(Required) Name of group set to remove
permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT=0DF_ EXTENDED_PERMIT = 1DF_ APPLICATION_PERMIT = 2DF_ACCESS_RESTRICTION = 3DF_EXTENDED_RESTRICTION = 4DF_APPLICATION_RESTRICTION = 5DF_REQUIRED_GROUP = 6DF_REQUIRED_GROUP_SET = 7

---

## removerestriction (type dm\_restriction)

**Purpose** Removes an access restriction for selected users or group. Trusted Content Services must be enabled in the repository.

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

**Scope**


---

type	dm_restriction
------	----------------

---

**Caller**

aclacelist component

**Parameters**


---

accessorName	(Required) Name of user or group for which to remove the restriction
permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT=0DF_ EXTENDED_PERMIT = 1DF_ APPLICATION_PERMIT = 2DF_ACCESS_RESTRICTION = 3DF_EXTENDED_RESTRICTION = 4DF_APPLICATION_RESTRICTION = 5DF_REQUIRED_GROUP = 6DF_REQUIRED_GROUP_SET = 7

---

**removestartwfattachment (type dmi\_package)****Purpose** Removes an attachment before the workflow has started**Configuration File**

webcomponent\config\actions\dmi\_package\_actions.xml

**Scope**


---

type	dmi_package
------	-------------

---

## Caller

startwfwattachment component UI

## Parameters

---

attachmentName	(Optional) Name of the attachment
objectId	(Required) ID of the workflow task.
packageName	(Required) Name of the workflow package.

---

# removestartwfwattachment (type dmi\_wf\_attachment)

**Purpose** Removes an attachment before the workflow has started

## Configuration File

webcomponent\config\actions\dmi\_wf\_attachment\_actions.xml

## Scope

---

type	dmi_wf_attachment
------	-------------------

---

## Caller

startattachment component UI

**Parameters**

attachmentName	(Optional) (Required) Name of the attachment to be removed from the workflow.
docId	(Required) ID of the attachment.
objectId	(Required) ID of the workflow process.

**removeuserorgroup (type dm\_member\_group)**

**Purpose** Allows the administrator to remove a group from the repository

**Configuration File**

webcomponent\config\admin\group\dm\_member\_actions.xml

**Scope**

type	dm_member_group
------	-----------------

**Caller**

menubar component

**Parameters**

groupId	(Required) ID of the group to remove.
---------	---------------------------------------

## removeuserorgroup (type dm\_member\_user)

**Purpose** Allows the administrator to remove a user from the repository

### Configuration File

webcomponent\config\admin\group\dm\_member\_actions.xml

### Scope

---

type	dm_member_user
------	----------------

---

### Caller

menubar component

### Parameters

---

groupId	(Required) ID of the user to remove.
---------	--------------------------------------

---

## removevirtualdocumentnode (type dm\_sysobject)

**Purpose** Removes a node from a virtual document

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml



**Scope**

type	dm_sysobject
------	--------------

**Caller**

menubar component

**Parameters**

isCurrentVDMNode	(Optional) Set to true if the object to be removed is the current virtual document node (node cannot delete itself)
nodeId	(Required) ID of the node to be removed
vdmRootObjectId	(Required) ID of the virtual document root

## removewfattachment (type dmi\_wf\_attachment)

**Purpose** Removes an attachment from a workflow

**Configuration File**

webcomponent\config\actions\dmi\_wf\_attachment\_actions.xml

**Scope**

type	dmi_wf_attachment
------	-------------------

**Caller**

taskattachment component UI

**Parameters**

---

attachmentName	(Optional) Name of the attachment to be removed from the workflow
objectId	(Required) ID of the queue item.
packageName	(Optional) Name of the workflow package
wf_attachment_id	(Required) ID of the attachment to be removed from the workflow.

---

## removeworkqueuemember

**Purpose** Removes a user from the list of valid processors for a work queue

**Configuration File**

webcomponent\config\library\queuemanagement\queueusersandgroups\_actions.xml

**Scope**

All.

**Caller**

menubar component

**Parameters**

---

addQueueMemberContext	(Required) Boolean that must have a value of true to perform action
queueName	(Required) Name of queue from which to remove a user.

---

## rename (type dm\_folder)

**Purpose** Renames a folder in the repository. The user must have a role of contributor or higher and write\_permit access to the parent folder or cabinet (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_folder\_actions.xml

### Scope

type	dm_folder
------	-----------

### Caller

menubar

### Parameters

objectId	(Required) The ID of the folder to be renamed.
ownerName	(Optional) User who owns the folder

## rename (type dm\_sysobject)

**Purpose** Renames an object in the repository. The user must have a role of contributor or higher and write\_permit access to the parent folder or cabinet (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

Not called

### Parameters

---

lockOwner	(Optional) User who has checked out the object
objectId	(Required) The ID of the object to be renamed.
ownerName	(Optional) User who owns the checked in object

---

## rendition\_transformation (type dmr\_content)

**Purpose** Requires Media Transformation Services. Customization not supported.

### Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

### Scope

---

type	dmr_content
------	-------------

---

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**


---

contentType	(Optional) File format to use when transforming the object (passed to the operating system)
folderId	(Optional) ID of the folder from which the action was called
objectId	(Required) ID of the selected object.
ownerName	(Optional) Name of the object owner
transformableToRendition	(Optional) The valid rendition to which the object can be transformed

---

**renditions (type dm\_sysobject)**

**Purpose** Displays the renditions for a selected object

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**


---

type	dm_sysobject
------	--------------

---

**Caller**

menubar component

## Parameters

---

contentType	(Optional) The content type of the renditions to display.
folderId	(Optional) ID of the folder that contains the selected object.
folderPath	(Optional) Path to the folder that contains the selected object. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder
objectId	(Required) ID of the selected object.

---

## Elements

The configurable elements are the same as those for [locations](#), page 837.

# reordercomponents (type dm\_sysobject)

**Purpose** Changes the order of nodes in a virtual document

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

menubar component

**Parameters**

compoundArchitecture	(Optional) Boolean specifying whether the object is a compound virtual document
isVirtualDoc	(Optional) Boolean flag that specifies whether the object is a virtual document (required for pre-4.1 repository virtual documents)
linkCount	(Optional) Number of links (child nodes) in a document. If greater than 0, the document is a virtual document.
lockOwner	(Optional) User who has checked out the object
nodeId	(Optional) ID of the child node to be removed
objectId	(Required) ID of the object to be converted into a virtual document.

**reordervirtualdocumentnodes (type dm\_sysobject)**

**Purpose** Changes the order of nodes in a virtual document

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

**Caller**

menubar component

## Parameters

---

nodeId	(Required) ID of the node to be changed
vdmRootObjectId	(Required) ID of the virtual document root node

---

# repeatworkflowtask (type dm\_task)

**Purpose** Repeats a task in a workflow

## Configuration File

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

## Scope

---

type	dm_task
------	---------

---

## Caller

taskmanagercontainer component

## Parameters

---

canCompleteTask	(Optional) Specifies that the task is active and can be completed.
isFailedAutoTask	(Optional) Specifies that an automatic task has failed and can be re-run.
isRepeatable	(Optional) Boolean flag that specifies whether a task can be repeated
objectId	(Required) ID of the task.



---

userCost	(Optional) Customer-defined cost data, stored as double in Server. Can be audited and summed up for completed workflows.
userTime	(Optional) Customer-defined time data, stored as integer in Server. Can be audited and summed up for completed workflows.

---

## reportdetailsauditdrilldown (type dm\_workflow)

**Purpose** Displays several workflow report components.

### Configuration File

webcomponent\config\actions\dm\_workflow\_actions.xml

### Scope

---

type	dm_workflow
------	-------------

---

### Caller

This action is called by a link in the component [reportmaindrilldown](#), page 967.

### Parameters

---

objectId	(Required) ID of the selected workflow.
----------	---

---

## reportdetailsauditlist (type dm\_workflow)

**Purpose** Displays several workflow report components.

### Configuration File

webcomponent\config\actions\dm\_workflow\_actions.xml

### Scope

---

type	dm_workflow
------	-------------

---

### Caller

This action is called by a link in the component [reportmainlist](#), page 968.

### Parameters

---

objectId	(Required) ID of the selected workflow.
----------	---

---

## reportdetailsmap (type dm\_workflow)

**Purpose** Displays several workflow report components

### Configuration File

webcomponent\config\actions\dm\_workflow\_actions.xml

**Scope**

---

type	dm_workflow
------	-------------

---

**Caller**

menubar component.

**Parameters**

---

objectId	(Required) ID of the selected workflow.
----------	---

---

## reportdetailsummarydrilldown (type dm\_workflow)

**Purpose** Displays several workflow report components

**Configuration File**

webcomponent\config\actions\dm\_workflow\_actions.xml

**Scope**

---

type	dm_workflow
------	-------------

---

**Caller**

reportmaindrilldown component

**Parameters**

---

objectId	(Required) Object ID of the dm_workflow object to report on.
----------	--

---

## reportdetailssummarylist (type dm\_workflow)

**Purpose** Displays several workflow report components.

### Configuration File

webcomponent\config\actions\dm\_workflow\_actions.xml

### Scope

---

type	dm_workflow
------	-------------

---

### Caller

This action is called by a link in the component [reportmainlist](#), page 968.

### Parameters

---

objectId	(Required) ID of the selected workflow.
----------	---

---

## reportmaindrilldown

**Purpose** Displays a list of workflow instances and their attributes.

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

---

# reportmainsettings

**Purpose** Displays report settings.

## Configuration File

webcomponent\config\actions\generic\_actions.xml

## Scope

All.

## Caller

This action is called by a link in the component [reportmaindrilldown](#), page 967.

## Parameters

---

documentId	(Optional) Specifies a document for which workflows will be reported
filter	(Required) Sets the selected filter. Valid values: 0 = all   1 = completed   2 = running
overdueDays	(Optional) Specifies the number of days overdue for which to display a report
overdueOnly	(Required) True to filter for overdue task status.
templateId	(Optional) Specifies a template for which workflows will be reported
userName	(Required) Name of the user for which to display workflows.

---

## repositioncomponent (type dm\_sysobject)

**Purpose** Rearranges the structure of a virtual document

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

Not called

### Parameters

---

insertAfterNodeId	(Optional) ID of the virtual document node after which to move the object.
nodeId	(Required) Node ID of the current parent node for the object.
preconditionShouldDisplayErrors	(Optional) Set to true to display errors on evaluation of the precondition
targetParentNodeId	(Required) Node ID of the new parent node for the object.
vdmRootObjectId	(Required) Object ID of the root of the virtual document.

---

## rerunfailedautoworkflowtask (type dm\_task)

**Purpose** Reruns a failed automatic workflow task

### Configuration File

webcomponent\config\library\workflow\taskmanager\taskmgr\_actions.xml

### Scope

type	dm_task
------	---------

### Caller

taskmanagercontainer component

### Parameters

canCompleteTask	(Optional) Specifies that the task is active and can be completed.
isFailedAutoTask	(Optional) Specifies that an automatic task has failed and can be re-run.
objectId	(Required) ID of the task.
userCost	(Optional) Customer-defined cost data, stored as double in Server. Can be audited and summed up for completed workflows.
userTime	(Optional) Customer-defined time data, stored as integer in Server. Can be audited and summed up for completed workflows.

## restorelifecycle (type dm\_sysobject)

**Purpose** Restores an object from the suspended state to the active state (resumes previous state)

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

lockOwner	(Optional) User who has checked out the object
objectId	(Required) The ID of the object.

---

## resumetask (type dm\_task)

**Purpose** Resumes a task in a workflow that was previously halted

### Configuration File

webcomponent\config\actions\dm\_task\_actions.xml



**Scope**

type	dm_task
------	---------

**Caller**

reportdetailssummarylist component

**Parameters**

objectName	(Required) Task name.
taskRuntimeFlag	(Required) State of the workflow task. Valid values: 0=dormant   1=acquired   2=finished   3 – 6 =paused   7=ready
workitemId	(Required) Object ID of the work item to resume.

**resumeworkflow (type dm\_workflow)**

**Purpose** Resumes a halted workflow

**Configuration File**

webcomponent\config\actions\dm\_workflow\_actions.xml

**Scope**

type	dm_workflow
------	-------------

**Caller**

menubar, reportdetailssummarydrilldown, reportdetailssummarylist, reportmainlist components

## Parameters

---

objectId	(Required) ID of the workflow.
supervisor	(Optional) User name of workflow supervisor
workflowRuntimeState	(Required) State of the workflow. 0=dormant   1=running   2=finished   3=halted   4=terminated

---

## rmaccessor (type dm\_ace)

**Purpose** Removes an entry from an ACL

### Configuration File

webcomponent\config\admin\acl\acl\_actions.xml

### Scope

---

type	dm_ace
------	--------

---

### Caller

aclacelist, permissions components

### Parameters

---

accessorName	(Required) Name of user or group to remove from the ACL.
--------------	--

---

## roommembers (type dmc\_room)

**Purpose** Displays the members of a room. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\rooms\room\_actions.xml

### Scope

type	dmc_room
------	----------

### Caller

doclist component

### Parameters

enableShowAll	(Optional) Not used
objectId	(Required) ID of the room

## saveasxforms (type dm\_folder)

**Purpose** Creates a new form from a current form. The user must have the role form\_user. Launched as a menu item for a selected object in the list view. Customization is not supported.

### Configuration File

webcomponent\config\actions\dm\_folder\_actions.xml

### Scope

---

type	dm_folder
------	-----------

---

### Caller

menubar and component pages that contain actionlinklist tag

### Parameters

---

contentType	(Optional) File format of the object (passed to the operating system)
objectId	(Required) Object ID of the current folder in which the form will be created.
ownerName	(Optional) Not implemented

---

## saveasxforms (type dm\_sysobject)

**Purpose**      Creates a new form from a current form. The user must have the role form\_user.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

## Parameters

---

contentType	(Optional) File format of the object (passed to the operating system)
objectId	(Required) Object ID of the current folder in which the form will be created.
ownerName	(Optional) Owner of object

---

# savechanges

**Purpose** Saves changes made to a virtual document structure

## Configuration File

webcomponent\config\actions\generic\_actions.xml

## Scope

All.

## Caller

menubar component

## Parameters

---

closeUponCompletion	(Optional) Set to true to close virtual document structure after saving
objectId	(Optional) Object ID of the virtual document to save.
ownerName	(Optional) Owner of the virtual document

---

## saveSearch

**Purpose** Saves a search query

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

### Caller

search component

### Parameters

---

queryId	(Optional) Internal ID assigned to the query
---------	--

---

## search

**Purpose** Performs a search based on a query.

### Configuration File

webcomponent\config\actions\generic\_actions.xml

**Scope**

All.

**Caller**

Not called

**Parameters**

query	(Required) Query or DQL string
queryType	(Optional) Specifies whether the query is a String value or DQL. Valid values: string   dql
ranking	(Optional) Boolean flag. True to display the ranking of search results.

**searchuser (type dm\_user)****Purpose** Displays a list of users**Configuration File**

webcomponent\config\admin\user\dm\_user\_actions.xml

**Scope**

type	dm_user
------	---------

**Caller**

Not called

## Elements

The <execution> element contains the following configurable element:

---

<navigation>	Type of navigation upon execution. Valid values: jump   returnjump   nested (default).
--------------	--

---

## selectrepository

**Purpose** This action allows the user to select a repository from a list of preferred repositories. Displays visible repositories for the list of connection brokers. The user can manually enter the name of a connection broker to select a repository that connects to that broker.

### Configuration File

wdk\config\session\_actions.xml

### Scope

All.

### Caller

Called from a link in the login page and menu items in the streamline and classic views (**Add Repository**)

## selectuser (type selectuser)

**Purpose** Selects a user



### Configuration File

webcomponent\config\admin\user\selectuser\_action.xml

### Scope

---

type	selectuser
------	------------

---

### Caller

Not called

## sendlocator (type dm\_notification)

**Purpose** Sends one or more object locators through email

### Configuration File

webcomponent\config\actions\dm\_notification\_actions.xml

### Scope

---

type	dm_notification
------	-----------------

---

### Caller

menubar and component pages that contain actionlinklist tag

## Parameters

---

objectId	(Required) ID of the object to send.
type	(Optional) Specifies the object type. If null, an object fetch will determine the type.

---

# sendlocator (type dm\_queued)

**Purpose** Sends one or more object locators through email

## Configuration File

webcomponent\config\actions\dm\_queued\_actions.xml

## Scope

---

type	dm_queued
------	-----------

---

## Caller

menubar component

## Parameters

---

objectId	(Required) ID of the object to send.
type	(Optional) Specifies the object type. If null, an object fetch will determine the type.

---

## sendlocator (type dm\_router\_task)

**Purpose** Sends one or more object locators through email

### Configuration File

webcomponent\config\actions\dm\_router\_task\_actions.xml

### Scope

type	dm_router_task
------	----------------

### Caller

menubar component

### Parameters

objectId	(Required) ID of the object to send.
type	(Optional) Specifies the object type. If null, an object fetch will determine the type.

## sendlocator (type dm\_sysobject)

**Purpose** Sends one or more object locators through email

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the object to send.
type	(Optional) Specifies the object type. If null, an object fetch will determine the type.

---

## sendlocator (type dm\_task)

**Purpose** Sends one or more object locators through email

### Configuration File

webcomponent\config\actions\dm\_task\_actions.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

menubar component

**Parameters**

---

objectId	(Required) ID of the object to send.
type	(Optional) Specifies the object type. If null, an object fetch will determine the type.

---

## sendto (type dm\_sysobject)

**Purpose** Sends an object to a user or group

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

Not called

**Parameters**

---

objectId	(Required) ID of the object to send.
----------	--------------------------------------

---

## sendtodistributionlist (type dm\_sysobject)

**Purpose** Sends an object to a distribution list

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the object to send.
----------	--------------------------------------

---

## setbindingrule (type dm\_sysobject)

### Purpose

Allows a user to set a binding rule on a virtual document node. The setbindingrule precondition determines that the selected object is a virtual document node. Also, the precondition checks that the user has at least version permission on the virtual document parent node.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar, vdmList, vdmListStreamline components

**Parameters**

---

nodeId	(Required) ID of the document node on which to set a binding rule.
vdmRootObjectId	(Required) ID of virtual document root

---

## showjobstatus

**Purpose** Displays the status of running asynchronous jobs.

**Configuration File**

webcomponent\config\actions\generic\_actions.xml

**Scope**

All.

**Caller**

This action is called by the Webtop statusbar component.

## showjobstatushistory

**Purpose** Displays the history of asynchronous jobs.

## Configuration File

webcomponent\config\actions\generic\_actions.xml

## Scope

All.

## Caller

This action is called by the component [jobstatus](#), page 826.

## Parameters

---

jobId	(Required) ID of the selected job.
jobName	(Required) Name of the selected job.

---

# showtopic (entitlement collaboration, type dm\_sysobject)

**Purpose** Displays a discussion with threaded topics. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\discussion\topic\_actions.xml

## Scope

---

entitlement	collaboration
type	dm_sysobject

---



**Caller**

Called from the showtopicaction execution class (ViewTopic)

**Parameters**

---

folderId	(Optional) ID of folder containing a discussion
folderPath	(Optional) Path to folder containing a discussion
objectId	(Required) ID of the discussion.
topicStatus	(Optional) Valid values of TopicStatusAttributeDataHandler: 0 (STATUS_DISABLED)   1 (STATUS_NONE)   2 (STATUS_NO_COMMENTS)   3 (STATUS_READ)   4 (STATUS_UNREAD)
type	(Optional) Not used

---

## showtopicaction (entitlement collaboration, type dm\_sysobject)

**Purpose** Displays a topic. Requires Collaborative Edition.

**Configuration File**

webcomponent\config\library\discussion\topic\_actions.xml

**Scope**

---

entitlement	collaboration
type	dm_sysobject

---

**Caller**

menubar, myobjectsdrilldown, myobjectslist, doclist, searchresultsdrilldown, searchresultslist, subscriptionslist, subscriptionsdrilldown, drilldown, objectgrid, vdmlist, vdmstreamline, component.

**Parameters**

---

folderId	(Optional) ID of the current folder
objectId	(Required) ID of the topic's parent
topicStatus	(Optional) Valid values of TopicStatusAttributeDataHandler: 0 (STATUS_DISABLED)   1 (STATUS_NONE)   2 (STATUS_NO_COMMENTS)   3 (STATUS_READ)   4 (STATUS_UNREAD)
type	(Optional) Object type of the selected object

---

## startworkflow (type dm\_process)

**Purpose**

Starts a workflow using a selected template. The user starts the workflow by selecting a document. The startworkflow component is added to the container dynamically. The user must have a contributor level of client capability.

**Configuration File**

webcomponent\config\actions\dm\_process\_actions.xml

**Scope**

---

type	dm_process
------	------------

---

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**

attachmentIds	(Optional) IDs of the objects attached to the workflow
objectId	(Required) ID of the workflow template.
ownerName	(Optional) Name of the user who starts the workflow
startworkflowId	(Required) Component ID of the task manager component to be displayed as the first tab in the container.

**startworkflowfromdoc (type dm\_sysobject)****Purpose**

Starts a workflow with the selected document. The user starts the workflow by selecting a workflow template. The user must have a contributor level of client capability.

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

**Caller**

menubar component

**Parameters**

---

objectId	(Required) ID of the object to route through the workflow
ownerName	(Optional) Name of the user who starts the workflow
startworkflowId	(Optional) Component ID of the task manager component to be displayed as the first tab in the container

---

## startworkflownotemplate

**Purpose** Starts a workflow without a workflow template

**Configuration File**

webcomponent\config\actions\generic\_actions.xml

**Scope**

All.

**Caller**

menubar, myobjectsdrilldown, subscriptionsdrilldown, inboxdrilldown components

**Parameters**

---

ownerName	(Optional) Owner of the object for which to start a workflow
startworkflowId	(Optional) Component ID of the taskmanager component to be displayed as the first tab in the container.

---

## Elements

---

<precondition>.	Precondition argument. Default = startworkflowId
<argument>	

---

## streamlineviewassembly (type dm\_sysobject)

**Purpose** Displays in drilldown (streamline) view a list of virtual document snapshots for the selected object

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

Extended by the assemblyliststreamline action but not called directly

### Parameters

---

assembledFromId	(Optional) ID of the virtual document from which the assembly was assembled
home_component	(Optional) In Portal application, specifies portal component that calls this component. Used to construct a breadcrumb back to the portal.

home_component_args	(Optional) In Portal application, specifies arguments that are passed to the portal component that calls this component when the user clicks the breadcrumb to go back.
objectId	(Required) ID of the assembly

---

## streamlineviewvirtualdoc (type dm\_sysobject)

**Purpose** Displays a virtual document in the drilldown view

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

Extended by the vdmstreamline action but not called directly

### Parameters

---

home_component	(Optional) In Portal application, specifies portal component that calls this component. Used to construct a breadcrumb back to the portal.
home_component_args	(Optional) In Portal application, specifies arguments that are passed to the portal component that calls this component when the user clicks the breadcrumb to go back.

---

isCurrentVDMNode	(Optional) Boolean flag that specifies whether the node is the root node of the current virtual document. The menu item to remove a node is disabled if the node is the root node.
isVirtualDoc	(Optional) Boolean flag that specifies whether the object is a virtual document (required for pre-4.1 repository virtual documents)
linkCount	(Optional) Number of links (child nodes) in a document. If greater than 0, the document is a virtual document.
objectId	(Required) ID of the virtual document.

---

## submitforcategorization (type dm\_document)

**Purpose** Submits a document for categorization by the CI Server. The repository must have Content Intelligence Services installed and configured.

### Configuration File

webcomponent\config\actions\dm\_document\_actions.xml

### Scope

---

type	dm_document
------	-------------

---

### Caller

menubar and component pages that contain actionlinklist tag

### Parameters

---

objectId	(Required) ID of the selected object.
----------	---------------------------------------

---

## subscribe (type dm\_sysobject)

**Purpose** Subscribes the user to an object

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

folderPath	(Optional) Path to object
objectId	(Required) ID of the selected object.

---

## suspend\_queued\_task (type dm\_task)

**Purpose** Suspends a task from a work queue.



**Configuration File**

webcomponent\config\actions\dm\_queued\_task\_actions.xml

**Scope**

type	dm_task
------	---------

**Caller**

menubar component

**Parameters**

objectId	(Required) ID of the task
runtimeState	(Required) State of the workflow. 0=dormant   1=running   2=finished   3=halted   4=terminated. Only 0 or 1 can be suspended.

**suspendlifecycle (type dm\_sysobject)****Purpose** Suspends the lifecycle state of the selected object**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

### Caller

menubar component

### Parameters

---

lockOwner	(Optional) User who has checked out the object
objectId	(Required) ID of the selected object.

---

## testcustomattrs (type wdk5\_testtype\_1)

**Purpose** One of a set of components and actions that test controls and actions. This action can be used for internal testing to display custom attributes.

### Configuration File

wdk\config\fxtest\_component.xml

### Parent Definition

versions:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

---

type	wdk5_testtype_1
------	-----------------

---

### Caller

figservicetest component

## testcustomrepeatingattrs (type wdk5\_testtype\_2)

**Purpose** One of a set of components and actions that test controls and actions. This action can be used for internal testing of repeating attributes display.

### Configuration File

wdk\config\fxtest\_component.xml

### Parent Definition

versions:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

type	wdk5_testtype_2
------	-----------------

### Caller

configservicetest

## testdocbasescope (docbase CS52ORABuild162wdkForm not w2k52WDK\_CustomTest01, type wdk5\_testtype\_3)

**Purpose** One of a set of components and actions that tests controls and actions. This action can be used for internal testing of scope on repository and type values.

### Configuration File

wdk\config\fxtest\_component.xml

### Parent Definition

renditions:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

---

docbase	CS52ORABuild162, wdkForm, not w2k52WDK_CustomTest01
type	wdk5_testtype_3

---

### Caller

figservicetest component

## testfilterattrs (type wdk5\_testtype\_3)

**Purpose** One of a set of components and actions that test controls and actions. This action can be used for internal testing to filter an action by a scope value.

### Configuration File

wdk\config\fxtest\_component.xml

### Parent Definition

versions:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

---

type	wdk5_testtype_3
------	-----------------

---

**Caller**

configservicetest component

**testmultirole (role consumer not administrator, type wdk5\_testtype\_4)**

**Purpose**

One of a set of components and actions that test controls and actions. This action can be used for internal testing of roles.

**Configuration File**

wdk\config\fxtest\_component.xml

**Parent Definition**

renditions:webcomponent/config/actions/dm\_sysobject\_actions.xml

**Scope**

---

role	consumer, not administrator
type	wdk5_testtype_4

---

**Caller**

configservicetest component

## testmultiscope (type wdk5\_testtype\_1 wdk5\_testtype\_2)

**Purpose** One of a set of components and actions that test controls and actions. This action can be used for internal testing of multiple scope values.

### Configuration File

wdk\config\fxtest\_component.xml

### Parent Definition

renditions:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

---

type	wdk5_testtype_1, wdk5_testtype_2
------	----------------------------------

---

### Caller

configservicetest component

## testmultiscopeandnot (type wdk5\_testtype\_1 not wdk5\_testtype\_2)

**Purpose** One of a set of components and actions that test controls and actions. This action can be used for internal testing of multiple scopes and excluding values.

### Configuration File

wdk\config\fxtest\_component.xml

**Parent Definition**

renditions:webcomponent/config/actions/dm\_sysobject\_actions.xml

**Scope**

---

type	wdk5_testtype_1, not wdk5_testtype_2
------	--------------------------------------

---

**Caller**

configservicetest

## testnotkeyword (type not wdk5\_testtype\_1)

**Purpose** One of a set of components and actions that test controls and actions. This action can be used for internal testing of scope that excludes certain values.

**Configuration File**

wdk\config\fxtest\_component.xml

**Parent Definition**

history:webcomponent/config/actions/dm\_sysobject\_actions.xml

**Scope**

---

type	not wdk5_testtype_1
------	---------------------

---

**Caller**

configservicetest component

## transformation (type dm\_sysobject)

**Purpose** Requires Media Transformation Services. Customization not supported.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

contentType	(Optional) Customization not supported.
folderId	(Optional) Customization not supported.
objectId	(Required) Customization not supported.
ownerName	(Optional) Customization not supported.

---

## transformationtonewrelation (type dm\_sysobject)

**Purpose** Requires Media Transformation Services. Customization not supported.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml



**Scope**

type	dm_sysobject
------	--------------

**Caller**

Not called

**Parameters**

contentType	(Optional) Customization not supported.
folderId	(Optional) Customization not supported.
objectId	(Required) Customization not supported.
ownerName	(Optional) Customization not supported.
transformableToRendition	(Optional) Customization not supported.

**transformationtonewrelation (type dmr\_content)****Purpose** Requires Media Transformation Services. Customization not supported.**Configuration File**

webcomponent\config\actions\dmr\_content\_actions.xml

**Scope**

type	dmr_content
------	-------------

**Caller**

None

**Parameters**

---

contentType	(Optional) File format to use when opening the object (passed to the operating system)
folderId	(Optional) ID of the folder from which the action was called
objectId	(Required) ID of the selected object.
ownerName	(Optional) Name of the object owner
transformableToRendition	(Optional) Not used

---

**transformationtonewrendition (type dm\_sysobject)**

**Purpose** Requires Media Transformation Services. Customization not supported.

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

Not called

**Parameters**

---

contentType	(Optional) Customization not supported.
folderId	(Optional) Customization not supported.
objectId	(Required) Customization not supported.

---

ownerName	(Optional) Customization not supported.
transformableToRendition	(Optional) Customization not supported.

---

## transformationtonewrendition (type dmr\_content)

**Purpose** Transforms the object to the selected rendition. Requires Media Transformation Services. Customization not supported.

### Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

### Scope

---

type	dmr_content
------	-------------

---

### Caller

None

### Parameters

---

contentType	(Optional) File format to use when transforming the object (passed to the operating system)
folderId	(Optional) ID of the folder from which the action was called
objectId	(Required) ID of the selected object.
ownerName	(Optional) Name of the object owner
transformableToRendition	(Optional) The valid rendition to which the object can be transformed

---

## unassign\_queued\_task (type dm\_task)

**Purpose** Removes a work queue task from an individual processor's task list and returns it to the work queue.

### Configuration File

webcomponent\config\actions\dm\_queued\_task\_actions.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the task
ownerName	(Optional) Task owner
performer	(Required) Performer to whom the task was assigned.
queueName	(Required) Name of work queue

---

## unfreezeassembly (type dm\_sysobject)

**Purpose** Unfreezes a frozen virtual document snapshot. The user must have a contributor role assigned in the repository.

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

**Caller**

menubar component

**Parameters**

isFrozenAssembly	(Optional) Boolean parameter that specifies whether the snapshot to be unfrozen is frozen
lockOwner	(Optional) User who has checked out the snapshot
objectId	(Required) ID of the virtual document snapshot.
ownerName	(Optional) User who owns the snapshot

**ungovern (entitlement collaboration)****Purpose** Causes an object to become ungoverned by a room. Requires Collaborative Edition.**Configuration File**

webcomponent\config\library\rooms\room\_actions.xml

### Scope

---

entitlement	collaboration
-------------	---------------

---

### Caller

menubar component

### Parameters

---

objectId	(Required) ID of the selected object.
roomId	(Required) ID of the room that governs the object.

---

## ungovernfolder (entitlement collaboration)

**Purpose** Causes a folder to become ungoverned by a room. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\rooms\room\_actions.xml

### Scope

---

entitlement	collaboration
-------------	---------------

---

### Caller

Called by the ungovern action class, UngovernAction, for folder types

**Parameters**

objectId	(Required) ID of the selected folder.
ownerName	(Optional) User who owns the selected folder.
roomId	(Required) ID of the room that currently governs the folder.

**unregisterevents (type dm\_sysobject)**

**Purpose** Removes registration for notification when certain events occur on an object

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

**Caller**

menubar component

**Parameters**

events	(Optional) List of specific events for which to remove registration.
objectId	(Required) Object ID of the object from which to remove event registration.

## unsubscribe (type dm\_sysobject)

**Purpose** Unsubscribes an object

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar, categorydrilldown components

### Parameters

---

events	(Optional) List of the events for which to remove subscriptions.
objectId	(Required) ID of the object.

---

## unsuspend\_queued\_task (type dm\_task)

**Purpose** Returns a suspended queue task to active state.

### Configuration File

webcomponent\config\actions\dm\_queued\_task\_actions.xml



**Scope**

---

type	dm_task
------	---------

---

**Caller**

menubar component

**Parameters**

---

objectId	(Required) Task ID
runtimeState	(Required) State of the workflow. 0=dormant   1=running   2=finished   3=halted   4=terminated. Only 0 or 1 can be suspended.

---

## updateprocessorauthentication

**Purpose** Updates the skill level and authorized document profiles for a work queue processor

**Configuration File**

webcomponent\config\library\queuemanagement\userprofile\_actions.xml

**Scope**

All.

**Caller**

queueusersandgroups component

## Parameters

---

docprofile_name	(Optional) Name to assign to the profile
skill	(Optional) Adds a value to the dropdown list of skills in the form of an NLS ID
username	(Required) Sets a label for the User Name, in the form of an NLS ID
wq_name	(Required) Work queue name

---

## userbrowse (privilege sysadmin, type dm\_user)

**Purpose** Allows the administrator to browse a list of users in the repository

### Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

### Scope

---

privilege	sysadmin
type	dm_user

---

### Caller

userrename component

## userimport (privilege sysadmin, type dm\_user)

**Purpose** Imports a user from another repository in the domain

**Configuration File**

webcomponent\config\admin\user\dm\_user\_actions.xml

**Scope**

---

privilege	sysadmin
type	dm_user

---

**Caller**

menubar component

## userorgroupchooser

**Purpose** Displays a locator component for selecting users or groups for use with work queues

**Configuration File**

webcomponent\config\library\queuemanagement\queue\_actions.xml

**Scope**

All.

**Caller**

queuemanagement utility class (QueueManagementUtil), called by various queuemanagement component classes to launch the wfuserorgroupfromgrouplocatorcontainer component.

## Parameters

---

groupName	(Required) Name of group from which to locate a member user or group
-----------	--

---

# userrenamelog (type dm\_user)

**Purpose** Displays job requests for renaming users.

## Configuration File

webcomponent\config\admin\user\dm\_user\_actions.xml

## Scope

---

type	dm_user
------	---------

---

## Caller

This action is called as a menuitem from the menubar component.

## Parameters

---

renametarget	(Optional) Type of objects requested for rename. Valid values: USER   GROUP
--------------	---

---

## Elements

The <execution> element contains the following configurable element:

---

<navigation>	Type of navigation upon execution. Valid values: jump   returnjump   nested (default).
--------------	--

---

## vdmclickactionprompt (type dm\_sysobject)

**Purpose** Displays a dialog box allowing the user to view a virtual document structure or edit the document

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

openvirtualdocument execution class

### Parameters

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system)
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
launchViewer	(Optional) Boolean flag that indicates whether to launch a viewing application.
nodeId	(Optional) ID of the node in a virtual document
objectId	(Required) ID of the object to be viewed.
pathInfo	(Optional) Path to be displayed in the breadcrumb

selectedNodeIds	(Optional) Comma-separated list of nodeIds that should be initially selected when the vdmList component is first displayed
structureComponent	(Optional) Name of component to be used for displaying the virtual document structure
useAssembly	(Optional) Set to true to display a list of assemblies. Set to false to display a list of virtual documents.

---

## Elements

The configurable elements are the same as those in [doclist](#), page 684.

# vdmcopyoption (type dm\_sysobject)

**Purpose**       Presents the user with options for the type of copy operation on a virtual document

## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

clipboard component class

---

**Parameters**

objectName	(Required) Name of the object that will be copied.
------------	--

**Elements**

The configurable elements are the same as those in [doclist](#), page 684.

**vdmlist (type dm\_sysobject)**

**Purpose** Displays a list of virtual documents in a list view

**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Parent Definition**

viewvirtualdoc:webcomponent/config/actions/dm\_sysobject\_actions.xml

**Scope**

type	dm_sysobject
------	--------------

**Caller**

Not called

**Elements**

The configurable element is the same as that for [renditions \(type dm\\_sysobject\)](#), page 397.

## vdmliststreamline (type dm\_sysobject)

**Purpose** Displays a list of virtual documents in a streamline (drilldown) view

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Parent Definition

streamlineviewvirtualdoc:webcomponent/config/actions/dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

Not called

### Elements

The configurable element is the same as that for [renditions \(type dm\\_sysobject\)](#), page 397.

### Elements

The configurable element is the same as that for [renditions \(type dm\\_sysobject\)](#), page 397.

## versions (type dm\_sysobject)

**Purpose** Displays the versions of an object. Versions are disabled for folders and cabinets



## Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

## Scope

type	dm_sysobject
------	--------------

## Caller

menubar component

## Parameters

folderId	(Optional) ID of the folder that contains the selected object.
folderPath	(Optional) Path to the folder that contains the selected object. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder
objectId	(Required) ID of the selected object.

## Elements

<failoverenabled>	This component supports failover. To disable failover, enter a value of false.
-------------------	--

This action provides options for launching different components from the same action using the <dynamicfilter> element. For information on the <dynamicfilter> element and its sub-elements, see *WDK and Client Applications Development Guide*.

## Elements

The configurable element is the same as that for [renditions \(type dm\\_sysobject\), page 397](#).

## view (entitlement collaboration, type dmc\_notepage)

**Purpose** Displays a note object. A note is a content file that can be displayed directly in the browser. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\notepage\dmc\_notepage\_actions.xml

### Scope

---

entitlement	collaboration
type	dmc_notepage

---

### Parameters

---

folderPath	(Optional) Path to the folder in which the dmc_notepage object resides.
objectId	(Required) Object ID of the dmc_notepage object to view.

---

## view (type dm\_externalresult)

**Purpose** Views the content of an external search result

### Configuration File

webcomponent\config\actions\dm\_externalresult\_actions.xml

**Scope**

type	dm_externalresult
------	-------------------

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**

entryId	(Required) ID of the result entry.
queryId	(Required) ID of the query used for searching the external source.

**view (type dm\_notification)**

**Purpose**      Displays a notification

**Configuration File**

webcomponent\config\actions\dm\_notification\_actions.xml

**Scope**

type	dm_notification
------	-----------------

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**

---

objectId	(Required) ID of the notification object.
taskmanagerId	(Required) Component ID of the task manager component.

---

**view (type dm\_process)**

**Purpose** Launches the Web Workflow Manager for viewing a workflow template. The user must have read\_permit access to the object (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

**Configuration File**

webcomponent\config\actions\dm\_process\_actions.xml

**Scope**

---

type	dm_process
------	------------

---

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**

---

objectId	(Required) ID of the workflow template.
ownerName	(Optional) Name of the user who owns the object to be viewed.
setError	(Optional) Set to true to pass the precondition error message to the WebcomponentErrorService as a non-fatal message

---

## view (type dm\_query)

**Purpose** Launches the search component for viewing a query. The user must have the read\_permit access to the object (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_query\_actions.xml

### Scope

type	dm_query
------	----------

### Caller

menubar and component pages that contain actionlinklist tag

### Parameters

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system)
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
lockOwner	(Optional) User who has checked out the object (dm_query only)
navigation	(Optional) Type of navigation to perform after the action
objectId	(Required) ID of the object to be viewed.

ownerName	(Optional) Name of the object owner (dm_query only)
type	(Optional) Object type (dm_query only)

---

## view (type dm\_queued)

**Purpose** Launches the viewing application for the selected object. The user must have read\_permit access to the object (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_queued\_actions.xml

### Scope

---

type	dm_queued
------	-----------

---

### Caller

reportdetailssummarydrilldown, reportdetailssummarylist components

### Parameters

---

objectId	(Required) ID of the object to be viewed.
taskmanagerId	(Required) Component ID of the task manager component.

---

## view (type dm\_router\_task)

**Purpose** Launches the viewing application for the selected object. The user must have read\_permit access to the object (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_router\_task\_actions.xml

### Scope

type	dm_router_task
------	----------------

### Caller

inboxdrilldown, inboxlist components

### Parameters

objectId	(Required) ID of the object to be viewed.
taskmanagerId	(Required) Component ID of the task manager component.

## view (type dm\_smart\_list)

**Purpose** Launches the viewing application to run the saved search. The user must have read\_permit access to the object (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

## Configuration File

webcomponent\config\actions\dm\_smart\_list\_actions.xml

## Scope

---

type	dm_smart_list
------	---------------

---

## Caller

allsavedsearches, mysavedsearches, savedsearches components

## Parameters

---

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system).
isReference	(Optional) Boolean flag that indicates whether the object is a reference object.
lockOwner	(Optional) Name of the user who has checked out the object
navigation	(Optional) Type of navigation upon execution.
objectId	(Required) ID of the object to be viewed.
ownerName	(Optional) Name of the object owner.
query	(Optional) DQL query (for queryType = dql) or search string (for queryType = string).
queryType	(Optional) Specifies whether the query is a string value or DQL. Valid values: string   dql
type	(Optional) Object type

---



## view (type dm\_sysobject)

**Purpose** Launches the viewing application for the selected object. The user must have read\_permit access to the object (valid values: DELETE\_PERMIT | WRITE\_PERMIT | VERSION\_PERMIT | RELATE\_PERMIT | READ\_PERMIT | BROWSE\_PERMIT | NONE\_PERMIT).

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

menubar and component pages that contain actionlinklist tag: menubar, myobjectsdrilldown, myobjectslist, renditionsclassic, renditionsstreamline, doclist, searchresultsdrilldown, searchresultslist, doclist, drilldown,vdmlist, vdmlist\_streamline, managequeueinbox, processorworkloadinbox components

### Parameters

assembledFromId	(Optional) ID of the virtual document from which the assembly was assembled
contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system)
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
launchViewer	(Optional) Boolean flag that indicates whether to launch a viewing application.

nodeId	(Optional) Node ID of the virtual document node to view.
objectId	(Required) ID of the object to be viewed.
type	(Optional) Object type
vdmRootObjectId	(Optional) Object ID of the root of the virtual document.

---

## Elements

See [Dynamic component launching, page 9](#). The following additional element is defined:

---

<execution>.	Set enabled to true if your action class can process OLE compound documents
<olecompound>	

---

## view (type dm\_task)

**Purpose** Launches the task manager for viewing a task.

### Configuration File

webcomponent\config\actions\dm\_task\_actions.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

inboxdrilldown, inboxlist components

**Parameters**

objectId	(Required) dm_queue_id of the object to be viewed.
taskmanagerId	(Required) Component ID of the task manager component.

**view (type dm\_workflow)**

**Purpose** Launches the Web Workflow Manager to display the selected workflow.

**Configuration File**

webcomponent\config\actions\dm\_workflow\_actions.xml

**Scope**

type	dm_workflow
------	-------------

**Caller**

menubar and component pages that contain actionlinklist tag

**Parameters**

objectId	(Required) ID of the object to be viewed.
workflowRuntimeState	(Required) Integer that specifies whether the workflow runtime state is Dormant (0), Running (1), Terminated (2), Halted (3), or Finished (4).

## view (type dmr\_content)

**Purpose** Launches the viewing application for the selected rendition

### Configuration File

webcomponent\config\actions\dmr\_content\_actions.xml

### Scope

---

type	dmr_content
------	-------------

---

### Caller

menubar and any component that contains actionlinklist tag

### Parameters

---

contentType	(Required) File format to use when opening the object (passed to the operating system)
objectId	(Required) ID of the object to be viewed.
pageModifier	(Optional) A valid value for the page_modifier attribute of the dmr_content object. You should not pass in your own value for PDF renditions.

---

## Elements

---

<code>&lt;dynamicfilter&gt;</code>	Specifies the filter class. It contains two or more options for launching different components from the same action.
<code>&lt;dynamicfilter&gt;&lt;option&gt;</code>	Defines an option that will be launched when criteria are met. You must provide at least two options for the filter. Your evaluator class must return a value to the filter class that matches one of the criterion names in your filter definition.
<code>&lt;option&gt;&lt;criteria&gt;</code>	Contains zero or more <code>&lt;criterion&gt;</code> elements whose values must match conditions as implemented by the filter class in order for the component specified in the <code>&lt;selection&gt;</code> element to be launched. If this element is empty, the selection is the default selection.
<code>&lt;criteria&gt;&lt;criterion name='string'&gt;</code>	Defines a criterion whose value must be matched. The criterion name is evaluated by the filter or the evaluator class, which matches the value attribute against its business logic and determines which component should be launched. Criteria are matched in the order that they are listed; the last option does not need to have a criterion.
<code>&lt;option&gt;&lt;selection&gt;</code>	Specifies the component and container that will be launched when the criteria are matched
<code>&lt;selection&gt;&lt;component&gt;</code>	Specifies the component that will be launched.
<code>&lt;selection&gt;&lt;container&gt;</code>	Specifies the container that will be launched
<code>&lt;selection&gt;&lt;navigation&gt;</code>	Type of navigation upon execution. Valid values: <code>jump</code>   <code>returnjump</code>   <code>nested</code> (default).

---

For more information about dynamic component launching, see *Web Development Kit and Client Application Development Guide*.

## viewassemblies (type dm\_sysobject)

**Purpose** View all assemblies generated from a virtual document

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

menubar component

### Parameters

---

assembledFromId	(Optional) ID of the virtual document from which the assembly was assembled
folderId	(Optional) ID of folder containing the virtual document
folderPath	(Optional) Path to folder containing the virtual document
isVirtualDoc	(Optional) Boolean flag that specifies whether the object is a virtual document (required for pre-4.1 repository virtual documents)
linkCount	(Optional) Number of links (child nodes) in a document. If greater than 0, the document is a virtual document.
objectId	(Required) ID of the virtual document for which to display assemblies

---

---

## viewassembly (type dm\_sysobject)

**Purpose** View an assembly object

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

Extended by the assemblylist action definition but not called directly

### Parameters

---

assembledFromId	(Optional) ID of the virtual document from which the assembly was assembled
objectId	(Required) Object ID of the selected object.

---

## viewclipboard

**Purpose** Displays the clipboard

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

### Caller

menubar, myobjectsdrilldown, subscriptionsdrilldown, inboxdrilldown components

## viewcontent (type dm\_sysobject)

### Purpose

Opens a virtual document for viewing. The user must have at least read permit on the virtual document.

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

openvirtualdocument action execution class and clickactionprompt component class

### Parameters

---

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system)
isReference	(Optional) Boolean flag that indicates whether the object is a reference object



---

launchViewer	(Optional) Boolean flag that indicates whether to launch a viewing application.
nodeId	(Optional) Node ID of the virtual document node to view.
objectId	(Required) ID of the object to be viewed.
vdmRootObjectId	(Optional) Object ID of the root of the virtual document.

---

## Elements

See [Dynamic component launching, page 9](#). The following additional element is defined:

---

<execution>.	Set enabled to true if your action class can process OLE
<olecompound>	compound documents

---

# viewqueuemanagementclipboard

**Purpose** Displays the contents of the queue management clipboard

## Configuration File

webcomponent\config\library\queuemanagement\queue\_actions.xml

## Scope

All.

## Caller

queuelist component

## viewrecordrelationships (entitlement recordsmanager, type dm\_sysobject not dm\_folder)

**Purpose** View relationships between records. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\records\relationships\dm\_relation\_actions.xml

### Scope

---

entitlement	recordsmanager
type	dm_sysobject, not dm_folder

---

### Caller

menubar component

### Parameters

---

folderId	(Optional) ID of the folder in which the source formal record resides.
folderPath	(Optional) Path to the folder in which the source formal record resides.
objectId	(Required) Object ID of the source formal record object.

---

## viewreport (type dm\_rename\_log)

**Purpose** Displays a selected job report for a user or group rename or user change home repository job

### Configuration File

webcomponent\config\admin\renamelog\renamelog\_actions.xml

### Scope

type	dm_rename_log
------	---------------

### Caller

renamejobrequestlist, userchangehomedblist components

### Parameters

objectId	(Required) Job request ID.
oldname	(Optional) Old name of user or group

## viewvdm (type dm\_sysobject)

**Purpose** Displays a virtual document tree

### Configuration File

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

Not called

**Parameters**

---

isCurrentVDMNode	(Optional) Set to true if user has selected the current node for viewing
isVirtualDoc	(Optional) Boolean flag that specifies whether the object is a virtual document (required for pre-4.1 repository virtual documents)
linkCount	(Optional) Number of links (child nodes) in a document. If greater than 0, the document is a virtual document.
objectId	(Required) ID of the virtual document.

---

## viewvirtualdoc (type dm\_sysobject)

**Purpose**      Displays a virtual document in the objectlist view**Configuration File**

webcomponent\config\actions\dm\_sysobject\_actions.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

menubar component

**Parameters**


---

isCurrentVDMNode	(Optional) Boolean flag that specifies whether the node is the root node of the current virtual document. The menu item to remove a node is disabled if the node is the root node.
isVirtualDoc	(Optional) Boolean flag that specifies whether the object is a virtual document (required for pre-4.1 repository virtual documents)
linkCount	(Optional) Number of links (child nodes) in a document. If greater than 0, the document is a virtual document.
objectId	(Required) ID of the virtual document.

---

**workflowavailability****Purpose** Sets the user's availability to receive workflow tasks**Configuration File**

webcomponent\config\actions\generic\_actions.xml

**Scope**

All.

**Caller**

inboxdrilldown, inboxlist components

## workflowstatusdrilldown

**Purpose** Displays in a drilldown view the workflows initiated by the user and the status of the workflows

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

### Caller

inboxdrilldown component

## workflowstatuslist

**Purpose** Displays in an object list view the workflows initiated by the user and the status of the workflows

### Configuration File

webcomponent\config\actions\generic\_actions.xml

### Scope

All.

**Caller**

Not called





## Components Reference

### abortworkflow (type dm\_workflow)

**Purpose** Terminates Documentum workflows prematurely and then deletes them. This component is generally used to abort obsolete workflows.

#### Configuration File

webcomponent\config\library\workflow\abort\abortworkflow\_component.xml

#### Scope

---

type	dm_workflow
------	-------------

---

#### Caller

This component is called by the action [abortworkflow \(type dm\\_workflow\)](#), page 205 and displayed in the component [combocontainer](#), page 644.

#### Parameters

---

objectId	(Required) The ID of the workflow object to be terminated.
----------	--

---

## about

**Purpose** Displays information about the version of the application.

### Configuration File

webcomponent\config\library\about\about\_component.xml

### Scope

All.

### Caller

This component is called from an action menu item in the menubar component or by clicking on the logo in the Webtop titlebar component.

### Parameters

---

enableTools	(Optional) True to enable the display of DQL editor and API tester tools
-------------	--

---

## accessoractions

**Purpose** Displays actions for user or group permissions. The available actions are **Edit** and **Remove** for the selected accessor.

### Configuration File

webcomponent\config\library\accessoractions\accessoractions\_component.xml

**Scope**

All.

**Caller**

accessibilityaction action definition

**Parameters**


---

accessorName	(Required) Name of accessor (user or group)
aceavailableactions	(Optional) accessor action link list name
basicpermit	(Optional) The name of the basic permit for the accessor. Valid values are 1 (none), 2 (browse), 3 (read), 4 (relate), 5 (version), 6 (write), 7 (delete). Default = 3 (read)
extpermit	(Optional) The extended permission for the accessor. Valid values are EXECUTE_PROCEDURE_BIT, CHANGE_LOCATION_BIT, CHANGE_STATE_BIT, CHANGE_PERMISSIONS_BIT, TAKE_OWNERSHIP_BIT. Default = 0 (no extended permissions)
extpermit_readonly	(Optional) Set to true to disable extended permission controls
objectId	(Optional) Not used
objectName	(Optional) Accessor name
permit	(Optional) String representation of the permission that is being applied to the accessor

permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT=0DF_ EXTENDED_PERMIT = 1DF_ APPLICATION_PERMIT = 2DF_ACCESS_RESTRICTION = 3DF_EXTENDED_RESTRICTION = 4DF_APPLICATION_RESTRICTION = 5DF_REQUIRED_GROUP = 6DF_REQUIRED_GROUP_SET = 7
type	(Optional) Not used

---

## accessoractionscontainer

**Purpose** Contains the accessoractions component to display actions on accessors (**Edit** and **Remove**)

### Configuration File

webcomponent\config\library\accessoractions\accessoractionscontainer\_component.xml

### Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml

### Scope

All.

### Caller

accessibilityactions action definition

## Parameters

---

component	(Optional) Specifies the contained component
-----------	--

---

## aclacelist

**Purpose** Displays and allows editing of the current list of accessors (users and groups) and their permissions within access control lists (ACLs, known as permission sets).

## Configuration File

webcomponent\config\admin\acl\aclacelist\_component.xml

## Scope

All.

## Caller

This component is called by the action `rmaccessor` (type `dm_ace`), page 410 in the Permissions tab for `dm_acl` objects.

## Parameters

---

objectId	(Required) The ID of the ACL object to be displayed.
----------	--

---

## acldelete

**Purpose** Deletes access control lists (ACLs, also called permission sets). Not used in Webtop.

**Configuration File**

webcomponent\config\admin\acl\acldelete\_component.xml

**Scope**

All.

**Caller**

Not called

**Parameters**

---

objectId	(Required) The ID of the ACL object to be deleted.
----------	--

---

## aclinfo

**Purpose** Displays and updates the simple (single-valued) attributes of access control lists (ACLs, known as permission sets).

**Configuration File**

webcomponent\config\admin\acl\aclinfo\_component.xml

**Scope**

All.

**Caller**

This component is called by these actions: [newacl \(type dm\\_acl\), page 339, properties \(type dm\\_acl\), page 365, attributes \(type dm\\_acl\), page 231, attributes \(type dm\\_acl\), page 231.](#)

## Parameters

---

objectId	(Required) The ID of the ACL object to be displayed and updated.
----------	--

---

## Elements

---

<objecttype>	Type of ACL object. Default = dm_acl
--------------	--------------------------------------

---

# acllist

**Purpose** Displays repository access control lists (ACLs, known as permission sets) according to various criteria.

## Configuration File

webcomponent\config\admin\acl\acllist\_component.xml

## Scope

All.

## Caller

This component is jumped to from the component [aclwhereused](#), page 495.

## Elements

The configurable elements in this component are the same as those in [objectlocator](#), page 886.

# acobjectlocator

**Purpose** Locates and selects any ACL in the repository

## Configuration File

webcomponent\config\library\locator\acobjectlocator\_component.xml

## Parent Definition

persistentobjectlocator:webcomponent/config/library/locator/persistentobjectlocator\_component.xml

## Scope

All.

## Caller

permissions and extendedpermissions components

## Parameters

---

flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

For information on configurable columns, see [doclist](#), page 684.



---

<objecttype>	Base type to be located.
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document

<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## Elements

The configurable elements in this component are the same as those in [objectlocator](#), page 886.

# acobjectlocatorcontainer

**Purpose** Container component for locating and selecting any ACL in the repository

## Configuration File

webcomponent\config\library\locator\acobjectlocator\_component.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Caller

permissions and extendedpermissions components

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) The ID number of the folder in which to begin the search.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

repository	(Optional) The doctype in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## aclproperties

**Purpose** Allows the user to view the attributes of an access control list (ACL). Contains [aclinfo](#), page 486 and [aclacelist](#), page 485.

### Configuration File

webcomponent\config\admin\acl\aclproperties\_component.xml

### Parent Definition

adminpropertycontainer:webcomponent/config/admin/container/  
adminpropertycontainer\_component.xml

### Scope

All.

### Caller

This component is called by the following actions: [newacl](#) (type dm\_acl), page 339, [properties](#) (type dm\_acl), page 365, [attributes](#) (type dm\_acl), page 231.

## Parameters

component	(Required) Specifies the starting component to display (default = aclinfo). The components that should be displayed within this container must be listed as child elements of <contains>.
objectId	(Required) The ID of the ACL object for which attributes are to be displayed.

## Elements

The aclproperties container uses the value of the <objecttype> element to determine how to instantiate the ACL class. For example, the object type dm\_acl instructs it to instantiate IDfACL.

# acsaveas

**Purpose** Saves an ACL with a new name.

## Configuration File

webcomponent\config\admin\acl\acsaveas\_component.xml

## Parent Definition

aclproperties:webcomponent/config/admin/acl/aclproperties\_component.xml

## Scope

All.

## Caller

This component is not used.

## Parameters

These parameters are inherited from `aclproperties:webcomponent/config/admin/acl/aclproperties_component.xml`.

---

<code>component</code>	(Required) Specifies the starting component to display (default = <code>aclinfo</code> ). The components that should be displayed within this container must be listed as child elements of <code>&lt;contains&gt;</code> .
<code>objectId</code>	(Required) The ID of the ACL object for which attributes are to be displayed.

---

# acvalidate

**Purpose** This component is used to display the accessors for an access control list (ACL). Some features of the UI are turned on only if Trusted Content Services is enabled in the Content Server.

## Configuration File

`webcomponent\config\admin\acl\acvalidate_component.xml`

## Scope

All.

## Caller

`aclacelist` component class

## Parameters

---

<code>validation</code>	(Required) (Required) This is a background parameter passed by the server to the component.
-------------------------	---

---

# aclwhereused

**Purpose** Displays lists of all objects that use specific access control lists.

## Configuration File

webcomponent\config\admin\acl\aclwhereused\_component.xml

## Scope

All.

## Caller

This component is called by the action [locations \(type dm\\_acl\)](#), page 324.

## Parameters

---

objectId	(Required) The ID of the ACL object for which attributes are to be shown.
----------	---

---

# actionbuttonlisttest

**Purpose** Tests action button list controls

## Configuration File

wdk\config\fxtest\_component.xml

**Scope**

All.

**Caller**

Must call by URL to test

## actionbuttontest

**Purpose**

Tests action button controls

**Configuration File**

wdk\config\fxtest\_component.xml

**Scope**

All.

**Caller**

Must call by URL to test

## actiondispatcher

**Purpose**

Executes the passed action and, if necessary, displays the [login, page 840](#) component. Any additional arguments are passed through to the action.

**Configuration File**

wdk\config\actiondispatcher\_component.xml



**Scope**

All.

**Caller**

ActionDispatcherServlet

**Parameters**

---

action	(Required) The action that is called by the dispatcher
--------	--

---

## actionlinklisttest

**Purpose** Tests action link list controls**Configuration File**

wdk\config\fxtest\_component.xml

**Scope**

All.

**Caller**

Must call by URL to test

## actionlinktest

**Purpose** Tests action link controls

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

### Caller

Must call by URL to test

## actionmenuitemtest

**Purpose** Tests action menu item controls

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

### Caller

Must call by URL to test

## actionmultiselecttest

**Purpose** Tests action multiselect controls

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

### Caller

Must call by URL to test

## addaccessor

**Purpose** Adds or edits individual accessors and accessor groups, including their permissions, for inclusion in access control lists.

### Configuration File

webcomponent\config\admin\acl\addaccessor\_component.xml

### Scope

All.

### Caller

This component is called by the action [editaceaction](#) (type `dm_ace`), page 286.

## Parameters

---

accessorName	(Required) The name of the individual or group to be added.
basicpermit	(Optional) The name of the basic permit for the accessor. Valid values are 1 (none), 2 (browse), 3 (read), 4 (relate), 5 (version), 6 (write), 7 (delete). Default = 3 (read)
extpermit	(Optional) The extended permission for the accessor. Valid values are EXECUTE_PROCEDURE_BIT, CHANGE_LOCATION_BIT, CHANGE_STATE_BIT, CHANGE_PERMISSIONS_BIT, TAKE_OWNERSHIP_BIT. Default = 0 (no extended permissions)
extpermit_readonly	(Optional) Set to true to disable extended permission controls
objectName	(Optional) Accessor name passed from container
permit	(Optional) String representation of the permission that is being applied to the accessor
permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT=0DF_ EXTENDED_PERMIT = 1DF_ APPLICATION_PERMIT = 2DF_ACCESS_RESTRICTION = 3DF_EXTENDED_RESTRICTION = 4DF_APPLICATION_RESTRICTION = 5DF_REQUIRED_GROUP = 6DF_REQUIRED_GROUP_SET = 7

---

# addchildsysobjectlocator

**Purpose** Locates sysobjects to add to a virtual document. Two types of views are supported: flatlist, in which all selectable objects in the repository are shown, and hierarchical, in which the user can drill down from a list of root container (cabinets).

## Configuration File

```
webcomponent\config\library\vdm\addcomponent\addchildsysobjectlocator_
component.xml
```

## Parent Definition

```
sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator_component.xml
```

## Scope

All.

## Caller

addcomponentfileselectorcontainer and addvirtualdocumentnodefromclipboard  
component classes

## Parameters

flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

---

<containerselectable>

Boolean flag that specifies whether the container type is selectable when the container is a subtype of the objecttype.

---

## addchildsysobjectlocatorcontainer

**Purpose** Locates sysobjects to add to a virtual document. Two types of views are supported: flatlist, in which all selectable objects in the repository are shown, and hierarchical, in which the user can drill down from a list of root container (cabinets).

### Configuration File

webcomponent\config\library\vdm\addcomponent\addchildsysobjectlocatorcontainer\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

### Scope

All.

### Caller

addcomponentfileselectorcontainer and addvirtualdocumentnodefromclipboard component classes

---

## Parameters

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display a drop-down list
multiselect	(Required) Boolean. Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

# addcomponent

**Purpose** Adds a child to a virtual document

## Configuration File

webcomponent\config\library\vdm\addcomponent\addcomponent\_component.xml

## Parent Definition

checkout:webcomponent/config/library/checkout/checkout\_component.xml

## Scope

All.

## Caller

addvirtualdocumentnodefromclipboard and addvirtualdocumentnodefromfileselector  
components classes

## Parameters

---

isReference	(Optional) Boolean flag that specifies whether the object is a reference
lockOwner	(Optional) Name of owner of root node of the virtual document
nodeId	(Optional) ID of the root node of the virtual document
objectId	(Required) ID of the task.

---

# addcomponentcontainer

**Purpose** Add components to the list of items in a checkout operation

## Configuration File

webcomponent\config\library\vdms\addcomponent\addcomponentcontainer\_component.xml

## Parent Definition

checkoutcontainer:webcomponent/config/library/checkout/checkoutcontainer\_component.xml

## Scope

All.

## Caller

addvirtualdocumentnodefromclipboard and addvirtualdocumentnodefromfileselector components classes



## Parameters

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

# addcomponentfileselector

**Purpose** Dummy component that is called from the menubar component. The actual work of adding a component is done in the container component, addcomponentfileselectorcontainer.

## Configuration File

webcomponent\config\library\vdm\addcomponent\addcomponentfileselector\_component.xml

## Parent Definition

checkout:webcomponent/config/library/checkout/checkout\_component.xml

## Scope

All.

## Caller

menubar component

**Parameters**

---

isReference	(Optional) Boolean flag that specifies whether the object is a reference
lockOwner	(Optional) Name of object owner
nodeId	(Optional) ID of the root node of the virtual document
objectId	(Required) ID of the task.

---

## addcomponentfileselectorcontainer

**Purpose** Enables a user to select a file to add as a virtual document node

**Configuration File**

webcomponent\config\library\vdm\addcomponent\addcomponentfileselectorcontainer\_component.xml

**Parent Definition**

checkoutcontainer:webcomponent/config/library/checkout/checkoutcontainer\_component.xml

**Scope**

All.

**Caller**

Not called

**Parameters**

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

**addnewvirtualdocumentnode (type dm\_sysobject)**

**Purpose** Adds a new node to a virtual document, enabling the user to create the content of the node.

**Configuration File**

webcomponent\config\library\vdm\addcomponent\addnewvirtualdocumentnode\_component.xml

**Scope**

type	dm_sysobject
------	--------------

**Caller**

This component is called by the action [addnewvirtualdocumentnode \(type dm\\_sysobject\)](#), page 214.

## Parameters

---

lockOwner	(Optional) Name of object owner
nodeId	(Optional) Node ID of the parent node under which to add the new content.
objectId	(Required) Object ID of the parent object.
vdmRootObjectId	(Optional) Object ID of the root of the virtual document.

---

# addroommembercontainer

**Purpose** Container object for adding a user as a member of a room. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\rooms\addroommembercontainer\_component.xml

## Parent Definition

userorgrouplocatorcontainer:webcomponent/config/library/locator/userorgrouplocator\_component.xml

## Scope

All.

## Caller

Nested from the roommembership component class

## Parameters

arrangeselect	(Optional) Boolean: Set to true to allow selection of users in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objectId	(Required) ID of the dmc_room object.

# addroomuserorgrouplocator

**Purpose** A locator that finds users or groups in a doctype. Used in operations that add groups or users to a room.

## Configuration File

webcomponent\config\library\rooms\addroommembercontainer\_component.xml

## Parent Definition

userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml

## Scope

All.

## Parameters

objectId	(Required) Object ID of the user or group for which to search.
----------	--

# addtaskattachmentlocatorcontainer

**Purpose** Adds an attachment to a task. Contains [attachmentlocator](#), page 573.

## Configuration File

```
webcomponent\config\library\workflow\addattachment\addtaskattachmentlocatorcontainer_
component.xml
```

## Parent Definition

```
locatorcontainer:webcomponent/config/library/locator/locatorcontainer_component.xml
```

## Scope

All.

## Caller

This component is called by the action [addattachment](#) (type `dmi_package`), page 209.

## Parameters

---

formTemplateId	(Optional) ID of the form template, if the attachment is a form instance
objectId	(Required) ID of the attachment
objecttype	(Required) Attachment object type
packageName	(Required) Name of the workflow package
versionlabel	(Optional) Version label for the attachment

---

## Elements

<multidocbasesupport>	Set value to true to enable the locator to search across repositories
-----------------------	---

# addvirtualdocumentnode (type dm\_sysobject)

**Purpose** Adds a node to a virtual document.

## Configuration File

webcomponent\config\library\vdm\addcomponent\addvirtualdocumentnode\_component.xml

## Scope

type	dm_sysobject
------	--------------

## Caller

This component is called by the action [addcomponent \(type dm\\_sysobject\)](#), page 210.

## Parameters

insertAfterNodeId	(Optional) Node ID of the virtual document node after which to add the new node.
lockOwner	(Optional) Name of object owner
newNodeObjectIds	(Required) Comma-delimited string of new object IDs for the objects that are to be added to the virtual document
nodeId	(Optional) Node ID of the parent node for the new node.

objectId	(Required) Object ID of the parent node for the new node.
vdmRootObjectId	(Optional) Object ID of the root of the virtual document.

---

## addvirtualdocumentnodefromclipboard (type dm\_sysobject)

**Purpose** Adds an object from the clipboard to a virtual document

### Configuration File

webcomponent\config\library\vdm\addcomponent\addvirtualdocumentnodefromclipboard\_component.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

This component is called by the action [addcomponentfromclipboard \(type dm\\_sysobject\)](#), [page 211](#).

### Parameters

---

lockOwner	(Optional) Name of the user who owns the root node of the virtual document
nodeId	(Optional) Node ID of the object to operate on, when it is part of a virtual document.



---

objectId	(Required) ID of the object being added to the virtual document
vdmRootObjectId	(Optional) Object ID of the root of a virtual document (when operating on a child node).

---

## addvirtualdocumentnodefromfileselector (type dm\_sysobject)

**Purpose** Adds an object to a virtual document that is selected by the user. This component nests to the addchildsysobjectlocatorcontainer component to make the object selection.

### Configuration File

webcomponent\config\library\vdm\addcomponent\addvirtualdocumentnodefromfileselector\_component.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

This component is called by the action [addcomponentfromfileselector \(type dm\\_sysobject\)](#), page 212.

### Parameters

---

lockOwner	(Optional) Name of the owner of the virtual document root node
nodeId	(Optional) Node ID of the object to operate on, when it is part of a virtual document.

objectId	(Required) ID of the object to be added to the virtual document
vdmRootObjectId	(Optional) Object ID of the root of a virtual document (when operating on a child node).

---

## admindelete

**Purpose** Enables the administrator to delete an ACL

### Configuration File

webcomponent\config\admin\admindelete\admindelete\_component.xml

### Scope

All.

### Caller

Delete action for dm\_acl object

### Parameters

---

objectId	(Required) The ID of the object to be deleted.
objectname	(Required) Name of the ACL.
type	(Required) Object type of the ACL.

---

# adminpropertycontainer

**Purpose** Creates administration objects such as IDfUser, IDfGroup, IDfACL based on the object ID, or based on the type specified as the value of the element <objecttype>.<object>. When necessary, it handles the saving and reverting of these objects.

## Configuration File

webcomponent\config\admin\container\adminpropertycontainer\_component.xml

## Parent Definition

propertywizardcontainer:webcomponent/config/library/  
propertywizardcontainer/propertywizardcontainer\_component.xml

## Scope

All.

## Caller

Extended by other container definitions but not called directly

## Parameters

objectId	(Required) ID of the object for which properties are displayed
----------	--

## Elements

<objecttype>.<object>	Type of object that is instantiated by the container
-----------------------	--

# adminuserorgrouplocator

**Purpose** Locates users and groups in a repository.

## Configuration File

webcomponent\config\admin\adminuserorgrouplocator\adminuserorgrouplocator\_ component.xml

## Parent Definition

userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_ component.xml

## Scope

All.

## Caller

This component is called by the actions [getowner \(type dm\\_acl\)](#), page 306 and [ownerbrowse \(type dm\\_group\)](#), page 358 in the container [adminuserorgrouplocatorcontainer](#), page 519.

## Parameters

These parameters are inherited from userorgrouplocator:webcomponent/config/library/ locator/userorgrouplocator\_ component.xml.

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of users and groups in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype
<flatlist>	Set to true to display a flat list of all selectable objects
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.

<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)

---

<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute <code>dqiformatted='false'</code> to quote and escape a value. Use the attribute <code>casesensitive='true'</code> to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## Elements

The configurable elements are the same as those for [objectlocator](#), page 886.

# adminuserorgrouplocatorcontainer

**Purpose** Locates users and groups in a repository.

## Configuration File

webcomponent\config\admin\adminuserorgrouplocator\adminuserorgrouplocator\_component.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Caller

This container is called by the actions [getowner \(type dm\\_acl\)](#), page 306 and [ownerbrowse \(type dm\\_group\)](#), page 358.

## Parameters

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.



<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)

<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute <code>dqlformatted='false'</code> to quote and escape a value. Use the attribute <code>casesensitive='true'</code> to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## advsearch

**Purpose** WDK 5.3 advanced search allows a user to select type, type attributes, and location for search. The data dictionary supplies searchable types and attributes as well as value assistance (except conditional value assistance). Search results display columns are configured in the search column and can be selected by users in the `display_preferences` component UI.

### Configuration File

`webcomponent\config\library\search\searchex\advsearch_component.xml`

### Scope

All.

### Caller

This component is called by the actions [edit \(type `dm\_smart\_list`\), page 285](#) and [editfile \(type `dm\_smart\_list`\), page 291](#), and is displayed in the component [advsearchcontainer](#), page 525.

## Parameters

---

drilldown	(Optional) Set to true to use the drilldown view by default to display the search results. This parameter is set to false in Webtop when the user selects classic view in preferences.
folderpath	(Optional) Used to populate the radio button that displays the current location
includetypes	(Optional) Comma-delimited list of object types to display for search. Subtypes of the selected type will also be searched. Overrides the default set in the <includetypes> element.
objectId	(Optional) Specifies the object id of the saved search to be revised
query	(Optional) Full-text query string
queryId	(Optional) Internally used to identify the current search context
showwait	(Optional) Specifies whether to show the wait page for a non-DQL query. By default, the wait page is displayed.
type	(Optional) Initial search type, defaults to the first type in includetypes
usepreviousinput	(Optional) Set to true to restore previous search criteria input, if possible

---

## Elements

---

<failoverenabled>	This component supports failover. To disable failover, enter a value of false.
<includetypes>	Comma-delimited list of object types to display for search. Set the descend attribute to true to search subtypes of the selected type; set to false to search only the selected type..

---

## advsearch (version 5.2.5)

**Purpose** WDK 5.2.5 search component that searches specific repositories in all views, based on sysobject or custom attributes.

### Configuration File

webcomponent\config\library\search\advsearch\_component.xml

### Scope

---

version	5.2.5
---------	-------

---

### Caller

Must be extended; cannot be called directly.

### Parameters

---

folderpath	(Optional) The path to the folder to be searched. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder
------------	---

---

### Usage

This component is not called by its component ID. You can only call it by extending it, using the path to this component definition. If you custom WDK 5.2.x component extends this component, it will be called.

---

# advsearchcontainer

**Purpose**        Contains the WDK 5.3 search components.

## Configuration File

webcomponent\config\library\search\searchex\advsearchcontainer\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [edit \(type dm\\_smart\\_list\), page 285](#) and [editfile \(type dm\\_smart\\_list\), page 291](#), and displays the component [advsearch, page 522](#).

## Parameters

---

basetype	(Optional) The base object type used to populate the available search types
component	(Optional) Name of the component within the container
drilldown	(Optional) Set to true to use the drilldown view by default to display the search results. This parameter is set to false in Webtop when the user selects classic view in preferences.

folderpath	(Optional) Specifies the initial folder path for the search location. This value is used if the user has not set a search location preference.
objectId	(Optional) Specifies the object id of the saved search to be revised
queryId	(Optional) Internally used to identify the current search context
showwait	(Optional) Not used
type	(Optional) Initial search type, defaults to the basetype
usepreviousinput	(Optional) Set to true to restore previous search criteria input, if possible

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
---------------------------	--

---

# advsearchcontainer (version 5.2.5)

**Purpose** WDK 5.2.5 component that contains the search components advsearch and savedsearches. This component cannot be addressed directly, but it is present to support custom components that extend it.

## Configuration File

webcomponent\config\library\search\advsearchcontainer\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

**Scope**

version	5.2.5
---------	-------

**Caller**

Must be extended; cannot be called directly.

**Parameters**

component	(Optional) Name of the component within the container to be initially displayed
-----------	---

**Elements**

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default:5.2.5
---------------------------	--

## aliasobjectlocator

**Purpose**      Selects any alias set in the repository

**Configuration File**

webcomponent\config\library\locator\aliasobjectlocator\_component.xml

**Parent Definition**

persistentobjectlocator:webcomponent/config/library/locator/persistentobjectlocator\_component.xml

### Scope

All.

### Caller

aliasbrowse action

### Parameters

---

multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

### Elements

For information on configurable columns, see [doclist, page 684](#). Inherits the configurable attributes of [persistentobjectlocator, page 891](#).

## aliasobjectlocatorcontainer

**Purpose**        Selects any alias set in the repository.

### Configuration File

webcomponent\config\library\locator\aliasobjectlocator\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

### Scope

All.



## Caller

aliasetbrowse action

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) The ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) The docbase that is to be searched
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

# alldocumentlocator

**Purpose** Locates dm\_objects in a repository. It allows users to navigate from the root cabinets to locate objects. Users can filter objects by the content format.

## Configuration File

webcomponent\config\library\locator\documentlocators\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Caller

Extended by other components but not called directly

## Parameters

These parameters are inherited from `sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator_component.xml`.

---

<code>contentType</code>	(Optional) Content type for filetype filter
<code>flatlist</code>	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<code>folderId</code>	(Optional) ID of folder in which to begin the search
<code>multiselect</code>	(Optional) Boolean: Set to true to support multiple object selection.
<code>objecttype</code>	(Optional) Type of the selected object

---

## Elements

---

<code>&lt;objecttype&gt;</code>	Base type to be located
<code>&lt;containerselectable&gt;</code>	Set to true to specify that the container type is selectable when the container is a subtype of the <code>objecttype</code>
<code>&lt;flatlist&gt;</code>	Set to true to display a flat list of all selectable objects.
<code>&lt;views&gt;</code>	Contains <code>&lt;view&gt;</code> elements
<code>&lt;view&gt;</code>	View element configures root (cabinet view), container, and flatlist views. The <code>applyto</code> attribute must specify one or more of the views in a comma-separated list, for example, <code>applyto='root,container,flatlist'</code> .

<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter

<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## allfolderlocator

**Purpose** Locates dm\_folder objects in a repository and displays them in a hierarchical view. Users can navigate from the root cabinets to locate an object.

### Configuration File

webcomponent\config\library\locator\folderlocators\_component.xml

### Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

### Scope

All.

## Caller

5.2.5 advsearch component, newdocument, transformationnewobjectdef, historicalprocessreport components

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

objecttype	Type of the selected object
flatlist	Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	Boolean: Set to true to support multiple object selection.
selectedobjectids	Comma-delimited or semicolon-delimited list of object IDs that are initially selected

# alllifecyclelocator

**Purpose** Locates dm\_policy objects in a repository.

## Configuration File

webcomponent\config\library\locator\lifecyclelocators\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Caller

applylifecycle container

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

For information on configurable columns, see [doclist](#), page 684.

---

<objecttype>	Base type to be located
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype
<flatlist>	Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document

<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## allsavedsearches

**Purpose** Lists all dm\_smartlist objects for which the user has access. The user can revise (if permitted), execute, view properties, and delete (if permitted).



**Configuration File**

webcomponent\config\library\search\searchex\allsavedsearches\_component.xml

**Scope**

All.

**Caller**

advsearchcontainer

## alluserwfemplatelocator

**Purpose** Lists all repository workflow templates that are not system-generated.

**Configuration File**

webcomponent\config\library\workflow\userwfemplatelocator\_component.xml

**Parent Definition**

allwfemplatelocator:webcomponent/config/library/locator/wfemplatelocators\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [startworkflowfromdoc](#) (type `dm_sysobject`), page [427](#) and displayed in the component [startworkflowtemplatelocatorcontainer](#), page [1020](#).

## Parameters

These parameters are inherited from `sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator_component.xml`.

<code>contentType</code>	(Optional) Content type for filetype filter
<code>flatlist</code>	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<code>folderId</code>	(Optional) ID of folder in which to begin the search
<code>multiselect</code>	(Optional) Boolean: Set to true to support multiple object selection.
<code>objecttype</code>	(Optional) Type of the selected object

## Elements

<code>&lt;objecttype&gt;</code>	Base type to be located
<code>&lt;containerselectable&gt;</code>	Set to true to specify that the container type is selectable when the container is a subtype of the <code>objecttype</code>
<code>&lt;flatlist&gt;</code>	Set to true to display a flat list of all selectable objects.
<code>&lt;views&gt;</code>	Contains <code>&lt;view&gt;</code> elements
<code>&lt;view&gt;</code>	View element configures root (cabinet view), container, and flatlist views. The <code>applyto</code> attribute must specify one or more of the views in a comma-separated list, for example, <code>applyto='root,container,flatlist'</code> .
<code>&lt;queryfiltersets&gt;</code>	Contains <code>&lt;queryfilterset&gt;</code> elements that present a drop-down list that will be visible if there is more than one filter ( <code>&lt;queryfilterset&gt;</code> ) defined.
<code>&lt;queryfilterset&gt;</code>	Contains a set of queries contained in <code>&lt;queryfilter&gt;</code> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.

<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)

<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute <code>dqlformatted='false'</code> to quote and escape a value. Use the attribute <code>casesensitive='true'</code> to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<columns>	Contains <column> elements to display or hide attributes
<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'>. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

---

## allwftemplatelocator

**Purpose** Locates dm\_process objects (workflow templates) in a repository

### Configuration File

webcomponent\config\library\locator\wftemplatelocators\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Caller

This component is called by the component [wftemplatelocatorcontainer](#), page 1132.

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

For information on configurable columns, see [doclist](#), page 684.

<objecttype>	Base type to be located
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements

<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.

---

<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<columns>	Contains <column> elements to display or hide attributes
<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

---

# allworkqueuecategorylocator

**Purpose** Lists all defined work queue categories.

## Configuration File

webcomponent\config\library\queuemanagement\workqueuecategorylocators\_  
component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Caller

This component is called by the component [workqueuecategorylocatorcontainer](#), page 1151.

## Parameters

---

arrangeselect	(Optional) Not used.
multiselect	(Optional) Boolean: Set to true to support multiple selection.
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---



## Elements

---

<objecttype>	Sets the user preference for the object type to be displayed. The default is <code>dmc_workqueue_category</code> .
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<privatecabinetvisible>	Set to true to allow the user to see private cabinets owned by other users.
<allversionsvisible>	Set to true to display all versions of the objects in the list; set to false to display only the current versions.
<view>	View element configures root (cabinet view), container, and flatlist views. The <code>applyto</code> attribute must specify one or more of the views in a comma-separated list, for example, <code>applyto='root,container,flatlist'</code> .
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: <code>select &lt;includetypes&gt; from &lt;containertypes&gt; not &lt;excludetypes&gt; where &lt;attribute&gt; &lt;predicate&gt; &lt;value&gt; &lt;and&gt; &lt;attribute&gt; &lt;predicate&gt; &lt;value&gt; ...</code>
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as <code>dm_cabinet</code> , <code>dm_folder</code> .

<queryfilter>.<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<queryfilter>.<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.

---

## api

**Purpose** Tests and executes Content Server API calls.

### Configuration File

webcomponent\config\admin\utilities\api\_component.xml

### Scope

All.

# appintgcontroller

**Purpose** This component launches an action or component and sets up a return listener to load a success or failure page.

## Configuration File

webcomponent\config\library\appintgcontroller\appintgcontroller\_component.xml

## Scope

All.

## Caller

This component is called by the actions [appintgnewfromtemplate, page 226](#), [appintgopenfrom, page 226](#), and [appintgsaveas, page 227](#).

## Parameters

---

disconnect	(Optional) Boolean: Set to true to disconnect all sessions before execution
dispatchargs	(Optional) Argument list to be passed to the dispatch item.
dispatchitem	(Required) (Optional) Item to be dispatched.
dispatchtype	(Required) (Optional) The action or component.
docbase	(Optional) ID of the Docbase to connect to.
locale	(Optional) ID of the Locale to use (default is "en_US").
theme	(Optional) ID of the Theme to use (default is "windowsxpsilver").
view	(Optional) ID of the view to use(Classic or Streamline view).

---

# appintgcontrollerlogin

**Purpose** This component is called when a user attempts to open a repository file from a Windows application but does not have an active repository session. It extends [appintgcontroller](#), page 547 to fire the loginsuccess event.

## Configuration File

webcomponent\config\library\appintgcontroller\appintgcontroller\_component.xml

## Parent Definition

appintgcontroller:webcomponent/config/library/appintgcontroller/appintgcontroller\_component.xml

## Scope

All.

## Caller

This component is called by the [appintgcontroller](#), page 547 component as necessary to complete a transaction initiated by the actions [appintgnewfromtemplate](#), page 226, [appintgopenfrom](#), page 226, and [appintgsaveas](#), page 227.

## Parameters

These parameters are inherited from appintgcontroller:webcomponent/config/library/appintgcontroller/appintgcontroller\_component.xml.

---

disconnect	(Optional) Boolean: Set to true to disconnect all sessions before execution
dispatchargs	(Optional) Argument list to pass to the dispatched item.
dispatchitem	(Required) Item to be dispatched.
dispatchtype	(Required) Action or component.

---

docbase	(Optional) ID of the Docbase to connect to.
locale	(Optional) ID of the Locale to use (default is "en_US").
theme	(Optional) ID of the Theme to use (default is "windowsxpsilver").
view	(Optional) ID of the view to use(Classic or Streamline view).

---

## appintghelp

**Purpose** This component is used to display help for Application Connector components.

### Configuration File

webcomponent\config\library\appintghelp\appintghelp\_component.xml

### Scope

All.

### Parameters

---

helpcontextid	(Optional) Used by the WDK framework to launch a URL to a help file for Application Connectors
---------------	--

---

## appintgnewfromtemplate

**Purpose** This component is used to create a new repository object from a template. This component is called from within a Microsoft Windows application.

### Configuration File

webcomponent\config\library\appintgnewfromtemplate\appintgnewfromtemplate\_component.xml

### Scope

All.

### Caller

This component is called by the action appintgnewfromtemplate.

### Parameters

---

contentType	(Optional) Content type for filetype filter.
-------------	--

---

## appintgopen

### Purpose

This component is used to download content for view or edit a file when opened from within a Windows application. It verifies a user's permissions, presents a view/edit prompt if necessary, then performs the view or checkout action as necessary.

### Configuration File

webcomponent\config\library\appintgopenfrom\appintgopen\_component.xml

### Scope

All.

### Caller

The component is called by the action appintgopenfrom.

## Parameters

launchViewer	(Optional) Boolean: When true, this component will load the finish page when called from Search, Versions, Locations, and Renditions.
objectId	(Required) ID of object to view
openDocument	(Optional) Set to true to open the document

## Elements

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
---------------	--

# appintgopenfrom

**Purpose** This is the base component used to display locator containers when a file is opened from within a Windows application.

## Configuration File

webcomponent\config\library\appintgopenfrom\appintgopenfrom\_component.xml

## Scope

All.

## Caller

The component is called by the action [appintgopenfrom](#), page 226.

## Parameters

---

contentType	(Optional) Content type for filetype filter
folderId	(Optional) ID of folder in which to begin navigation
launchViewer	(Optional) Boolean flag that indicates whether to launch a viewing application.
locatorcomponent	(Optional) ID of the locator component displayed

---

## Elements

For information on configurable columns, see [doclist, page 684](#).

# appintgopenfromcabinetslocator

**Purpose** This component is used to open a file from within a Windows application when the users is navigating the repository via cabinets.

## Configuration File

```
webcomponent\config\library\appintgopenfrom\appintgopenfromcabinetslocator_
component.xml
```

## Parent Definition

```
sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator_component.xml
```

## Scope

All.



## Caller

This component is called by the action [appintgopenfrom, page 226](#) and displays with the components [appintgopenfromlocatorcontainer, page 554](#) and [appintgopenfrom, page 551](#).

## Parameters

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to open

## Elements

For information on configurable columns, see [doclist, page 684](#).

# appintgopenfromcategorieslocator

## Purpose

This component is used to open files from a Windows application when the user is navigating the repository via categories.

## Configuration File

webcomponent\config\library\appintgopenfrom\appintgopenfromcategorieslocator\_component.xml

## Parent Definition

categorylocator:webcomponent/config/library/locator/categorylocator\_component.xml

## Scope

All.

## Caller

This component is called by the action [appintgopenfrom](#), page 226 and displayed with the components [appintgopenfromlocatorcontainer](#), page 554 and [appintgopenfrom](#), page 551.

## Parameters

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of file selected

---

## Elements

For information on configurable columns, see [doclist](#), page 684.

# appintgopenfromlocatorcontainer

## Purpose

This component is used to open files from the repository when called from within Windows applications. It displays a specific component for one of several different contexts.

## Configuration File

```
webcomponent\config\library\appintgopenfrom\appintgopenfromlocatorcontainer_
component.xml
```

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [appintgopenfrom](#), [page 226](#) and displays the components [appintgopenfrom](#), [page 551](#), [appintgopenfromcabinetslocator](#), [page 552](#), [appintgopenfromcategorieslocator](#), [page 553](#), [appintgopenfrommyfileslocator](#), [page 556](#), [appintgopenfromrecentfileslocator](#), [page 557](#), and [appintgopenfromsubscriptionslocator](#), [page 558](#).

## Parameters

component	(Optional) ID of the component to be displayed
contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of object to select
repository	(Optional) ID of the repository to be searched (Optional) ID of the repository to be searched (Optional) ID of the repository to be searched (Optional) ID of the repository to be searched (Optional) ID of the repository to be searched (Optional) ID of the repository to be searched (Optional) ID of the repository to be searched

repository	(Optional) ID of the repository to be searched ID of the repository to be searched (Optional) ID of the repository to be searched ID of the repository to be searched (Optional) ID of the repository to be searched ID of the repository to be searched (Optional) ID of the repository to be searched ID of the repository to be searched
selectedobjectids	(Optional) Comma- or semicolon-separated list of objects to be selected when the component is displayed

---

## Elements

---

<multidocbasesupport>	Set to true to enable locator for multiple repositories
-----------------------	---

---

# appintgopenfrommyfileslocator

**Purpose** This component is used to open a file from a Windows application when the user is navigating the repository via *my files* .

## Configuration File

webcomponent\config\library\appintgopenfrom\appintgopenfrommyfileslocator\_component.xml

## Parent Definition

myobjectlocator:webcomponent/config/library/locator/myobjectlocator\_component.xml

## Scope

All.

## Caller

This component is called by the action [appintgopenfrom](#), page 226 and displayed with the components [appintgopenfromlocatorcontainer](#), page 554 and [appintgopenfrom](#), page 551.

## Parameters

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to be selected

## Elements

For information on configurable columns, see [doclist](#), page 684.

# appintgopenfromrecentfileslocator

## Purpose

This component is used to open files from within a Windows application when the user is navigating the repository via *recent files*.

## Configuration File

webcomponent\config\library\appintgopenfrom\appintgopenfromrecentfileslocator\_component.xml

## Parent Definition

recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_component.xml

## Scope

All.

## Caller

This component is called by the action [appintgopenfrom](#), page 226 and displayed with the components [appintgopenfromlocatorcontainer](#), page 554 and [appintgopenfrom](#), page 551.

## Parameters

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to be selected

---

## Elements

For information on configurable columns, see [doclist](#), page 684.

# appintgopenfromsubscriptionslocator

## Purpose

This component is used to open a file from within a Windows application when the user is navigating the repository via *subscriptions*.

## Configuration File

```
webcomponent\config\library\appintgopenfrom\appintgopenfromsubscriptionslocator_
component.xml
```

## Parent Definition

subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_  
component.xml

## Scope

All.

## Caller

This component is called by the action [appintgopenfrom](#), page 226 and displayed with the components [appintgopenfromlocatorcontainer](#), page 554 and [appintgopenfrom](#), page 551.

## Parameters

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to be selected

## Elements

For information on configurable columns, see [doclist](#), page 684.

# appintgsaveascabinets

## Purpose

This component is used by the *save as* command when the user is navigating the repository via cabinets. This component is displayed in the container `appintgsaveascontainer`.

## Configuration File

webcomponent\config\library\appintgsaveas\appintgsaveascabinets\_component.xml

## Parent Definition

appintgopenfromcabinetslocator:webcomponent/config/library/appintgopenfrom/  
appintgopenfromcabinetslocator\_component.xml

## Scope

All.

## Caller

This component is called by the action [appintgsaveas](#), page 227. This component is displayed in the container [appintgsaveascontainer](#), page 561.

## Parameters

These parameters are inherited from appintgopenfromcabinetslocator:webcomponent/  
config/library/appintgopenfrom/appintgopenfromcabinetslocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to be saved

---

# appintgsaveascategories

## Purpose

This component is used by the *save as* command when the user is navigating the repository via categories. This component is displayed in the container [appintgsaveascontainer](#).



## Configuration File

webcomponent\config\library\appintgsaveas\appintgsaveascategories\_component.xml

## Parent Definition

appintgopenfromcategorieslocator:webcomponent/config/library/appintgopenfrom/appintgopenfromcategorieslocator\_component.xml

## Scope

All.

## Caller

This component is called by the action [appintgsaveas](#), page 227. This component is displayed in the container [appintgsaveascontainer](#), page 561.

## Parameters

These parameters are inherited from appintgopenfromcategorieslocator:webcomponent/config/library/appintgopenfrom/appintgopenfromcategorieslocator\_component.xml.

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to be saved

# appintgsaveascontainer

## Purpose

This component is used to display components used to perform save as transactions in integrated applications.

## Configuration File

webcomponent\config\library\appintgsaveas\appintgsaveascontainer\_component.xml

## Parent Definition

appintgopenfromlocatorcontainer:webcomponent/config/library/appintgopenfrom/  
appintgopenfromlocatorcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action appintgsaveas. This container displays the components [appintgsaveascabinets](#), [page 559](#), [appintgsaveascategories](#), [page 560](#), [appintgsaveasmyfiles](#), [page 563](#), and [appintgsaveassubscriptions](#), [page 565](#).

## Parameters

---

component	(Optional) The component to be displayed
contentType	(Optional) Content type for filetype filter
filenamewithpath	(Required) Name and path of the file to be checked in
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
folderId	(Optional) ID of the folder in which to save the object
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to be created

repository	(Optional) ID of the repository in which to save the objectID of the repository in which to save the object(Optional) ID of the repository in which to save the objectID of the repository in which to save the object(Optional) ID of the repository in which to save the objectID of the repository in which to save the object(Optional) ID of the repository in which to save the objectID of the repository in which to save the object
repository	(Optional) ID of the repository in which to save the objectID of the repository in which to save the object(Optional) ID of the repository in which to save the objectID of the repository in which to save the object(Optional) ID of the repository in which to save the objectID of the repository in which to save the object(Optional) ID of the repository in which to save the objectID of the repository in which to save the object
selectedobjectids	(Optional) Comma- or semicolon-delimited list of objects to be selected when the component is displayed

---

## appintgsaveasmyfiles

**Purpose** This component is used to perform the *save as* command when the user is navigating the repository via the *myfiles* file list.

### Configuration File

webcomponent\config\library\appintgsaveas\appintgsaveasmyfiles\_component.xml

## Parent Definition

appintgopenfrommyfileslocator:webcomponent/config/library/appintgopenfrom/  
appintgopenfrommyfileslocator\_component.xml

## Scope

All.

## Caller

This component is called by the action [appintgsaveas](#), page 227. This component is displayed in the container [appintgsaveascontainer](#), page 561.

## Parameters

These parameters are inherited from appintgopenfrommyfileslocator:webcomponent/  
config/library/appintgopenfrom/appintgopenfrommyfileslocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to be saved

---

# appintgsaveasrecentfiles

## Purpose

This component is used to perform the *save as* command when the user is navigating the repository via a list of recently viewed or edited files.

## Configuration File

webcomponent\config\library\appintgsaveas\appintgsaveasrecentfiles\_component.  
xml

## Parent Definition

appintgopenfromrecentfileslocator:webcomponent/config/library/appintgopenfrom/  
appintgopenfromrecentfileslocator\_component.xml

## Scope

All.

## Caller

This component is called by the action [appintgsaveas](#), page 227. This component is displayed in the container [appintgsaveascontainer](#), page 561.

## Parameters

These parameters are inherited from appintgopenfromrecentfileslocator:webcomponent/  
config/library/appintgopenfrom/appintgopenfromrecentfileslocator\_component.xml.

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to be saved

# appintgsaveassubscriptions

## Purpose

This component is used to perform the *save as* command when the user is navigating the repository via a list of *subscriptions*.

## Configuration File

webcomponent\config\library\appintgsaveas\appintgsaveassubscriptions\_  
component.xml

## Parent Definition

appintgopenfromsubscriptionslocator:webcomponent/config/library/appintgopenfrom/  
appintgopenfromsubscriptionslocator\_component.xml

## Scope

All.

## Caller

This component is called by the action [appintgsaveas](#), page 227. This component is displayed in the container [appintgsaveascontainer](#), page 561.

## Parameters

These parameters are inherited from  
appintgopenfromsubscriptionslocator:webcomponent/config/  
library/appintgopenfrom/appintgopenfromsubscriptionslocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of object to be saved

---

# applylifecycle (type dm\_sysobject)

## Purpose

Applies a lifecycle to the selected object. It displays the available lifecycles in the system. Contains [applylifecycleall](#) (type dm\_sysobject), page 567.

## Configuration File

webcomponent\config\library\applylifecycle\applylifecyclecontainer\_component.xml

**Parent Definition**

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

**Scope**


---

type	dm_sysobject
------	--------------

---

**Caller**

This component is called by the action [applylifecycle \(type dm\\_sysobject\), page 228](#) and displays the component [applylifecycleall \(type dm\\_sysobject\), page 567](#).

**Parameters**


---

component	(Required) Name of the component within the container
objectId	(Required) ID of the object to which a lifecycle shall be applied

---

**Elements**

This component inherits the configurable elements of [propertysheetcontainer, page 914](#).

**applylifecycleall (type dm\_sysobject)**

**Purpose** Applies a lifecycle to all selected objects.

**Configuration File**

webcomponent\config\library\applylifecycle\applylifecycle\_all\_component.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

This component is called by the action [applylifecycle \(type dm\\_sysobject\)](#), page 228 and displayed in the component [applylifecycle \(type dm\\_sysobject\)](#), page 566.

**Parameters**

---

objectId	(Required) ID of the object to which the lifecycle shall be applied
----------	---

---

## assemblylist

**Purpose**

Displays the structure of a virtual document snapshot. The user can navigate to the different parts of the virtual document snapshot by clicking on the breadcrumb or the child node. This component has an extended version in Webtop called assemblylist.

**Configuration File**

webcomponent\config\navigation\vdm\assemblylist\_component.xml

**Parent Definition**

vdmlist:webcomponent/config/navigation/vdm/vdmlist\_component.xml

**Scope**

All.



## Caller

This component is called by the action [viewassembly](#) (type `dm_sysobject`), page 471.

## Parameters

These parameters are inherited from `vdmlist:webcomponent/config/navigation/vdm/vdmlist_component.xml`.

<code>nodeId</code>	(Optional) ID of the node in a virtual document.
<code>objectId</code>	(Required) ID of the virtual document.
<code>pathInfo</code>	(Optional) Path to be displayed in the breadcrumb.
<code>selectedNodeIds</code>	(Optional) Comma-separated list of <code>nodeIds</code> that should be initially selected when the <code>vdmlist</code> component is first displayed.

## Elements

<code>&lt;showfilters&gt;</code>	Set to true to show the <code>objectfilters</code> drop-down control
<code>&lt;objectfilters&gt;</code>	Contains filters that define which objects should be shown in the objects selection list.
<code>&lt;objectfilter&gt;</code>	Specifies a filter for the items that are displayed. Contains <code>&lt;label&gt;</code> , <code>&lt;showfolders&gt;</code> , <code>&lt;type&gt;</code> , <code>&lt;showallversions&gt;</code>
<code>&lt;label&gt;</code>	Displays a label for the filter such as Folders or All Can contain a string or <code>&lt;nlsid&gt;</code> .
<code>&lt;objectfilter&gt;.&lt;showfolders&gt;   &lt;type&gt;</code>	To show folders only, set <code>&lt;showfolders&gt;</code> to true and <code>&lt;type&gt;</code> to null (no value). To show objects only, set <code>&lt;showfolders&gt;</code> to false and <code>&lt;type&gt;</code> to <code>dm_sysobject</code> . To show all, set <code>&lt;showfolders&gt;</code> to true and <code>&lt;type&gt;</code> to <code>dm_sysobject</code> . <code>&lt;type&gt;</code> can take any value that is a valid Documentum type.

<code>&lt;objectfilter&gt;.&lt;showallversions&gt;</code>	To add a versions column, set <code>&lt;showallversions&gt;</code> to true.
<code>&lt;columns&gt;.</code> <code>&lt;loadinvisibleattribute&gt;</code>	Uncomment this element and set to true to get invisible attribute values for use in your component. The invisible attributes can then be passed by configuring a column in the <code>&lt;columns&gt;</code> element. See for details.
<code>&lt;columns&gt;&lt;column&gt;</code> <code>&lt;column&gt;&lt;attribute&gt;</code>	Specifies columns to show or hide <code>&lt;attribute&gt;</code> sets the attribute to be displayed in the column.
<code>&lt;attribute&gt;</code> <code>&lt;label&gt;</code>	The value of <code>&lt;label&gt;</code> sets a label for the column.
<code>&lt;column&gt;&lt;attribute&gt;</code> <code>&lt;visible&gt;</code>	Set <code>&lt;visible&gt;</code> to true to show the column.

---

## assemblyliststreamline

**Purpose** Displays a list of virtual document snapshots in drilldown (streamline) view.

### Configuration File

webcomponent\config\navigation\vdm\assemblyliststreamline\_component.xml

### Parent Definition

vdmliststreamline:webcomponent/config/navigation/vdm/vdmliststreamline\_component.xml

### Scope

All.

## Caller

This component is called by the action [viewassemblies](#) (type `dm_sysobject`), page 470.

## Parameters

These parameters are inherited from `vdmliststreamline:webcomponent/config/navigation/vdm/vdmliststreamline_component.xml`.

<code>home_component</code>	(Optional) In Portal application, specifies portal component that calls this component. Used to construct a breadcrumb back to the portal.
<code>home_component_args</code>	(Optional) In Portal application, specifies arguments that are passed to the portal component that calls this component when the user clicks the breadcrumb to go back.
<code>nodeId</code>	(Optional) ID of the node in a virtual document.
<code>objectId</code>	(Required) ID of the virtual document.
<code>pathInfo</code>	(Optional) Path to be displayed in the breadcrumb.

# assignqueuedtask

**Purpose** Assigns a task from a work queue to a processor.

## Configuration File

`webcomponent\config\library\queuemanagement\assignqueuedtask_component.xml`

## Scope

All.

## Caller

This component is called by the actions [assign\\_queued\\_task \(type dm\\_task\), page 230](#) and [reassign\\_queued\\_task \(type dm\\_task\), page 376](#) and displayed in the component [assignqueuedtaskcontainer, page 572](#). It can also be called by the action [move\\_queued\\_task \(type dm\\_task\), page 336](#) and displayed in the component [movetoqueuecontainer, page 847](#).

## Parameters

---

objectId	(Required) ID of the task to assign
performer	(Required) (Required) ID of the user to whom the task will be assigned
queueName	(Required) Name of the work queue

---

# assignqueuedtaskcontainer

**Purpose** Container object for assigning tasks from a work queue to a processor. Contains the component [assignqueuedtask, page 571](#).

## Configuration File

webcomponent\config\library\queuemanagement\assignqueuedtaskcontainer\_component.xml

## Parent Definition

multiargdialogcontainer:webcomponent/config/library/multiargdialogcontainer/multiargdialogcontainer\_component.xml

## Scope

All.

**Caller**

This component is called by the action [assign\\_queued\\_task](#) (type `dm_task`), page 230 and displays the component [assignqueuedtask](#), page 571.

**Parameters**

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## attachmentlocator

**Purpose** Locates object attachments, based on their type and required version label or labels.

**Configuration File**

`webcomponent\config\library\workflow\attachmentlocator_component.xml`

**Parent Definition**

`sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator_component.xml`

**Scope**

All.

**Caller**

This component is called by the component [attachmentlocatorcontainer](#), page 574.

## Parameters

---

formTemplateId	(Optional) ID for the XForm template
objecttype	(Optional) Type of object to be located
versionlabel	(Optional) Version label on attachment, for example, CURRENT

---

## Elements

---

<objecttype>	Base type to be located
<containsselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

---

# attachmentlocatorcontainer

**Purpose** Locates object attachments, based on their type and required version label or labels.

## Configuration File

webcomponent\config\library\workflow\attachmentlocatorcontainer\_component.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Parameters

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
formTemplateId	(Optional) ID of the form template, if the attachment is a form instance
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected
versionlabel	(Optional) Version label on attachment; for example, CURRENT

## Elements

<multidocbasesupport>	Set value to true to enable the locator to search across repositories
-----------------------	---

# attachmentmyobjectlocator

**Purpose** Selects attachment objects from the user's objects.

## Configuration File

webcomponent\config\library\workflow\attachmentmyobjectlocator\_component.xml

## Parent Definition

myobjectlocator:webcomponent/config/library/locator/myobjectlocator\_component.xml

## Scope

All.

## Caller

This component is called by the component [attachmentlocatorcontainer](#), page 574.

## Parameters

---

formTemplateId	(Optional) ID for the XForm template
objecttype	(Optional) Type of object to be located
versionlabel	(Optional) Version label on attachment, for example, CURRENT

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.



<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)

<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

---

## attachmentsubscriptionlocator

**Purpose** Selects attachment objects from the user's subscriptions list.

### Configuration File

webcomponent\config\library\workflow\attachmentsubscriptionlocator\_component.xml

### Parent Definition

subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml

### Scope

All.

## Caller

This component is called by the component [attachmentlocatorcontainer](#), page 574.

## Parameters

formTemplateId	(Optional) ID for the XForm template
objecttype	(Optional) Type of object to be located
versionlabel	(Optional) Version label on attachment, for example, CURRENT

## Elements

<objecttype>	Base type to be located
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.

<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)

---

<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

---

## attributes (type dm\_category)

**Purpose** The attributes component displays the primary attributes of the selected object and includes a link to expand the attributes list or to display all attributes. The attributes component is scoped for dm\_sysobject, dm\_document, and dm\_folder, with three different JSP pages to display attributes.

### Configuration File

webcomponent\config\library\properties\dm\_category\_properties\_component.xml

### Parent Definition

attributes:webcomponent/config/library/attributes/attributes\_dm\_folder\_component.xml

### Scope

---

type	dm_category
------	-------------

---

## Caller

This component is called by the action [attributes \(type dmc\\_workqueue\\_category\)](#), page 236 and displayed in the container component [properties](#), page 908.

## Parameters

These parameters are inherited from `attributes:webcomponent/config/library/attributes/attributes_dm_sysobject_component.xml`.

---

<code>enableShowAll</code>	(Optional) Boolean: Displays the 'Show all attributes' link (true) or hides it (false). The default depends on the user's role: true for administrators, false for all non-administrator users.
<code>objectId</code>	(Required) (Required) The ID of the object for which attributes are to be shown.
<code>readOnly</code>	(Optional) Boolean: Displays the attributes as read-only (true) or in edit mode (false). This value does not override the state of the attributes in the repository. The value of this parameter does override individual readonly attributes that are set to false on the doabase attribute controls in the UI.

---

## Elements

By default, the attributes component is configured to display a set of attributes for doabase documents, folders, and system objects (sysobjects), depending on the selected object type. The component can be extended to present a custom UI for any object type.

The following table describes the elements that you can customize in the attributes component definition for `dm_sysobjects`. In the scopes of `dm_folder` and `dm_document`, this component has only the `<showpagesastabs>` configurable element.

---

<code>&lt;showpagesastabs&gt;</code>	Set to true to display attribute categories as tabs.
<code>&lt;filter&gt;</code>	Contains <code>&lt;enableShowAll&gt;</code> . Determines the criteria for activating the <code>enableShowAll</code> functionality. In the default configuration, the filter attribute "role" is set to "administrator". As a result, only administrators are able to view the extended set of attributes. Applies all types.
<code>&lt;filter&gt;.&lt;enableShowAll&gt;</code>	Boolean. True to display the 'Show all attributes' link within the filter. Applies to all types.
<code>&lt;readOnly&gt;</code>	Boolean. True to display the attributes as read-only. Applies to all types. This element does the same operation as the <code>readOnly</code> parameter. The <code>readOnly</code> parameter takes precedence.

---

## Usage

You can configure the layout of the attributes display using the `docbaseattributelist` control. See the control description for details. Each category of attributes that is defined in the list layout Configuration file can be shown as tabs or inline in the properties UI. Set the value of `showpagesastabs` in the Configuration file to true to display category tabs.

The **Show All** attributes functionality is provided by an alternative layout to the base attributes layout. The layout is switched when the user clicks a link. The link is displayed if the `enableshowall` parameter is set to true.

### Using filters

You can configure the attributes definition to display a different UI based on the user role. The following example allows administrators to see all attributes and allows the attributes to be editable if they are editable in the repository:

```
<filter role='administrator'>
  <enableShowAll>true</enableShowAll>
</filter>
<readOnly>>false</readOnly>
```

### How Attributes are Saved

The OK button in the properties component has an event handler in the parent class `DialogContainer`, which calls `canCommitChanges()` and `onCommitChanges()`. The latter method is implemented in the `Properties` class to save the object.

## attributes (type dm\_document)

**Purpose** Displays the primary attributes of the selected document and includes a link to expand the attributes list or to display all attributes. The attributes component is designed to run within the container component [properties, page 908](#).

### Configuration File

webcomponent\config\library\attributes\attributes\_dm\_document\_component.xml

### Parent Definition

attributes:webcomponent/config/library/attributes/attributes\_dm\_sysobject\_component.xml

### Scope

---

type	dm_document
------	-------------

---

### Caller

This component is called by the container component [properties, page 908](#).

### Parameters

These parameters are inherited from attributes:webcomponent/config/library/attributes/attributes\_dm\_sysobject\_component.xml.



---

enableShowAll	(Optional) Boolean: Displays the 'Show all attributes' link (true) or hides it (false). The default depends on the user's role: true for administrators, false for all non-administrator users.
objectId	(Required) (Required) The ID of the object for which attributes are to be shown.
readOnly	(Optional) Boolean: Displays the attributes as read-only (true) or in edit mode (false). This value does not override the state of the attributes in the repository. The value of this parameter does override individual readonly attributes that are set to false on the docbaseattribute controls in the UI.

---

## Elements

By default, the attributes component is configured to display a set of attributes for docbase documents, folders, and system objects (sysobjects), depending on the selected object type. The component can be extended to present a custom UI for any object type.

The following table describes the elements that you can customize in the attributes component definition for dm\_sysobjects. In the scopes of dm\_folder and dm\_document, this component has only the <showpagesastabs> configurable element.

---

<showpagesastabs>	Set to true to display attribute categories as tabs.
<filter>	Contains <enableShowAll>. Determines the criteria for activating the enableShowAll functionality. In the default configuration, the filter attribute "role" is set to "administrator". As a result, only administrators are able to view the extended set of attributes. Applies all types.

<code>&lt;filter&gt;.&lt;enableShowAll&gt;</code>	Boolean. True to display the 'Show all attributes' link within the filter. Applies to all types.
<code>&lt;readOnly&gt;</code>	Boolean. True to display the attributes as read-only. Applies to all types. This element does the same operation as the readOnly parameter. The readOnly parameter takes precedence.

---

## Usage

You can configure the layout of the attributes display using the `docbaseattributelist` control (see ). See the control description for details. Each category of attributes that is defined in the list layout Configuration file can be shown as tabs or inline in the properties UI. Set the value of `showpagesastabs` in the Configuration file to true to display category tabs.

The **Show All** attributes functionality is provided by an alternative layout to the base attributes layout. The layout is switched when the user clicks a link. The link is displayed if the `enableshowall` parameter is set to true.

### Using filters

You can configure the attributes definition to display a different UI based on the user role. The following example allows administrators to see all attributes and allows the attributes to be editable if they are editable in the repository:

```
<filter role='administrator'>
  <enableShowAll>true</enableShowAll>
</filter>
<readOnly>>false</readOnly>
```

### How Attributes are Saved

The OK button in the properties component has an event handler in the parent class `DialogContainer`, which calls `canCommitChanges()` and `onCommitChanges()`. The latter method is implemented in the `Properties` class to save the object.

## attributes (type dm\_externalresult)

### Purpose

This component allows the user to view name and URL for an object that is in an external source, returned by a query. Note that `dm_externalresult` is not a real doctype type, but a pseudotype used only in the WDK for scoping.

## Configuration File

webcomponent\config\library\attributes\attributes\_dm\_externalresult\_component.xml

## Scope

type	dm_externalresult
------	-------------------

## Caller

This component is called by the container component [properties, page 908](#).

## Parameters

entryId	(Required) Internal key to the query results.
queryId	(Required) Internally used to identify the current search context

# attributes (type dm\_folder)

**Purpose** Displays the primary attributes of the selected folder and includes a link to expand the attributes list or to display all attributes.

## Configuration File

webcomponent\config\library\attributes\attributes\_dm\_folder\_component.xml

## Parent Definition

attributes:webcomponent/config/library/attributes/attributes\_dm\_sysobject\_component.xml

## Scope

---

type	dm_folder
------	-----------

---

## Caller

This component is called by the container component [properties, page 908](#).

## Parameters

These parameters are inherited from `attributes:webcomponent/config/library/attributes/attributes_dm_sysobject_component.xml`.

---

enableShowAll	(Optional) Boolean: Displays the 'Show all attributes' link (true) or hides it (false). The default depends on the user's role: true for administrators, false for all non-administrator users.
objectId	(Required) (Required) This is the ID of the object for which attributes are to be shown.
readOnly	(Optional) Boolean: Displays the attributes as read-only (true) or in edit mode (false). This value does not override the state of the attributes in the repository. The value of this parameter does override individual readonly attributes that are set to false on the doabaseattribute controls in the UI.

---

## Elements

By default, the attributes component is configured to display a set of attributes for doabase documents, folders, and system objects (sysobjects), depending on the selected object type. The component can be extended to present a custom UI for any object type.

The following table describes the elements that you can customize in the attributes component definition for `dm_sysobjects`. In the scopes of `dm_folder` and `dm_document`, this component has only the `<showpagesastabs>` configurable element.

---

<code>&lt;showpagesastabs&gt;</code>	Set to true to display attribute categories as tabs.
<code>&lt;filter&gt;</code>	Contains <code>&lt;enableShowAll&gt;</code> . Determines the criteria for activating the <code>enableShowAll</code> functionality. In the default configuration, the filter attribute <i>role</i> is set to <i>administrator</i> . As a result, only administrators are able to view the extended set of attributes. Applies to all types.
<code>&lt;filter&gt;.&lt;enableShowAll&gt;</code>	Boolean: Set to true to display the 'Show all attributes' link within the filter. Applies to all types.
<code>&lt;readOnly&gt;</code>	Boolean: Set to true to display the attributes as read-only. Applies to all types. This element performs the same operation as the <code>readOnly</code> parameter. The <code>readOnly</code> parameter takes precedence.

---

## Usage

You can configure the layout of the attributes display using the `docbaseattributelist` control (see ). See the control description for details. Each category of attributes that is defined in the list layout Configuration file can be shown as tabs or inline in the properties UI. Set the value of `showpagesastabs` in the Configuration file to true to display category tabs.

The **Show All** attributes functionality is provided by an alternative layout to the base attributes layout. The layout is switched when the user clicks a link. The link is displayed if the `enableshowall` parameter is set to true.

### Using filters

You can configure the attributes definition to display a different UI based on the user role. The following example allows administrators to see all attributes and allows the attributes to be editable if they are editable in the repository:

```
<filter role='administrator'>
  <enableShowAll>true</enableShowAll>
</filter>
<readOnly>>false</readOnly>
```

### How Attributes are Saved

The OK button in the properties component has an event handler in the parent class `DialogContainer`, which calls `canCommitChanges()` and `onCommitChanges()`. The latter method is implemented in the `Properties` class to save the object.

## attributes (type dm\_sysobject)

**Purpose** The attributes component displays the primary attributes of the selected object and includes a link to expand the attributes list or to display all attributes. The attributes component is designed to run within the container component [properties](#), page 908.

### Configuration File

`webcomponent\config\library\attributes\attributes_dm_sysobject_component.xml`

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

This component is called by the actions [properties \(type dm\\_sysobject\)](#), page 369 and [attributes \(type dm\\_sysobject\)](#), page 234, and is displayed in the component [properties](#), page 908.

### Parameters

---

<code>enableShowAll</code>	(Optional) Boolean: Displays the 'Show all attributes' link (true) or hides it (false). The default depends on the user's role:
----------------------------	---

---

	true for administrators, false for all non-administrator users.
objectId	(Required) The ID of the object for which attributes are to be shown.
readOnly	(Optional) Boolean: Displays the attributes as read-only (true) or in edit mode (false). This value does not override the state of the attributes in the repository. The value of this parameter does override individual readOnly attributes that are set to false on the doctypeattribute controls in the UI.

---

## Elements

By default, the attributes component is configured to display a set of attributes for doctype documents, folders, and system objects (sysobjects), depending on the selected object type. The component can be extended to present a custom UI for any object type.

The following table describes the elements that you can customize in the attributes component definition for dm\_sysobjects. In the scopes of dm\_folder and dm\_document, this component has only the <showpagesastabs> configurable element.

---

<showpagesastabs>	Set to true to display attribute categories as tabs.
<filter>	Contains <enableShowAll>. Determines the criteria for activating the enableShowAll functionality. In the default configuration, the filter attribute "role" is set to "administrator". As a result, only administrators are able to view the extended set of attributes. Applies all types.
<filter>.<enableShowAll>	Boolean. True to display the 'Show all attributes' link within the filter. Applies to all types.
<readOnly>	Boolean. True to display the attributes as read-only. Applies to all types. This element does the same operation as the readOnly parameter. The readOnly parameter takes precedence.

---

## Usage

You can configure the layout of the attributes display using the doctypeattributelist control (see ). See the control description for details. Each category of attributes that

is defined in the list layout Configuration file can be shown as tabs or inline in the properties UI. Set the value of showpagesastabs in the Configuration file to true to display category tabs.

The **Show All** attributes functionality is provided by an alternative layout to the base attributes layout. The layout is switched when the user clicks a link. The link is displayed if the enableshowall parameter is set to true.

### Using filters

You can configure the attributes definition to display a different UI based on the user role. The following example allows administrators to see all attributes and allows the attributes to be editable if they are editable in the repository:

```
<filter role='administrator'>
  <enableShowAll>true</enableShowAll>
</filter>
<readOnly>false</readOnly>
```

### How Attributes are Saved

The OK button in the properties component has an event handler in the parent class DialogContainer, which calls canCommitChanges() and onCommitChanges(). The latter method is implemented in the Properties class to save the object.

## attributes (type dmc\_rm\_formal\_rec\_folder, dmc\_rm\_formal\_rec\_cabinet)

**Purpose** Displays the primary attributes of the selected dmc\_rm\_formal\_record object or dmc\_rm\_formal\_rec\_cabinet and includes a link to expand the attributes list or to display all attributes. Requires a Records Manager license.

### Configuration File

```
webcomponent\config\library\records\declarerecord\attribute_rm_formalfolder_
component.xml
```

### Parent Definition

```
recordattributes:webcomponent/config/library/records/declarerecord/attribute_rm_
formalrecord_component.xml
```



**Scope**

type	dmc_rm_formal_rec_folder, dmc_rm_formal_rec_cabinet
------	--

**Parameters**

These parameters are inherited from xforms:webcomponent/config/xforms/xforms\_component.xml.

objectId	(Required) Object ID of the form.
readOnly	(Optional) Specifies whether the form should be shown in edit or view mode. In edit mode, WDK buttons are not hidden (form is modal). In view mode, form buttons are disabled, and WDK buttons are displayed (Close and Help).

**authenticate**

**Purpose** Enables a user to log in to an additional content repository. Verifies the user's name and password.

**Configuration File**

wdk\config\authenticate\_component.xml

**Parent Definition**

login:wdk/config/login\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [launchauthenticate](#), page 321.

**Parameters**

---

docbase	(Required) Name of the content repository
docbroker	(Optional) Name of the connection broker
docbrokerPort	(Optional) Port number of the connection broker
suppressLoginEvent	(Optional) Boolean: Prevents or enables the login event. Set to true to suppress the client events loginCancel and loginSuccess used by Application Connectors.

---

## authenticateexternalsource

**Purpose** Enables a user to provide credentials to an external source.

**Configuration File**

wdk\config\authenticateexternalsource\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [launchauthenticateexternalsource](#), page 322.

**Parameters**

---

externalSource	(Required) Path to the external source
----------------	--

---

---

# backdetector

**Purpose** Detects the browser's back button event during nested navigation to another component and prevents re-execution of operation unintentionally for operations that execute on component initialization, for example, UCF content transfer.

## Configuration File

wdk\config\backdetector\_component.xml

## Scope

All.

## Caller

This component is called by all containers that extend the 5.3 ContentTransferServiceContainer class.

## Parameters

---

ticket	(Required) Ticket is generated by a call to BackDetector.setComponentNested() in ContentTransferServiceContainer. The caller can override BackDetector.setComponentNested() to generate a ticket internally and perform the redirection accordingly.
--------	--

---

## Usage

To handle the Back button event from a nested component in your custom component, add a call to BackDetector in your event handler that handles the return from nesting:

```
if (BackDetector.clickBackButton(map))
{
    //abort the operation
}
```

## cabinetslocator

**Purpose** The cabinetslocator component is a very basic sysobject locator that displays folders and the content type.

### Configuration File

webcomponent\config\library\locator\cabinetslocator\_component.xml

### Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

### Scope

All.

### Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.

---

contentType	(Optional) The selected content type for the filetype filter.
flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
folderId	(Optional) The ID of the folder in which to begin the search.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of the selected object.

---

## cancelcheckout (type dm\_sysobject, version 5.2.5)

**Purpose** This is the WDK 5.2.5 cancelcheckout component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility and must be used within the 5.2.5 cancelcheckout container.

### Configuration File

webcomponent\config\library\cancelcheckout\cancelcheckout\_component.xml

### Scope

type	dm_sysobject
version	5.2.5

### Caller

Cannot be called directly. Must be extended.

### Parameters

isReference	(Optional) When set to true, specifies that the object is a reference object
lockOwner	(Optional) The user ID of the user who checked out the object
objectId	(Required) The name of the object that is checked out

### Elements

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
---------------------------	--

## cancelcheckout (type dm\_sysobject)

**Purpose** This is the WDK 5.3 cancelcheckout component. It cancels the checkout of one or more previously checked out objects by removing the lock on the specified objects. This component must run within the 5.3 cancelcheckoutcontainer component or the httpcancelcheckoutcontainer. The cancelcheckout component is defined for dm\_sysobject and dm\_folder.

### Configuration File

webcomponent\config\library\contenttransfer\cancelcheckout\cancelcheckout\_component.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

This component is called by the action [cancelcheckout \(type dm\\_sysobject\), page 240](#) and is displayed in the component [cancelcheckoutcontainer, page 600](#).

### Parameters

---

nodeId	(Optional) Node ID of the checked out object, when it is part of a virtual document.
objectId	(Required)The ID of the checked out doctype object.
vdmRootObjectId	(Optional) Object ID of the root of a virtual document (when operating on a child node).

---

## Elements

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<init-controls>	Initializes the nodescendants radio button value

## cancelcheckout (type dmc\_rm\_formal\_record)

**Purpose** Cancels the checkout of one or more previously checked out dmc\_rm\_formal\_record objects by removing the lock on the specified objects.

### Configuration File

webcomponent\config\library\records\cancelcheckout\cancelcheckout\_component.xml

### Parent Definition

cancelcheckout:/webcomponent/config/library/contenttransfer/cancelcheckout/cancelcheckout\_component.xml

### Scope

type	dmc_rm_formal_record
------	----------------------

### Caller

Called by the [cancelcheckout \(type dm\\_sysobject\), page 240](#) action and runs in the [cancelcheckoutcontainer \(type dmc\\_rm\\_formal\\_record\), page 602](#) container.

## Parameters

These parameters are inherited from `cancelcheckout:/webcomponent/config/library/contenttransfer/cancelcheckout/cancelcheckout_component.xml`.

---

<code>nodeId</code>	(Optional) Node ID of the checked out <code>dmc_rm_formal_record</code> object, when it is part of a virtual document.
<code>objectId</code>	(Required) The ID of the checked out <code>dmc_rm_formal_record</code> object.
<code>vdmRootObjectId</code>	(Optional) Object ID of the root of a virtual document (when operating on a child node).

---

# cancelcheckoutcontainer

**Purpose** This is the WDK 5.3 `cancelcheckoutcontainer` component. It allows users to remove the lock on specified checked-out documents

## Configuration File

`webcomponent\config\library\contenttransfer\cancelcheckout\cancelcheckoutcontainer_component.xml`

## Parent Definition

`combocontainer:wdk/config/combocontainer_component.xml`

## Scope

All.

## Caller

This component is called by the action [cancelcheckout \(type `dm\_sysobject`\), page 240](#) and displays the component [cancelcheckout \(type `dm\_sysobject`\), page 598](#).



## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

<code>component</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) List of arguments and values to pass to each contained component.

## Elements

<code>&lt;bindingcomponentversion&gt;</code>	Specifies the version of components that are supported by the container component. Default: latest
<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propagating the enabled or visible property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<setrepositoryfromobjectid>	Set to true to execute the query against the source repository. Affects reference and foreign objects.

---

## cancelcheckoutcontainer (type dmc\_rm\_formal\_record)

**Purpose** Allows users to remove the lock on specified checked-out dmc\_rm\_formal\_record objects.

### Configuration File

webcomponent\config\library\records\cancelcheckout\cancelcheckoutcontainer\_component.xml

### Parent Definition

cancelcheckoutcontainer:/webcomponent/config/library/contenttransfer/cancelcheckout/cancelcheckoutcontainer\_component.xml

### Scope

---

type	dmc_rm_formal_record
------	----------------------

---

### Caller

Called by the [cancelcheckout \(type dm\\_sysobject\), page 240](#) action and runs the [cancelcheckout \(type dmc\\_rm\\_formal\\_record\), page 599](#) component.

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

<code>component</code>	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

## cancelcheckoutcontainer (version 5.2.5)

**Purpose** This is the WDK 5.2.5 `cancelcheckoutcontainer` component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to allow users to display the `cancelcheckout` component to remove the lock on specified checked-out documents

### Configuration File

`webcomponent\config\library\cancelcheckout\cancelcheckoutcontainer_component.xml`

### Parent Definition

`contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer_component.xml`

### Scope

<code>version</code>	5.2.5
----------------------	-------

## Caller

This component is called by the action [cancelcheckout \(type dm\\_sysobject\), page 240](#) and displays the component [cancelcheckout \(type dm\\_sysobject, version 5.2.5\), page 597](#).

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: 5.2.5
---------------------------	---

---

# categoriesnotenabled

**Purpose** Displays a message that CIS is not enabled for the repository.

## Configuration File

webcomponent\config\navigation\category\categorylist\_component.xml

## Scope

All.

# categoryattributes

**Purpose** Sets the attributes of a work queue category.

## Configuration File

webcomponent\config\library\queuemanagement\categoryattributes\_component.xml

## Scope

All.

## Caller

This component is called by the actions [attributes \(type dmc\\_workqueue\\_category\)](#), [page 236](#), [newWorkQueueCategory, page 351](#), and [properties \(type dmc\\_workqueue\\_category\)](#), [page 370](#), and displayed in the component [workqueuecategoryproperties](#), [page 1152](#).

## Parameters

categoryPath	(Optional) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs
objectId	(Required) (Required) ID of the work queue category object

# categorydrilldown

**Purpose** Allows the user to view CIS categories in a streamline style presentation. The repository must have CI types installed and configured to use this component.

## Configuration File

webcomponent\config\navigation\category\categorydrilldown\_component.xml

## Scope

All.

## Parameters

---

folderId	(Optional) ID of the category folder to navigate
folderPath	(Optional) Specifies the path to the folder in which to start browsing. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder

---

## Elements

Failover is enabled for this component.

This component has configurable columns (see [doclist](#), page 684).

# categorylist

## Purpose

Allows the user to browse CIS categories in a classic (tree and content frame) presentation. The repository must have CI types installed and configured to use this component.

## Configuration File

webcomponent\config\navigation\category\categorylist\_component.xml

**Scope**

All.

**Parameters**

---

folderId	(Optional) ID of the category folder to navigate
folderPath	(Optional) Specifies the path to the folder in which to start browsing. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder

---

**Elements**

Failover is enabled for this component.

This component has configurable columns (see [doclist](#), page 684).

## categorylocator

**Purpose** Enables users to select categories.

**Configuration File**

webcomponent\config\library\locator\categorylocator\_component.xml

**Parent Definition**

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

**Scope**

All.

## Parameters

These parameters are inherited from `sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator_component.xml`.

---

<code>contentType</code>	(Optional) Content type for filetype filter.
<code>flatlist</code>	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<code>folderId</code>	(Optional) ID of the folder in which to begin the search
<code>multiselect</code>	(Optional) Boolean: Set to true to support multiple object selection.
<code>objecttype</code>	(Optional) Type of the selected object

---

## Elements

This component has configurable columns (see [doclist](#), page 684).

# categorylocatorcontainer

**Purpose** Enables users to select categories.

## Configuration File

`webcomponent\config\library\locator\categorylocator_component.xml`

## Parent Definition

`locatorcontainer:webcomponent/config/library/locator/locatorcontainer_component.xml`

## Scope

All.



## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) The ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) The ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

# categorysubscriptionlocator

**Purpose** Enables user to select from a list of category subscriptions.

## Configuration File

webcomponent\config\library\locator\categorylocator\_component.xml

## Parent Definition

subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml.

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

# changehomedocbase

**Purpose** Changes the user's home docbase. The component offers the choice of running the associated job immediately or waiting until the scheduled job.

## Configuration File

webcomponent\config\admin\user\changehomedocbase\_component.xml

## Scope

All.

## Parameters

objectId	(Required) ID of the new home repository
objectname	(Required) User name

# changepassword

**Purpose** The component provides a user interface and functionality that allow users to change their repository passwords. The login and changepassword components are the only two Documentum components that do not require a Documentum connection.

## Configuration File

wdk\config\changepassword\_component.xml

## Scope

All.

## Caller

This component is called from the login dialog.

## Parameters

---

docbase	(Optional) Name of the repository in which the user is changing the password
domain	(Optional) Network domain for the repository
username	(Required) Name of user who is changing the password

---

## Elements

---

<editableusername>	Boolean. True to allow the username to be edited
--------------------	--

---

## changeperformer (type dm\_task)

**Purpose** Allows the user to change the performer of a future task in a workflow.

### Configuration File

webcomponent\config\library\workflow\performers\changeperformer\_component.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

This component is called by the action [changeperformer \(type dm\\_task\), page 243](#) and displayed in the component [changeperformercontainer, page 613](#).

### Parameters

---

objectName	(Required) Activity name of the future task.
taskRuntimeFlag	(Required) State of the workflow task. Valid values: 0=dormant   1=acquired   2=finished   3 – 6 =paused   7=ready
workflowId	(Required) ID of the workflow.
workflowRuntimeState	(Required) State of the workflow activity. Valid values: 0=dormant   1=running   2=finished   3=halted   4=terminated

---

# changeperformercontainer

**Purpose** Allows the user to change the performer of a future task in a workflow.

## Configuration File

webcomponent\config\library\workflow\performers\changeperformercontainer\_  
component.xml

## Scope

All.

## Caller

This component is called by the action [changeperformer \(type dm\\_task\), page 243](#) and displays the component [changeperformer \(type dm\\_task\), page 612](#).

## Parameters

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

# changeroommemberrole

**Purpose** Specifies whether a given user is a contributor or an owner to a room. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\rooms\changeroomrolecontainer\_component.xml

## Scope

All.

## Caller

This component is called by the component [changeroommemberrolecontainer, page 614](#).

# changeroommemberrolecontainer

**Purpose** Container component for changing the roles assigned to room members. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\rooms\changeroomrolecontainer\_component.xml

## Parent Definition

roomuserorgrouplocatorcontainer:webcomponent/config/library/rooms/  
roomuserorgrouplocatorcontainer\_component.xml

## Scope

All.

## Parameters

arrangeselect	(Optional) Boolean: Set to true to allow selection of members in a sequential order.
multiselect	(Optional) Boolean: Set to true to support multiple selections.
roomId	(Required) Object ID of the dmc_room object.

# changeroommemberrolelocator

**Purpose** Displays a list of room members to select for changing their roles. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\rooms\changeroomrolecontainer\_component.xml

## Parent Definition

roomuserorgrouplocator:webcomponent/config/library/rooms/  
roomuserorgrouplocatorcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the component [changeroommemberrolecontainer](#), page 614.

## Parameters

These parameters are inherited from roomuserorgrouplocator:webcomponent/config/library/rooms/roomuserorgrouplocatorcontainer\_component.xml.

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of members in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
groupName	(Optional) ID of a group to which the selection will be restricted
multiselect	(Optional) Boolean: Set to true to support multiple selection.
nobuildinggroups	(Optional) Boolean: Set to true to display only custom groups.
objecttype	(Optional) Object type to display.
roomId	(Required) Object ID of the dmc_room object.
toplevelusers	(Optional) Boolean: Set to true to display only top-level users.

---

# changeroommemberrolestep

**Purpose** This is a container component that displays the second step in the process to pick the role which the selected users will have.

## Configuration File

webcomponent\config\library\rooms\changeroomrolecontainer\_component.xml

## Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml



**Scope**

All.

**Caller**

This component is called by the component [changeroommemberrolecontainer](#), page 614.

**Parameters**

These parameters are inherited from container:wdk/config/container\_component.xml.

---

component	(Required) The component to be displayed (most likely the changeoommemberrole component).
-----------	---

---

## changesearchsources

**Purpose**

Enables a user to change the repositories to search.

**Configuration File**

webcomponent\config\library\changesearchsources\changesearchsources\_component.xml

**Scope**

All.

**Parameters**

---

searchsources	(Optional) A semicolon separated list of repositories to search in the format source:path
---------------	---

---

## changesupervisor (type dm\_workflow)

**Purpose** This component is a container that extends [locatorcontainer](#), page 839. It allows the user to select a new supervisor for a workflow. This container contains the component [userorgrouplocator](#), page 1079.

### Configuration File

```
webcomponent\config\library\workflow\changesupervisor\changesupervisor_
component.xml
```

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

### Scope

---

type	dm_workflow
------	-------------

---

### Caller

This component is called by the action [changesupervisor \(type dm\\_workflow\)](#), page 244.

### Parameters

---

objectId	(Required) ID of the workflow.
----------	--------------------------------

---

## changeucfcheckoutlocation

**Purpose** Sets the checkout location for objects.

**Configuration File**

webcomponent\config\environment\preferences\general\changeucfcheckoutlocation\_component.xml

**Scope**

All.

**Elements**

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
---------------	--

**checkin (type dm\_sysobject, version 5.2.5)****Purpose**

This is the WDK 5.2.5 checkin component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. This component must be used within 5.2.5 checkincontainer.

**Configuration File**

webcomponent\config\library\checkin\checkin\_component.xml

**Scope**

type	dm_sysobject
version	5.2.5

**Caller**

Cannot be called directly. Must be extended.

## Parameters

---

nodeId	(Optional) ID of the node in a virtual document to check in
objectId	(Required) The ID of the object to check in.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: 5.2.5
---------------------------	---

---

## Usage

To customize the checkin layout, use the control. First, create a definition file that is scoped to the custom type. In the JSP page for the custom type, add a tag for the custom attribute list that you wish to display on checkin. Add a tag similar to the following:

```
<dmfx:doctypeattributelist name="attrlist"
  object="obj" attrconfigid="checkin"/>
```

All custom attributes that are in a category appear unless they are listed in the <ignore\_attributes> element. Attributes are displayed in order of category. If the value of showpagesastabs

You can have different attribute lists for checkin, properties, import, and other components.

# checkin (type dm\_sysobject)

## Purpose

This is the WDK 5.3 checkin component. It checks in objects that were checked out and sets properties on the objects. The checkin component is available in all views for objects that are checked out on the user's current machine. This component must be used within 5.3 checkincontainer.

## Configuration File

webcomponent\config\library\contenttransfer\checkin\checkin\_component.xml

## Scope

type	dm_sysobject
------	--------------

## Caller

This component is called by the action [checkin \(type dm\\_sysobject\), page 244](#) and displayed in the component [checkincontainer, page 623](#).

## Parameters

nodeId	(Optional) ID of the node in a virtual document to check in
objectId	(Required) The ID of the object to check in.
vdmRootObjectId	(Optional) Root object ID or the virtual document to check in.

## Elements

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.CheckinProcessor
<transport-class>	Specifies the fully qualified class name for the class that implements the transport; in this case, UCF. Value is: com.documentum.web.contentxfer.ucf.UcfContentTransport
<init-controls>	Initializes values for the minorversion, newversion, and branchrevision radio controls and the makecurrent and checkindescendants checkboxes

## Usage

To customize the checkin layout, use the control. First, create a definition file that is scoped to the custom type. In the JSP page for the custom type, add a tag for the custom attribute list that you wish to display on checkin. Add a tag similar to the following:

```
<dmfx:docheckboxattributelist name="attrlist"
  object="obj" attrconfigid="checkin"/>
```

All custom attributes that are in a category appear unless they are listed in the `<ignore_attributes>` element. Attributes are displayed in order of category. If the value of `showpagesastabs`

You can have different attribute lists for checkin, properties, import, and other components.

# checkin (type dmc\_rm\_formal\_record)

**Purpose** Checks in `dmc_rm_formal_record` objects that were checked out and sets properties on objects.

## Configuration File

`webcomponent\config\library\records\checkin\checkin_component.xml`

## Parent Definition

`checkin:/webcomponent/config/library/contenttransfer/checkin/checkin_component.xml`

## Scope

---

type	dmc_rm_formal_record
------	----------------------

---

## Caller

Called by the [checkin \(type dm\\_sysobject\), page 244](#) action and runs in the [checkincontainer \(type dmc\\_rm\\_formal\\_record\), page 625](#) container.

## Parameters

nodeId	(Optional) ID of the node in a virtual document to check in.
objectId	(Required) The ID of the dmc_rm_formal_record object to check in.
vdmRootObjectId	(Optional) Root object ID or the virtual document to check in.

# checkincontainer

**Purpose** This is the WDK 5.3 checkincontainer component. Checking in multiple objects creates one checkin container and an inner checkin component instance for each object.

## Configuration File

webcomponent\config\library\contenttransfer\checkin\checkincontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [checkin \(type dm\\_sysobject\), page 244](#) and [commitchanges \(type dm\\_sysobject\), page 250](#), and displays the components [checkin \(type dm\\_sysobject\), page 620](#) and [commitchanges, page 646](#), respectively.

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

---

<code>component</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<code>&lt;bindingcomponentversion&gt;</code>	Specifies the version of components that are supported by the container component. Default: latest
<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;setrepositoryfromobjectid&gt;</code>	Set to true to execute the query against the source repository. Affects reference and foreign objects.



---

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<service-class>	Specifies the fully qualified class name for the class that implements the service. Value is: com.documentum.web.contentxfer.impl.CheckinService
<transport-class>	Specifies the fully qualified class name for the class that implements the transport; in this case, UCF. Value is: com.documentum.web.contentxfer.ucf.UcfContentTransport

---

## checkincontainer (type dmc\_rm\_formal\_record)

**Purpose** Checking in multiple dmc\_rm\_formal\_record objects creates one checkin container and an inner checkin component instance for each object.

### Configuration File

webcomponent\config\library\records\checkin\checkincontainer\_component.xml

### Parent Definition

checkincontainer:/webcomponent/config/library/contenttransfer/checkin/  
checkincontainer\_component.xml

### Scope

---

type	dmc_rm_formal_record
------	----------------------

---

### Caller

Called by the [checkin \(type dm\\_sysobject\), page 244](#) action and runs the [checkin \(type dmc\\_rm\\_formal\\_record\), page 622](#) component.

## Parameters

These parameters are inherited from combocontainer:wdk/config/combocontainer\_component.xml.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## checkincontainer (version 5.2.5)

**Purpose** This is the WDK 5.2.5 checkincontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. Checking in multiple objects creates one checkin container and an inner checkin component instance for each object.

### Configuration File

webcomponent\config\library\checkin\checkincontainer\_component.xml

### Parent Definition

contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml

### Scope

---

version	5.2.5
---------	-------

---

**Caller**

Cannot be called directly. Must be extended.

**Parameters**


---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

**Elements**


---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
---------------------------	--

---

**checkout (type dm\_sysobject, version 5.2.5)**

**Purpose** This is the WDK 5.2.5 checkout component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility and must be used within the 5.2.5 checkout container.

**Configuration File**

webcomponent\config\library\checkout\checkout\_component.xml

## Scope

---

type	dm_sysobject
version	5.2.5

---

## Parameters

---

isReference	(Optional) When set to true, specifies that the object is a reference object
lockOwner	(Optional) The user ID of the user who checked out the object
objectId	(Required) The name of the checked out docbase object

---

# checkout (type dm\_sysobject)

## Purpose

This is the WDK 5.3 checkout component. It checks out objects from the repository without launching an editing application. The checkout component is available in all views for objects that are checked out on the user's current machine. This component must be used within the 5.3 checkoutcontainer. The checkoutcontainer component extends [contentxfercontainer](#), page 653. Checking out multiple objects creates one checkout container and an inner checkout component instance for each object.

## Configuration File

webcomponent\config\library\contenttransfer\checkout\checkout\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [checkout \(type dm\\_sysobject\), page 247](#) and is displayed in the component [checkoutcontainer, page 632](#).

## Parameters

isIgnoreDescendents	(Optional) Boolean value that indicates whether or not to check out descendants in a virtual document.
isLockOnly	(Optional) Boolean value that indicates whether or not to check out objects that are locked.
nodeId	(Optional) r_object_id of a virtual document node to check out.
objectId	(Required) The ID of the object to check out.
vdmRootObjectId	(Optional) r_object_id of the virtual document root.

## Elements

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.CheckoutProcessor
<init-controls>	Initializes the nodescendents and xmlcheckoutall radio button values

# checkout (type dmc\_rm\_formal\_record)

## Purpose

Checks out dmc\_rm\_formal\_record objects from the repository without launching an editing application. This component is available in all views for

dmc\_rm\_formal\_record objects that are checked out on the user's current machine. Checking out multiple objects creates one checkout container and an inner checkout component instance for each dmc\_rm\_formal\_record object.

## Configuration File

webcomponent\config\library\records\checkout\checkout\_component.xml

## Parent Definition

checkout:/webcomponent/config/library/contenttransfer/checkout/checkout\_component.xml

## Scope

---

type	dmc_rm_formal_record
------	----------------------

---

## Caller

Called by the [checkout \(type dm\\_sysobject\), page 247](#) action and runs in the [checkoutcontainer \(type dmc\\_rm\\_formal\\_record\), page 634](#) container.

## Parameters

These parameters are inherited from checkout:/webcomponent/config/library/contenttransfer/checkout/checkout\_component.xml.

---

isIgnoreDescendents	(Optional) Boolean value that indicates whether or not to check out descendants in a virtual document.
isLockOnly	(Optional) Boolean value that indicates whether or not to check out objects that are locked.
nodeId	(Optional) r_object_id of a virtual document node to check out.

---

objectId	(Required) The ID of the object to check out.
vdmRootObjectId	(Optional) r_object_id of the virtual document root.

---

## checkoutbyother

**Purpose**        Informs the user that the requested object is checked out by another user.

### Configuration File

webcomponent\config\library\checkout\checkoutbyother\_component.xml

### Scope

All.

### Parameters

---

message	(Optional) The message to display to the user.
---------	--

---

## checkoutbyothercontainer

**Purpose**        Container for informing the user that a requested object is checked out by another user.

### Configuration File

webcomponent\config\library\checkout\checkoutbyothercontainer\_component.xml

**Parent Definition**

dialogcontainer:wdk/config/dialogcontainer\_component.xml

**Scope**

All.

**Parameters**

---

component	(Required) Supplied by the action execution class). The name of the component to contain.
message	(Optional) The message to display to the user.

---

## checkoutcontainer

**Purpose**

This is the WDK 5.3 checkoutcontainer component. It is the container for the checkout process.

**Configuration File**

webcomponent\config\library\contenttransfer\checkout\checkoutcontainer\_component.xml

**Parent Definition**

combocontainer:wdk/config/combocontainer\_component.xml

**Scope**

All.



## Caller

This component is called by the action [checkout](#) (type `dm_sysobject`), [page 247](#) and is displays the component [checkout](#) (type `dm_sysobject`), [page 628](#).

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

<code>component</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) List of arguments and values to pass to each contained component.

## Elements

<code>&lt;bindingcomponentversion&gt;</code>	Specifies the version of components that are supported by the container component. Default: latest
<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.

<control-propagation>.<property name="visible"/>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<service-class>	Specifies the fully qualified class name for the class that implements the service. Value is: com.documentum.web.contentxfer.impl.CheckoutService
<transport-class>	Specifies the fully qualified class name for the class that implements the transport; in this case, UCF. Value is: com.documentum.web.contentxfer.ucf.UcfContentTransport

---

## checkoutcontainer (type dmc\_rm\_formal\_record)

**Purpose** Container for the checkout process for dmc\_rm\_formal\_record objects.

### Configuration File

webcomponent\config\library\records\checkout\checkoutcontainer\_component.xml

### Parent Definition

checkoutcontainer:/webcomponent/config/library/contenttransfer/checkout/checkoutcontainer\_component.xml

### Scope

---

type	dmc_rm_formal_record
------	----------------------

---

## Caller

Called by the [checkout \(type dm\\_sysobject\), page 247](#) action and runs the [checkout \(type dmc\\_rm\\_formal\\_record\), page 629](#) component.

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

---

<code>component</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class. The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class. List of arguments and values to pass to each contained component.

---

# checkoutcontainer (version 5.2.5)

**Purpose** This is the WDK 5.2.5 `checkoutcontainer` component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is included for backward compatibility. This is the WDK 5.2.5 container used to display the `checkout` component during the checkout process.

## Configuration File

`webcomponent\config\library\checkout\checkoutcontainer_component.xml`

## Parent Definition

`contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer_component.xml`

**Scope**

---

version	5.2.5
---------	-------

---

**Caller**

Cannot be called directly. Must be extended.

**Parameters**

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

**Elements**

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: 5.2.5
---------------------------	---

---

## checkoutprocess (type dm\_process)

**Purpose** Checks out a dm\_process object (workflow template). The container component extends [contentxfercontainer](#), [page 653](#).

**Configuration File**

webcomponent\config\library\checkout\checkout\_component.xml

**Scope**

---

type	dm_process
------	------------

---

**Caller**

This component is called by the action [checkout](#) (type `dm_process`), page 246.

**Parameters**

---

objectId	(Required) The ID of the workflow template to check out.
----------	--

---

## choosecontributors

**Purpose** Sets the list of contributors for a room. Requires Collaborative Edition.

**Configuration File**

`webcomponent\config\library\rooms\newroom_component.xml`

**Parent Definition**

`userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator_component.xml`

**Scope**

All.

**Parameters**

These parameters are inherited from `userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator_component.xml`.

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of members in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display.

---

## chooseowners

**Purpose** Sets the owners for a room. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\rooms\newroom\_component.xml

### Parent Definition

userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml

### Scope

All.

### Parameters

These parameters are inherited from userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml.

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of members in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display.

---

## choosetransformationtype

**Purpose** A transformation selection component. Requires Media Transformation Services. Enables the user to select the available media profiles.

### Configuration File

webcomponent\config\library\transformation\choosetransformationtype\_component.xml

### Scope

All.

### Caller

This component is called by the actions [importrenditiongeneric \(type dmr\\_content\)](#), page 321 and [transformation \(type dm\\_sysobject\)](#), page 440, and displayed in the component [transformationcontainer](#), page 1056.

# clientapplicationinstaller

**Purpose** Lists the Application Connector installers that can be accessed from the current application server. Supports command line arguments for the installers and does not require authentication.

## Configuration File

webcomponent\config\library\clientapplicationinstaller\clientapplicationinstaller\_component.xml

## Scope

All.

## Caller

This component can be called by URL, JavaScript, or a component class.

## Parameters

---

application	(Optional) Specifies a comma-delimited list of applications to display
-------------	--

---

## Elements

The application installer is constructed by concatenating current application server URL, client application installer, and installer parameters

---

<applicationinstallers>	Contains one or more <applicationinstaller> elements that define the installers
<applicationinstaller>	Defines an application installer. Contains <application>, <label>, <description>, <installer>, and <cmdargs>.
<application>	



---

<label>	Contains a string label or NLS ID key to be displayed
<description>	Contains a string description or NLS ID key to be displayed that describes the application
<installer>	Path to the installer program relative to the Web application root, for example, /webcomponent/install/appconnector/installer.exe
<cmdargs>	Supports the following system variables as command line arguments: \$PROTOCOL, \$PORT, APPLICATION, \$APP_SERVER, \$ROOT_PATH. \$APP_SERVER dynamically passes the application server URL to the installer, for example, /v"webtopUrl=\$APP_SERVER"http://myserver/webtop.

---

## clipboard

**Purpose** The clipboard component displays the contents of the current user's clipboard and supports link and copy operations.

### Configuration File

webcomponent\config\environment\clipboard\clipboard\_component.xml

### Scope

All.

### Caller

This component is called by the action [viewclipboard](#), page 471.

## Elements

---

<code>&lt;dragdrop&gt;</code>	Supports the drag and drop feature.
<code>&lt;sourceactions&gt;</code>	Contains zero or more <code>&lt;sourceaction&gt;</code> classes that support drag and drop actions on sources in the component.
<code>&lt;dataproviders&gt;</code>	Contains one or more <code>&lt;dataprovider&gt;</code> elements.
<code>&lt;dataprovider&gt;</code>	Contains a <code>&lt;format&gt;</code> element and a <code>&lt;provider&gt;</code> element.
<code>&lt;dataprovider&gt;.&lt;format&gt;</code>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<code>&lt;dataprovider&gt;.&lt;provider&gt;</code>	Fully qualified class name for the class that implements <code>IDragDropDataProvider</code> and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

## Usage

To disable clipboard functions for specific object types, use the `addtoclipboard` action (see [addtoclipboard \(type dm\\_sysobject\)](#), page 219).

# columnselector

**Purpose** Edits display preferences. The JSP page contains an `attributeselector` control that generates the lists of columns, selected columns, and selection buttons.

## Configuration File

`webcomponent\config\environment\preferences\display\columnselector_component.xml`

## Scope

All.

## Parameters

---

columns	(Required) A serialized list of items to populate the available items list box. See multiselector control for details on the list format.
defaultSelectedColumns	(Optional) A serialized list of items to populate the selected items list box before the user performs an initial selection.
docbaseTypes	(Optional) Specifies the repository types, in addition to the 'default' option, to populate the attribute selector's docbase types drop-down list. If omitted, no docbase types will be available for selection. This is a simple list of alternating type names and values. To pass in as a string argument to the column selector, the list may be passed to ColumnListUtil.
enableOrdering	(Optional) Boolean: true to enable the user to control the order of columns.
selectedColumns	(Optional) A serialized list of items to populate the selected columns list box.
showRepeatingAttributes	(Optional) Specifies whether repeating attributes should be shown when selecting one of the repository types specified by 'docbasetypes.' Default is true.

---

# columnselectorcontainer

## Purpose

Edits display preferences. The JSP page contains an attributeselector control that generates the lists of columns and selected columns and the selection buttons.

## Configuration File

webcomponent\config\environment\preferences\display\columnselectorcontainer\_component.xml

## Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml

## Scope

All.

## Parameters

component	(Required) Supplied by the action execution class). The name of the component to contain.
displaytitle	(Optional) A title for the component.

# combocontainer

## Purpose

Extends the component [wizardcontainer, page 1148](#) and adds support for multiple instances of the same contained component. This allows the user to perform multi-select operations for a component. An Apply to all prompt is displayed when the user selects multiple operations.

## Configuration File

wdk\config\combocontainer\_component.xml

## Parent Definition

wizardcontainer:wdk/config/wizardcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [abortworkflow](#) (type `dm_workflow`), page 205, [delete](#) (type `dm_sysobject`), page 269, [delete](#) (type `dm_workflow`), page 270, [delete](#) (type `dmc_workqueue`), page 274, [delete](#) (type `dmc_workqueue_category`), page 272, [delete](#) (type `dmc_workqueue_doc_profile`), page 272, [delete](#) (type `dmc_workqueue_policy`), page 273, [editstartwfnote](#) (type `dmi_package`), page 296, [edittasknote](#) (type `dmi_package`), page 297, [modifyversionlabels](#) (type `dm_sysobject`), page 335, [removeattachment](#) (type `dmi_package`), page 383, [removestartwfattachment](#) (type `dmi_package`), page 389, [removestartwfwattachment](#) (type `dmi_wf_attachment`), page 390, [removewfattachment](#) (type `dmi_wf_attachment`), page 393, [rename](#) (type `dm_sysobject`), page 395, [resumetask](#) (type `dm_task`), page 408, [resumeworkflow](#) (type `dm_workflow`), page 409, [sendto](#) (type `dm_sysobject`), page 421, [submitforcategorization](#) (type `dm_document`), page 431, [suspend\\_queued\\_task](#) (type `dm_task`), page 432, and [vdmcopyoption](#) (type `dm_sysobject`), page 454.

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<control-propagation>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls
-----------------------	--

<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	are propagated to the JSP shown for each object being checked in.  (Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

---

## Usage

The parameters, component and componentArgs, are added by the LaunchComponent action execution class in order to support actions on multiple objects. Do not call the container by URL.

# commitchanges

**Purpose** Extends the component [checkin](#) (type `dm_sysobject`), [page 620](#) and checks in or cancels changes changes to a virtual document (removes attributes from session state) when a user chooses to view a virtual document that is checked out.

## Configuration File

`webcomponent\config\library\vdm\commitchanges\commitchanges_component.xml`

## Parent Definition

`checkin:webcomponent/config/library/checkin/checkin_component.xml`

## Scope

All.

**Caller**

This component is called by the action [commitchanges](#) (type `dm_sysobject`), page 250 and displayed in the component [checkincontainer](#), page 623.

**Parameters**

objectId	(Required) ID of the virtual document to check in
----------	---

## completefailedautoworkflowtask

**Purpose**

The `completefailedautoworkflowtask` component forces completion of a failed automatic workflow task. It is invoked from the task manager component.

**Configuration File**

`webcomponent\config\library\workflow\taskmanager\completeautotask_component.xml`

**Scope**

All.

**Parameters**

objectId	(Required) ID of the workflow task
----------	------------------------------------

## component\_preferences

**Purpose**

Enables users to set preferences for portals.

### Configuration File

webcomponent\config\environment\preferences\portal\component\component\_preferences\_component.xml

### Scope

All.

### Parameters

---

componentid	(Optional) For internal use only.
context	(Optional) For internal use only.

---

## componentincludetest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

### Caller

Must call by URL to test



## componentlist

**Purpose** Generates a list of components with their scopes and other information. This list can be generated (after login to the application) through a URL such as the following: `http:// host_name:8080/ webtop/component/componentlist/` . Displays all components in the application, with links. Each link displays the name of the config file, NLS bundle, parameters, whether it is a container, whether it is configurable, the fully qualified class name, and the component description from the component definition file.

### Configuration File

`webcomponent\config\componentList\component_list.xml`

### Scope

All.

## componenttestbed

**Purpose** Obsolete component that was used to test WDK 5.1 components.

### Configuration File

`webcomponent\config\componentTestbed\testbed_index_component.xml`

### Scope

All.

## componenturltest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

### Caller

Must call by URL to test

## configservicemaintest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

## Parameters

---

folderPath	(Optional) Full repository path to the folder.
------------	--

---

# configservicetest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

## Configuration File

wdk\config\fxtest\_component.xml

## Scope

All.

## Parameters

---

testInstructs	(Optional) Instructions for testing.
---------------	--------------------------------------

---

# configuredocbroker

**Purpose** This component allows the user to select the connection brokers that will supply repositories for login. A dropdown list of connection brokers that are specified in the application server dmcl.ini file is displayed. The user can enter the name of another connection broker to the list. The component verifies the existence and connection to connection brokers when the user clicks **OK**.

## Configuration File

wdk\config\configuredocbroker\_component.xml

## Scope

All.

## Caller

This component is called by the configuredocbroker action.

# container

## Purpose

The container component is a generic container component definition. It is the base component for all containers.

## Configuration File

wdk\config\container\_component.xml

## Scope

All.

## Parameters

---

component	(Required) The contained component. Set the requiresVisit attribute to true to require the component to be visited before an OK button is displayed.
-----------	--

---

## Usage

Set `requiresVisit` attribute to `true` to declare that a particular component must be visited before the container can commit changes. In the following example, the `attributes` component must be viewed before the user can commit changes:

```
<contains>
  <component>newFolder</component>
  <component requiresVisit='true'>attributes</component>
  <component>permissions</component>
</contains>
```

You can also declare the visit requirement in the individual component definition. Add the following tag to a component definition:

```
<requiresVisitBeforeCommit>true</requiresVisitBeforeCommit>
```

# contentxfercontainer

**Purpose** Extends the container [combocontainer](#), [page 644](#) and serves as the base component for all content transfer containers. This container ensures that the applets have been installed and reads the content transfer settings in the application definition (`app.xml`) and passes them to the contained component. Contains `cancelcheckoutcontainer`, `checkincontainer`, and `checkoutcontainer`.

## Configuration File

```
webcomponent\config\library\contentxfer\contentxfercontainer_component.xml
```

## Parent Definition

```
combocontainer:wdk/config/combocontainer_component.xml
```

## Scope

All.

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<control-propagation>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<control-propagation>.<property name="enabled"/>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<control-propagation>.<property name="visible"/>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

---

## Usage

The required parameters are supplied by the action execution class. Do not call this container by URL.

## debug\_preferences

**Purpose** Sets debugging preferences during application development. This component is designed to run within the preferences component, which is a property sheet container.

### Configuration File

webcomponent\config\environment\preferences\debug\debug\_preferences\_component.xml

### Scope

All.

## declareformalrecord (entitlement recordsmanager)

**Purpose** Declare—that is, create—formal record objects from existing objects. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\records\declarerecord\declareformalrecord\_component.xml

### Scope

---

entitlement

recordsmanager

---

## Caller

Runs in the [declareFormalRecordContainer \(entitlement recordsmanager\), page 656](#) container.

# declareFormalRecordContainer (entitlement recordsmanager)

**Purpose** Container for the declareformalrecord (entitlement recordsmanager) component. Requires a Records Manager license.

## Configuration File

webcomponent\config\library\records\declarerecord\declareformalrecordcontainer\_component.xml

## Parent Definition

newdoccontainer:webcomponent/config/library/create/newdoccontainer\_component.xml

## Scope

---

entitlement	recordsmanager
-------------	----------------

---

## Caller

Runs the [declareformalrecord \(entitlement recordsmanager\), page 655](#) component.



## Parameters

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Optional) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

# delegateworkflowtask

**Purpose** The delegateworkflowtask component delegates workflow tasks.

## Configuration File

webcomponent\config\library\workflow\taskmanager\delegateworkflowtask\_component.xml

## Scope

All.

## Caller

This component is invoked from the task manager component.

## Parameters

objectId	(Required) ID of the task
----------	---------------------------

# delete

**Purpose** Base component for deleting work queue-related objects.

## Configuration File

webcomponent\config\library\queuemanagement\delete\deleteworkqueueobject\_component.xml

## Scope

All.

## Parameters

---

objectId	(Required) ID of the object to delete.
objectname	(Optional) The name of the object.

---

# delete (type dm\_folder)

**Purpose** The delete component deletes dm\_folders and other dm\_sysobjects, deletes inbox notifications (dm\_notification and dm\_queued), and deletes folders (queries the user whether all objects should be deleted from a folder selected for deletion). This component must be used within the deletecontainer component.

## Configuration File

webcomponent\config\library\delete\deletefolder\_component.xml

**Scope**

type	dm_folder
------	-----------

**Parameters**

folderId	(Optional) ID of the parent folder of the file or folder that is to be deleted. Applies only to dm_folder and dm_sysobject types.
objectId	(Required) ID of the object to be deleted. Applies to all types.

**delete (type dm\_notification)****Purpose**

The delete component deletes dm\_folders and other dm\_sysobjects, deletes inbox notifications (dm\_notification and dm\_queued), and deletes folders (queries the user whether all objects should be deleted from a folder selected for deletion). This component must be used within the deletecontainer component.

**Configuration File**

webcomponent\config\library\delete\delete\_notification\_component.xml

**Scope**

type	dm_notification
------	-----------------

**Parameters**

---

objectId	(Required) ID of the object to be deleted. Applies to all types.
objectName	(Optional) Name of the object to be deleted. Applies only to dm_notification and dm_queued types.

---

## delete (type dm\_queued)

**Purpose** The delete component deletes dm\_folders and other dm\_sysobjects, deletes inbox notifications (dm\_notification and dm\_queued), and deletes folders (queries the user whether all objects should be deleted from a folder selected for deletion). This component must be used within the deletecontainer component.

**Configuration File**

webcomponent\config\library\delete\delete\_queued\_component.xml

**Scope**

---

type	dm_queued
------	-----------

---

**Parameters**

---

objectId	(Required) ID of the object to be deleted. Applies to all types.
objectName	(Optional) Name of the object to be deleted. Applies only to dm_notification and dm_queued types.

---

## delete (type dm\_sysobject)

**Purpose** The delete component deletes dm\_folders and other dm\_sysobjects, deletes inbox notifications (dm\_notification and dm\_queued), and deletes folders (queries the user whether all objects should be deleted from a folder selected for deletion). This component must be used within the deletecontainer component.

### Configuration File

webcomponent\config\library\delete\deletedocument\_component.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Parameters

---

folderId	(Optional) ID of the parent folder of the file or folder that is to be deleted. Applies only to dm_folder and dm_sysobject types.
objectId	(Required) ID of the object to be deleted. Applies to all types.

---

## delete (type dmc\_comment)

**Purpose** Deletes a comment from a thread.

### Configuration File

webcomponent\config\library\discussion\deletecomment\_component.xml

## Scope

---

type	dmc_comment
------	-------------

---

## Parameters

---

objectId	(Required) ID of the dmc_comment object to delete.
----------	--

---

# delete (type dmc\_notepage)

**Purpose** This component is used to delete dmc\_notepage objects. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\notepage\deletenotepage\_component.xml

## Parent Definition

delete:webcomponent/config/library/delete/deletedocument\_component.xml

## Scope

---

type	dmc_notepage
------	--------------

---

## Parameters

These parameters are inherited from delete:webcomponent/config/library/delete/deletedocument\_component.xml.

folderId	(Optional) ID of the parent folder of the file or folder that is to be deleted. Applies only to dm_folder and dm_sysobject types.
objectId	(Required) Object ID of the notepage to be deleted

## delete (type dmc\_room)

**Purpose** Deletes room objects. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\rooms\deleteroom\_component.xml

### Scope

type	dmc_room
------	----------

### Parameters

folderId	(Optional) Not required for deleting dmc_room objects.
objectId	(Required) Object ID of the dmc_room object to delete.

## delete (type dmc\_workqueue\_category)

**Purpose** Deletes a work queue category. A work queue category can be deleted only if it does not contain any work queues or other categories.

## Configuration File

webcomponent\config\library\queuemanagement\delete\deleteworkqueuecategory\_ component.xml

## Parent Definition

delete:webcomponent/config/library/queuemanagement/delete/deleteworkqueueobject\_ component.xml

## Scope

---

type	dmc_workqueue_category
------	------------------------

---

## Parameters

These parameters are inherited from delete:webcomponent/config/library/ queuemanagement/delete/deleteworkqueueobject\_component.xml.

---

objectId	(Required) ID of the work queue category to delete.
objectname	(Optional) Name of the work queue category.

---

# delete (type dmc\_workqueue\_doc\_profile)

**Purpose** Deletes a work queue document profile. A work queue document profile can be deleted only if no work items reference it.

## Configuration File

webcomponent\config\library\queuemanagement\delete\deleteworkqueuedocprofile\_ component.xml



## Parent Definition

delete:webcomponent/config/library/queuemanagement/delete/deleteworkqueueobject\_component.xml

## Scope

type	dmc_workqueue_doc_profile
------	---------------------------

## Parameters

These parameters are inherited from delete:webcomponent/config/library/queuemanagement/delete/deleteworkqueueobject\_component.xml.

objectId	(Required) ID of the work queue document profile to delete.
objectname	(Optional) Name of the work queue document profile.

# delete (type dmc\_workqueue\_policy)

**Purpose** Deletes a work queue policy. A work queue policy can be deleted only if the work queue policy is not referenced by any work item, work queue, or work queue document profile.

## Configuration File

webcomponent\config\library\queuemanagement\delete\deleteworkqueuepolicy\_component.xml

## Parent Definition

delete:webcomponent/config/library/queuemanagement/delete/deleteworkqueueobject\_component.xml

## Scope

---

type	dmc_workqueue_policy
------	----------------------

---

## Parameters

These parameters are inherited from delete:webcomponent/config/library/queuemanagement/delete/deleteworkqueueobject\_component.xml.

---

objectId	(Required) ID of the work queue policy to delete.
objectname	(Optional) Name of the work queue policy.

---

# delete (type dmc\_workqueue)

**Purpose** Deletes a work queue. A work queue can be deleted only if it contains no work items.

## Configuration File

webcomponent\config\library\queuemanagement\delete\deleteworkqueue\_component.xml

## Parent Definition

delete:webcomponent/config/library/queuemanagement/delete/deleteworkqueueobject\_component.xml

## Scope

---

type	dmc_workqueue
------	---------------

---

## Parameters

These parameters are inherited from delete:webcomponent/config/library/queuemanagement/delete/deleteworkqueueobject\_component.xml.

---

objectId	(Required) ID of the work queue to delete.
objectname	(Optional) Name of the work queue.

---

## deletesmartlistcontainer (type dm\_smart\_list)

**Purpose** This component is the component container used to delete dm\_smart\_list objects (saved search objects)

## Configuration File

webcomponent\config\library\delete\deletesmartlistcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

---

type	dm_smart_list
------	---------------

---

## Parameters

These parameters are inherited from combocontainer:wdk/config/combocontainer\_component.xml.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<control-propagation>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<control-propagation>.<property name="enabled"/>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<control-propagation>.<property name="visible"/>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
setrepositoryfromobjectid	Set to true to execute the query against the source repository. Affects reference and foreign objects.

---

---

## deleteworkflow (type dm\_workflow)

**Purpose** The deleteworkflow component deletes aborted workflow objects.

### Configuration File

webcomponent\config\library\workflow\delete\deleteworkflow\_component.xml

### Scope

---

type	dm_workflow
------	-------------

---

### Caller

This component is called by the action [delete \(type dm\\_workflow\), page 270](#), and is displayed in the component [combocontainer, page 644](#).

### Parameters

---

objectId	(Required) ID of the workflow to be deleted
----------	---

---

## dialogcontainer

**Purpose** Extends the container [container, page 652](#), making a single set of arguments available to a single contained component and providing a set of dialog buttons.

### Configuration File

wdk\config\dialogcontainer\_component.xml

## Parent Definition

container:wdk/config/container\_component.xml

## Scope

All.

## Caller

This component is called by the [actionsabortjob](#), [page 204](#), [delete \(type dmc\\_comment\)](#), [page 271](#), [savechanges](#), [page 413](#), [showjobstatusthrough](#), [page 423](#), and [workflowavailability](#), [page 477](#), and displays the components [jobabort](#), [page 824](#), [jobstatusthrough](#), [page 826](#), [savechanges](#), [page 987](#), and [workflowavailability](#), [page 1149](#).

## Parameters

These parameters are inherited from container:wdk/config/container\_component.xml.

---

component	(Required) The contained component. Set the requiresVisit attribute to true to require the component to be visited before an OK button is displayed.
-----------	--

---

# display\_preferences

## Purpose

Sets the user's preferences for the columns to be displayed by another component. This component is presented within the preferences container. Displays the following (generated by the preference control): the component's list of columns in the order they will appear, and checkboxes (if appropriate) to use the cabinets view and to use thumbnails.

## Configuration File

webcomponent\config\environment\preferences\display\display\_preferences\_component.xml

**Scope**

All.

**Usage**

WDK display preferences are edited by the columnselector component. The editing component is configurable as the value of the display\_preferences <editcomponent> element. You cannot display an attribute if is\_searchable is set to false. For more information on configuring display preferences, see *WDK Development Guide*.

## dmr\_contentlocator

**Purpose**

This component is the locator for all repository documents and dmr\_content with folders shown.

**Configuration File**

webcomponent\config\library\locator\dmr\_contentlocators\_component.xml

**Parent Definition**

alldocumentlocator:webcomponent/config/library/locator/documentlocators\_component.xml

**Scope**

All.

## Parameters

---

flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The logical (base) type of the selectable type

---

# dmr\_contentlocatorcontainer

**Purpose** This component is the locator for all docbase documents and dmr\_content with folders shown.

## Configuration File

webcomponent\config\library\locator\dmr\_contentlocators\_component.xml

## Parent Definition

documentlocatorcontainer:webcomponent/config/library/locator/documentlocators\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.



---

flatlist	(Optional) Boolean: Set to true to display a flat list of all selectable objects.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The logical (base) type of the selectable type
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-separated list of objectIds that are selected when the component is displayed

---

## docbaseattributeedittest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

### Caller

Must call by URL to test

## Parameters

---

testInstructs	(Optional) Instructions for testing.
---------------	--------------------------------------

---

## docbaseattributetest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

### Caller

Must call by URL to test

## docbaseattributelistmaintest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

**Scope**

All.

**Caller**

Must call by URL to test

## docbaseattributelisttest

**Purpose**

One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

**Configuration File**

wdk\config\fxtest\_component.xml

**Scope**

All.

**Caller**

Must call by URL to test

## docbaseattributetest

**Purpose**

One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### **Configuration File**

wdk\config\fxtest\_component.xml

### **Scope**

All.

### **Caller**

Must call by URL to test

## **docbaseattributevalidatortest**

### **Purpose**

One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### **Configuration File**

wdk\config\fxtest\_component.xml

### **Scope**

All.

### **Caller**

Must call by URL to test

## docbaseattributevaluetest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

### Caller

Must call by URL to test

## docbasefoldertreetest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

## Caller

Must call by URL to test

# docbaseicontest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

## Configuration File

wdk\config\fxtest\_component.xml

## Scope

All.

## Caller

Must call by URL to test

# docbaselockicontest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

## Configuration File

wdk\config\fxtest\_component.xml

**Scope**

All.

**Caller**

Must call by URL to test

## docbaseobjectMainTest

**Purpose**

One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

**Configuration File**

wdk\config\fxtest\_component.xml

**Scope**

All.

**Caller**

Must call by URL to test

## docbaseobjecttest

**Purpose**

One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

### Caller

Must call by URL to test

### Parameters

---

lifecycleId	(Optional) r_object_id of the document lifecycle to which the object is attached.
objectId	(Optional) r_object_id of the object to be tested.

---

## docbaseobjectvalidatortest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.



**Caller**

Must call by URL to test

**Parameters**


---

testInstructs	(Optional) Instructions for testing.
---------------	--------------------------------------

---

## doabaserepeatingattribute

**Purpose**

Component called by the [doabaseattributevalue, page 80](#) control to display an editing page for a repeating attribute.

**Configuration File**

wdk\config\doabaserepeatingattribute\_component.xml

**Scope**

All.

**Caller**

This component is called by the [doabaseattributevalue, page 80](#) control.

**Parameters**


---

control	(Required) ID of the doabaseattributevalue control.
---------	---

---

**Usage**

When a [doabaseattributevalue, page 80](#) control is used to display an attribute, an Edit link is displayed if the attribute is repeating, that is, if the attribute allows multiple

values. The `docbaserepeatingattribute` component is called by an `onClick` event handler internal to the `docbaseattributevalue` control when the user clicks the Edit link. The event handler nests to the `docbaserepeatingattribute` component, which means that the `docbaserepeatingattribute` component is nested within the component that contains the `docbaseattributevalue` control.

By default, the `docbaserepeatingattribute` component opens the `repeatingAttributes.jsp` page, which displays an editing screen that allows the user to construct a list of values with an Add/Remove button. For repeating attributes without value assistance, a text box is displayed to for entry of values. For repeating attributes with closed-end value assistance, the user adds values by selecting values from a list. For repeating attributes with open-ended value assistance, there is both a text box and a list of values to select from. The JSP page can be customized by modifying the component definition file.

**Note:** The `docbaserepeatingattribute` component can only be called from the `docbaseattributevalue` control.

## docbasesingleattribute

**Purpose** Component called by the [docbaseattributevalue, page 80](#) control to display an editing page for a single-value attribute.

### Configuration File

`wdk\config\docbasesingleattribute_component.xml`

### Scope

All.

### Caller

This component is called by the [docbaseattributevalue, page 80](#) control.

## Parameters

control	(Required) ID of the docbaseattributevalue control.
---------	---

## Usage

When the [docbaseattributevalue, page 80](#) control is used to display an attribute, an Edit link is displayed if the attribute is single-valued with open-ended value assistance. The docbasesingleattribute component is called by an onClick event handler internal to the docbaseattributevalue control when the user clicks the Edit link. The event handler nests to the docbasesingleattribute component, which means that the docbasesingleattribute component is nested within the component that contains the docbaseattributevalue control.

By default, the docbasesingleattribute component opens the singleAttribute.jsp page, which displays an editing screen that allows the user to either select a value from a list or enter a value in a text box. The JSP page can be customized by modifying the component definition file.

**Note:** The docbasesingleattribute component can only be called from the docbaseattributevalue control.

# docformatvalueformattertest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

## Configuration File

wdk\config\fxtest\_component.xml

## Scope

All.

**Caller**

Must call by URL to test

**doclist****Purpose**

Displays the contents of a folder as a list of objects and folders. The user can filter the list to see folders only, documents only, folders and documents, or all objects. The user navigates through a separate tree frame or by clicking on a folder in the doc list. This component is extended in Webtop as the objectlist component.

**Configuration File**

webcomponent\config\navigation\doclist\doclist\_component.xml

**Scope**

All.

**Parameters**

---

folderId	(Optional) Specifies the ID of the folder in which to start browsing
folderPath	(Optional) Specifies the path to the folder in which to start browsing. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder
showAllVersions	(Optional) Set to true to show all versions of the object

---

## Elements

---

<failoverenabled>	This component supports failover. To disable failover, enter a value of false.
<showfilters>	Set to true to show the objectfilters drop-down control
<objectfilters>	Contains filters that define which objects should be shown in the objects selection list.
<objectfilter>	Specifies a filter for the items that are displayed. Contains <label>, <showfolders>, <type>, <showallversions>
<label>	Displays a label for the filter such as Folders or All Can contain a string or <nlsid>.
<objectfilter>.<showfolders>   <type>	To show folders only, set <showfolders> to true and <type> to null (no value). To show objects only, set <showfolders> to false and <type> to dm_sysobject. To show all, set <showfolders> to true and <type> to dm_sysobject. <type> can take any value that is a valid Documentum type.
<objectfilter>.<showallversions>	To add a versions column, set <showallversions> to true.
<columns>. <loadinvisibleattribute>	Uncomment this element and set to true to get invisible attribute values for use in your component. The invisible attributes can then be passed by configuring a column in the <columns> element. See <i>Web Development Kit and Client Applications Development Guide</i> for details.
<columns><column>	Specifies columns to show or hide
<column><attribute>	<attribute> sets the attribute to be displayed in the column.
<attribute> <label>	The value of <label> sets a label for the column.
<column><attribute> <visible>	Set <visible> to true to show the column.
<dragdrop>	Supports the drag and drop feature.

<sourceactions>	Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.
<dataproviders>	Contains one or more <dataprovider> elements.
<dataprovider>	Contains a <format> element and a <provider> element.
<dataprovider>.<format>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<dataprovider>.<provider>	Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

## Usage

The doclist and drilldown components have a default set of standard sysobject attributes that are displayed and formatted in columns. You can also configure columns to be displayed dynamically based on other attributes. See *Web Development Kit and Client Applications Development Guide* for more information.

# docprofileadd

**Purpose** Adds a work queue document profile.

## Configuration File

webcomponent\config\library\queuemanagement\docprofileadd\_component.xml

**Scope**

All.

**Parameters**

---

objectId	(Optional) Not used.
----------	----------------------

---

## docprofileattributes

**Purpose** Edits the attributes of a work queue document profile.

**Configuration File**

webcomponent\config\library\queuemanagement\docprofileattributes\_component.xml

**Scope**

All.

**Caller**

This component is called by the actions [attributes](#) (type `dmc_workqueue_doc_profile`), [page 237](#), [newWorkQueueDocProfile](#) (type `dmc_workqueue_doc_profile`), [page 352](#), and [properties](#) (type `dmc_workqueue_doc_profile`), [page 371](#), and displayed in the component [workqueuedocprofileproperties](#), [page 1153](#).

**Parameters**

---

objectId	(Required) ID of the document profile object for which you want to set attributes.
----------	--

---

## docprofilelist

**Purpose** Displays a list of the available queue management document profiles.

### Configuration File

webcomponent\config\library\queuemanagement\list\docprofilelist\_component.xml

### Scope

All.

## docsizevalueformattertest

**Purpose** One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

### Configuration File

wdk\config\fxtest\_component.xml

### Scope

All.

## documentlocatorcontainer

**Purpose** Extends the container [locatorcontainer](#), page 839 and contains locator components: [alldocumentlocator](#), page 529, [documentsubscriptionlocator](#), page 689, [mydocumentlocator](#), page 850, and [recentdocumentlocator](#), page 922.



## Configuration File

webcomponent\config\library\locator\documentlocators\_component.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The logical (base) type of the selectable type
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

# documentsubscriptionlocator

## Purpose

Extends the component [subscriptionlocator, page 1024](#) and locates objects of type dm\_document to which the user has subscribed.

## Configuration File

webcomponent\config\library\locator\documentlocators\_component.xml

## Parent Definition

subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

---

<objecttype>	The (logical) base type of the selectable type
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<privatecabinet_visible>	Boolean: Set to true to show private cabinets that are not owned by the session user.

---

<allversionsvisible>	Boolean: Set to true to show all versions.
<views>	See <a href="#">objectlocator, page 886</a> for information on configuring views.

---

## download

**Purpose** The download component calls a servlet that forces download of an object to the browser.

### Configuration File

webcomponent\config\library\httpview\download\_action\_component.xml

### Scope

All.

### Parameters

---

docId	ID of the object to download
servletAction	Type of action. Valid values: view   download

---

## downloadcontentcontainer (type dm\_sysobject)

**Purpose** Container for components that enable downloading content objects.

### Configuration File

webcomponent\config\library\contenttransfer\view\viewcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [downloadcontent](#) (type `dm_sysobject`), page 281 and displays the component [view](#) (type `dm_sysobject`), page 1104.

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

# dql

## Purpose

The `dql` component edits and executes queries to a repository. This component is nested to from the Webtop about component.

## Configuration File

`webcomponent\config\admin\utilities\dql_component.xml`

---

**Scope**

All.

## dqleditor

**Purpose** The dqleditor component edits and executes queries to a repository.

**Configuration File**

webcomponent\config\library\dqleditor\dqleditor\_component.xml

**Scope**

All.

**Parameters**

---

query	(Optional) DQL query
-------	----------------------

---

## dqlsavesearchdelegate

**Purpose** Calls the WDK 5.2.x savesearch component. You can use this component to call the WDK 5.2.x savesearch. Alternatively, you can call the new savesearch component without the queryId parameter to use the 5.2.x savesearch component.

**Configuration File**

webcomponent\config\library\savesearch\savesearchex\dqlsavesearchdelegate\_component.xml

**Parent Definition**

save:search:webcomponent/config/library/save:search/save:search\_component.xml

**Scope**

All.

## dqlsearchdelegate

**Purpose**

Calls the WDK 5.2.5 search component. You can use this component to call the WDK 5.2.x5 search. Alternatively, you can call the new search component with a null queryType parameter to use the 5.2.x search component.

**Configuration File**

webcomponent\config\library\search\searchex\dqlsearchdelegate\_component.xml

**Parent Definition**

search:webcomponent/config/library/search/search\_component.xml

**Scope**

All.

**Parameters**

These parameters are inherited from search:webcomponent/config/library/search/search\_component.xml.

---

query	(Required) The query string on which to search
queryType	(Optional) The type of object to be returned by the search

---

ranking	(Optional) Boolean: Set to true to display the ranking of search results.
useColumnsFromQuery	(Optional) Set to true to display results of the columns specified in the query rather than the columns configured in the component definition.

---

## drilldown

**Purpose** Displays a folder frame and a folder contents frame. The content pane supports filters similar to those in the doclist component. The user navigates by clicking on a folder or on a folder name in the breadcrumb control. The drilldown component has an extended version in Webtop called streamlineview.

### Configuration File

webcomponent\config\navigation\drilldown\drilldown\_component.xml

### Scope

All.

### Parameters

---

folderId	(Optional) Specifies the ID of the folder in which to start browsing
folderPath	(Optional) Specifies the path to the folder in which to start browsing. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder
home_component	(Optional) In Portal application, specifies portal component that calls this component. Used to construct a breadcrumb back to the portal.

home_component_args	(Optional) In Portal application, specifies arguments that are passed to the portal component that calls this component when the user clicks the breadcrumb to go back.
isolate	(Optional) Set to true to disable clicking in the path to the drilldown component
showAllVersions	(Optional) Set to true to show all versions of the object

---

## Elements

---

<showfilters>	Set to true to show the objectfilters drop-down control
<objectfilters>	Contains filters that define which objects should be shown in the objects selection list.
<objectfilter>	Specifies a filter for the items that are displayed. Contains <label>, <showfolders>, <type>, <showallversions>
<label>	Displays a label for the filter such as Folders or All Can contain a string or <nlsid>.
<objectfilter>.<showfolders>   <type>	To show folders only, set <showfolders> to true and <type> to null (no value). To show objects only, set <showfolders> to false and <type> to dm_sysobject. To show all, set <showfolders> to true and <type> to dm_sysobject. <type> can take any value that is a valid Documentum type.
<objectfilter>.<showallversions>	To add a versions column, set <showallversions> to true.
<columns>	For information on configuring columns, see <a href="#">Usage, page 686</a> .



---

<code>&lt;columns&gt;.</code> <code>&lt;loadinvisibleattribute&gt;</code>	Uncomment this element and set to true to get invisible attribute values for use in your component. The invisible attributes can then be passed by configuring a column in the <code>&lt;columns&gt;</code> element. See <i>Web Development Kit and Client Applications Development Guide</i> for details.
<code>&lt;columns&gt;&lt;column&gt;</code> <code>&lt;column&gt;&lt;attribute&gt;</code>	Specifies columns to show or hide <code>&lt;attribute&gt;</code> sets the attribute to be displayed in the column.
<code>&lt;attribute&gt;</code> <code>&lt;label&gt;</code>	The value of <code>&lt;label&gt;</code> sets a label for the column.
<code>&lt;column&gt;&lt;attribute&gt;</code> <code>&lt;visible&gt;</code>	Set <code>&lt;visible&gt;</code> to true to show the column.
<code>&lt;dragdrop&gt;</code>	Supports the drag and drop feature.
<code>&lt;sourceactions&gt;</code>	Contains zero or more <code>&lt;sourceaction&gt;</code> classes that support drag and drop actions on sources in the component.
<code>&lt;dataproviders&gt;</code>	Contains one or more <code>&lt;dataprovider&gt;</code> elements.
<code>&lt;dataprovider&gt;</code>	Contains a <code>&lt;format&gt;</code> element and a <code>&lt;provider&gt;</code> element.
<code>&lt;dataprovider&gt;.&lt;format&gt;</code>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<code>&lt;dataprovider&gt;.&lt;provider&gt;</code>	Fully qualified class name for the class that implements <code>IDragDropDataProvider</code> and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

# drl

**Purpose** Opens a DRL object (document resource locator) and prompts the user to view or edit it. You can also send an inbox item as a DRL. If the DRL is to an inbox item, the DRL will open the user's inbox, and the task manager will open the inbox item. The login component is invoked unless the user has a session or a default account is configured for anonymous access. The appropriate repository is derived from the object ID.

## Configuration File

wdk\config\drl\_component.xml

## Scope

All.

## Parameters

---

format	(Optional) If the user can edit and view the passed object, a user interface is presented to allow the user to choose the action. If no format is specified for the action, the object's native format is used.
objectId	(Required) ID of the object identified by the DRL.
versionLabel	(Optional) Any labelled version can be passed, such as CURRENT. If no value is set for versionLabel and the object's version is not current, an open current checkbox is shown.

---

## Elements

<defaultaccounts>	Contains anonymous access login information in one or more <defaultaccount> elements
<filter docbase= 'docbasename'>	Optional filter to specify the account for a single repository. Set the repository attribute on this element to the repository name.
<defaultaccount>	Sets up anonymous access for virtual links.
<username>	Name of a user in the target repository
<password>	Password for the anonymous user. Can be encrypted with the Trusted Authenticator Tool. Refer to <i>Web Development Kit and Client Applications Development Guide</i> for instructions.
<domain>	Optional network domain name for login to the repository
<logoffoncomplete>	Set to true to log the default account off and close the window after the DRL action is performed

## Usage

The DRL component displays **View** and **Edit** buttons. The event handlers for these buttons call the [drlview](#) (type `dm_sysobject`), [page 284](#) and [drledit](#) (type `dm_sysobject`), [page 282](#) actions, respectively. Requires the Nautilus File Manager on Mozilla clients.

DRLs have the following syntax: `http://host_name:port/app_name/component/drl?objectId=xxx`. For example, the following DRL specifies an object and the format:

```
http://localhost:8080/webtop/component/drl?
  objectId=090056ce8002d6ef&format=msw8
```

**Note:** The format argument should not be supplied if the object has no content.

The DRL component supports the following anonymous access functionality:

- Virtual links and DRLs have independent sets of stored credentials for anonymous access.
- If the anonymous user account user does not have permission to perform the requested DRL operation (view or edit), a login dialog is presented.
- Anonymous credentials are used only for content DRLs, not DRLs to folders or inbox.
- User is logged out of the application after the anonymous DRL operation has been performed. (Configurable as the <logoffoncomplete> element in the drl component definition.)

You can construct a DRL in your component class by importing the `DRLComponent` class and calling `constructDRL(String strObjectId, String strVersionLabel, String strFormat, Component component)`. The version label and format parameters are optional. The following example initializes a bookmark control in component JSP page:

```
BookmarkLink bookmark = (BookmarkLink)getControl(  
    "bookmark", BookmarkLink.class);  
bookmark.setHREF(DRLComponent.constructDRL(  
    strObjectId, null, null, this));
```

## drlauthenticate

**Purpose** This component opens an object, prompting the user to select view or edit mode. Note that the opened object may differ if a version label is passed (for example, `CURRENT`). The login component is invoked if necessary.

### Configuration File

`wdk\config\drlauthenticate_component.xml`

### Parent Definition

`drl:wdk/config/drl_component.xml`

### Scope

All.

### Parameters

These parameters are inherited from `drl:wdk/config/drl_component.xml`.

---

format	(Optional) (Required) The format of the object to display
objectId	(Required) (Required) The object ID of the object to display
versionLabel	(Optional) The label identifying which version of the object to display

---

## edit (type dm\_sysobject, version 5.2.5)

**Purpose** This is the WDK 5.2.5 edit component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to edit objects in the repository.

### Configuration File

webcomponent\config\library\edit\edit\_component.xml

### Scope

---

type	dm_sysobject
version	5.2.5

---

### Caller

Cannot be called directly. Must be extended.

## Parameters

---

isReference	(Optional) Boolean: Indicates whether the object is a reference
lockOwner	(Optional) ID of the owner of a document that is currently locked
objectId	(Required) The ID of the object to check out

---

# edit (type dm\_sysobject)

## Purpose

This is the WDK 5.3 edit component. It edits objects in the repository. This component must run within the editcontainer component. The edit component is defined for dm\_sysobject scope. Editing is excluded for dm\_folder objects by the notdefined attribute on the component element. The container extends [contentxfercontainer](#), page 653.

## Configuration File

webcomponent\config\library\contenttransfer\edit\edit\_component.xml

## Parent Definition

checkout:webcomponent/config/library/contenttransfer/checkout/checkout\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [editfile](#) (type dm\_sysobject), page 292 and displayed in the component [editcontainer](#) (type dm\_sysobject), page 705.

## Parameters

These parameters are inherited from checkout:webcomponent/config/library/contenttransfer/checkout/checkout\_component.xml.

isIgnoreDescendents	(Optional) Boolean value that indicates whether or not to check out descendants in a virtual document.
isLockOnly	(Optional) Boolean value that indicates whether or not to check out objects that are locked.
nodeId	(Optional) r_object_id of a virtual document node to check out.
objectId	(Required) The ID of the object to check out.
vdmRootObjectId	(Optional) r_object_id of the virtual document root.

## Elements

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<processor-class>	Specifies the fully qualified class name for the class that implements a processor class; in this case a class that starts an associated application on the local machine. Value is: com.documentum.web.contentxfer.impl.EditProcessor

# edit (type dmc\_notepage)

**Purpose** This component is used to edit dmc\_notepage objects. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\notepage\editnotepage\_component.xml

### Scope

---

type	dmc_notepage
------	--------------

---

### Caller

This component is called by the action .

### Parameters

---

objectId	(Required) Object ID of the note
----------	----------------------------------

---

## editcontainer (type dm\_sysobject, version 5.2.5)

### Purpose

This is the WDK 5.2.5 editcontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 container used to display edit components.

### Configuration File

webcomponent\config\library\edit\editcontainer\_component.xml

### Parent Definition

contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml



**Scope**

---

type	dm_sysobject
version	5.2.5

---

**Parameters**

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

**Elements**

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
---------------------------	--

---

## editcontainer (type dm\_sysobject)

**Purpose** This is the WDK 5.3 editcontainer component. It is the container for components that enable checking out objects and opening them for editing in an associated application.

**Configuration File**

webcomponent\config\library\contenttransfer\edit\editcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [editfile \(type dm\\_sysobject\), page 292](#) and displays the component [edit \(type dm\\_sysobject\), page 702](#).

## Parameters

These parameters are inherited from combocontainer:wdk/config/combocontainer\_component.xml.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
<ucfrequired>	If this element is present, then the component requires UCF to be installed.

<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;service-class&gt;</code>	Specifies the fully qualified class name for the class that implements the service. Value is: <code>com.documentum.web.contentxfer.impl.EditService</code>
<code>&lt;transport-class&gt;</code>	Specifies the fully qualified class name for the class that implements the transport; in this case, UCF. Value is: <code>com.documentum.web.contentxfer.ucf.UcfContentTransport</code>
<code>&lt;post-processor-class&gt;</code>	Specifies the fully qualified class name for the class that implements a post-processor class; in this case a class that starts an associated application on the local machine. Value is: <code>com.documentum.webcomponent.library.contenttransfer.edit.LaunchEditor</code>

---

## editprocessorauthentication

**Purpose** Displays a page in the Properties component for a work queue processor, enabling the user to set the processor's skill level and available document profiles. The component does not save the work queue processor profile, but passes any changes back to the calling component.

## Configuration File

webcomponent\config\library\queuemanagement\editprocessorauthentication\_  
component.xml

## Scope

All.

## Parameters

---

docprofile_enable	(Optional) Boolean: set to true to enable the user to change the work queue document profile assigned to the processor. Default is false.
docprofile_name	(Optional) Name of the work queue document profile.
skill	(Optional) Skill level of the user for this work queue. Valid values are the skill level strings from the data dictionary.
username	(Required) User name of the work queue processor.
wq_name	(Required) Name of the work queue.

---

# editstartwfnote (type dmi\_package)

**Purpose** Adds, edits, and removes attachment notes from dmi\_package objects during the workflow initialization process.

## Configuration File

webcomponent\config\library\workflow\editnote\editstartwfnote\_component.xml

**Scope**

type	dmi_package
------	-------------

**Caller**

This component is invoked from the action [editstartwfnote \(type dmi\\_package\)](#), page 296 in the component [UIstartwfcomments](#), page 1016 and displayed in the component [combocontainer](#), page 644.

**Parameters**

operationType	(Required) Type of operations to perform on a note. Valid values: add   edit   remove.
packageName	(Required) Name of the workflow package
processId	(Required) ID of the workflow process

**edittasknote (type dmi\_package)****Purpose**

The edittasknote component adds, edits, and removes attachment notes. This component is defined for dmi\_package objects in edittasknote\_component.xml.

**Configuration File**

webcomponent\config\library\workflow\editnote\edittasknote\_component.xml

**Scope**

type	dmi_package
------	-------------

## Caller

This component is called by the action [edittasknote](#) (type `dmi_package`), page 297 and displayed in the component [combocontainer](#), page 644.

## Parameters

---

<code>operationType</code>	(Required) Type of operations to perform on a note. Valid values: add   edit   remove.
<code>packageName</code>	(Required) Name of the workflow package
<code>queueItemId</code>	(Required) ID of the queue item with which the note is associated.

---

# editxforms

**Purpose** Extends `dialogcontainer` to edit forms. Contains the component [xforms](#), page 1159. If the user does not have the `form_user` role, the form will be displayed as XML.

## Configuration File

`webcomponent\config\xforms\editxforms_component.xml`

## Parent Definition

`dialogcontainer:wdk/config/dialogcontainer_component.xml`

## Scope

All.

## Caller

This component is called by the action [editfile](#) (type `dm_sysobject`), page 292 when the selected object is a form.

## Parameters

deleteOnCancel	(Optional) Specifies whether the new form instance should be deleted when the user clicks Cancel. The default is false; that is, the new form instance is not deleted.
isNew	(Optional) Specifies whether the form is new. This parameter is set by the NewXFormsContainer. Because XFormsComponent can be extended, you might want to change this behavior.
objectId	(Required) ID of the selected form.
readOnly	(Optional) A value of true renders the form as readonly (view mode).
templateId	(Optional) ID of the associated form template. If this parameter is not specified, the default form template associated with the form instance is used.

# editxformsproperties

**Purpose** This component is used to edit a form component definition.

## Configuration File

webcomponent\config\xforms\editxformsproperties\_component.xml

## Parent Definition

editxforms:webcomponent/config/xforms/editxforms\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from editxforms:webcomponent/config/xforms/editxforms\_component.xml.

---

deleteOnCancel	(Optional) Specifies whether the new form instance should be deleted when the user clicks Cancel. The default is false; that is, the new form instance is not deleted. This parameter is set by the NewXFormsContainer. Because XFormsComponent can be extended, you might want to change this behavior.
isNew	(Optional) Specifies whether the form is new.
objectId	(Required)ID of the selected form
readOnly	(Optional) Boolean: Set to true to render the form as readonly (view mode).
templateId	(Optional) ID of the associated form template

---

## embeddedtopic

**Purpose** Enables a user to add, edit, or delete a topic in an embedded view. The embedded view appears at the bottom of dm\_folder and dmc\_notepage objects. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\discussion\topicembedded\_component.xml

### Scope

All.



## Parameters

folderPath	(Optional) Folder path of the parent topic.
nodeIds	(Optional) List of dot-separated object IDs used to build an absolute location to display the embedded topic
objectId	(Required) ID of the parent topic.
type	(Optional) Object type of the topic parent

## Elements

<comments>	Contains one or more <comment> elements
<comment>	Contains <attribute> and <visible> elements. Specifies comment attributes to include in the TopicResultSet.
<attribute>	Sets the name of the attribute to be included in the result set
<visible>	Set to true to make the attribute visible

In the following example, the attributes `r_modify_date` and `r_modifier` will be available for databound controls within the `commentrow` control.

In addition to the required attributes that make up a topic, attributes such as `modify_date` and/or `modifier` can be specified. Include the name in the attribute element and set visibility to `true` so that the attribute will be available for databound controls contained within a `commentrow`. Invalid attributes will throw a runtime exception.

```
<comments>
  <comment>
    r_modify_date</attribute>
    true</visible>
  </comment>
  <comment>
    <attribute>r_modifier</attribute>
    <visible>true</visible>
  </comment>
</comments>
```

## Usage

Use the `componentinclude` tag to embed the component at the bottom of parent page. The parameter `objectId` is that of the parent page and is used to acquire the existing topic or is used in the creation of a topic upon entry of the first comment.

```
<dmfx:componentinclude name="showtopic" component="embeddedtopic">
  <dmfx:argument name="objectId" contextvalue="objectId">
</dmfx:componentinclude>
```

# embeddedtopic (type dmc\_room)

**Purpose** Displays the topic for a folder or note at the bottom of the parent objects page in a room. Requires Collaborative Edition. A user adds, edits or deletes a comment using the controls rendered by the components page.

## Configuration File

`webcomponent\config\library\rooms\topicembedded_component.xml`

## Parent Definition

`embeddedtopic:webcomponent/config/library/discussion/topicembedded_component.xml`

## Scope

type	dmc_room
------	----------

## Parameters

These parameters are inherited from `embeddedtopic:webcomponent/config/library/discussion/topicembedded_component.xml`.

---

folderPath	(Optional) The folder path of the topic's parent.
nodeIds	(Optional) List of dot-separated object Ids are used to build an absolute location to display.
objectId	(Required) The object id of the parent object.
type	(Optional) Parent object type (dm_folder).

---

## errormessage

**Purpose** The errormessage component displays a fatal error message, which appears in a popup dialog box if and when the error message handler receives a fatal error.

### Configuration File

wdk\config\errormessage\_component.xml

### Scope

All.

### Elements

---

<displaystacktrace>	Set to true to display the stack trace in an error message.
---------------------	---

---

## export (type dm\_sysobject, version 5.2.5)

**Purpose** This is the WDK 5.2.5 export component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It must run within the 5.2.5 export container.

### Configuration File

webcomponent\config\library\export\export\_component.xml

### Scope

---

type	dm_sysobject
version	5.2.5

---

### Parameters

---

contentType	(Optional) File format to use when opening the object (passed to the operating system).
objectId	(Required) ID of the object to be exported(Required) ID of the object to be exported

---

## export (type dm\_sysobject)

**Purpose** This is the WDK 5.3 export component. Exports objects from the repository to the client file system. This component must run within the 5.3 component exportcontainer.

## Configuration File

webcomponent\config\library\contenttransfer\export\export\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [export \(type dm\\_sysobject\), page 298](#) and displayed in the component [exportcontainer \(type dm\\_sysobject\), page 721](#).

## Parameters

---

contentType	(Optional) (Required) File format to use when opening the object (passed to the operating system).
isIgnoreDescendents	(Optional) Specifies whether descendants of a virtual document should be ignored.
nodeId	(Optional) r_object_id of a virtual document node to edit.
objectId	(Required) (Required) ID of the object to be exported(Required) ID of the object to be exported
pageModifier	(Optional) A valid value for the page_modifier attribute of the dmr_content object. You should not pass in your own value for PDF renditions.
vdmRootObjectId	(Optional) r_object_id of the virtual document root.

---

## Elements

---

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.ExportProcesso
<init-controls>	Initializes a value for the downloadDesc-Checkbox

---

## export (type dmr\_content, version 5.2.5)

**Purpose** This is a WDK 5.2.5 export component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is a WDK 5.2.5 component used to export dm\_sysobjects ( type dmr\_content) and dmr\_content objects from the repository to the client file system. This component must run within the component exportcontainer. Export of multiple objects creates one export container and an inner export component instance for each object.

### Configuration File

webcomponent\config\library\export\export\_component.xml

### Scope

---

type	dmr_content
version	5.2.5

---

### Caller

This component is called by the component [exportcontainer \(type dmr\\_content, version 5.2.5\)](#), page 723.

---

## Parameters

---

contentType	(Optional) File format to use when opening the object (passed to the operating system).
objectId	(Required) ID of the object to be exported.

---

## export (type dmr\_content)

**Purpose** This is the WDK 5.3 export component. It is used to export dm\_sysobjects and dmr\_content objects from the repository to the client file system. This component must run within the component exportcontainer. Export of multiple objects creates one export container and an inner export component instance for each object.

## Configuration File

webcomponent\config\library\contenttransfer\export\export\_component.xml

## Scope

---

type	dmr_content
------	-------------

---

## Caller

This component is called by the action [exporthandition \(type dmr\\_content\)](#), page 299 and is displayed in the component [exportcontainer \(type dmr\\_content\)](#), page 725.

## Parameters

---

contentType	(Optional) (Required) File format to use when opening the object (passed to the operating system).
objectId	(Required) ID of the object to be exported (Required) ID of the object to be exported
pageModifier	(Optional) A valid value for the page_modifier attribute of the dmr_content object. You should not pass in your own value for PDF renditions.

---

## Elements

---

<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.ExportProcessor
<init-controls>	Initializes a value for the downloadDesc-Checkbox

---

# exportcontainer (type dm\_sysobject, version 5.2.5)

## Purpose

This is the WDK 5.2.5 exportcontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to display the component used to export dm\_sysobject and dmr\_content objects from the repository to the client file system. Export of multiple objects creates one export container and an inner export component instance for each object.

## Configuration File

webcomponent\config\library\export\exportcontainer\_component.xml



## Parent Definition

contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml

## Scope

---

type	dm_sysobject
version	5.2.5

---

## Caller

Cannot be called directly. Must be extended.

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

# exportcontainer (type dm\_sysobject)

## Purpose

This is a WDK 5.3 exportcontainer component. It is used to export dm\_sysobjects and dmr\_content objects from the repository to the client file system. Export of multiple objects creates one export container and an inner export component instance for each object.

## Configuration File

webcomponent\config\library\contenttransfer\export\exportcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [export \(type dm\\_sysobject\), page 298](#) and displays the component [export \(type dm\\_sysobject\), page 716](#).

## Parameters

These parameters are inherited from combocontainer:wdk/config/combocontainer\_component.xml.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<code>&lt;ucfrequired&gt;</code>	If this element is present, then the component requires UCF to be installed.
<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;service-class&gt;</code>	Specifies the fully qualified class name for the class that implements the service. Value is: <code>com.documentum.web.contentxfer.impl.ExportService</code>
<code>&lt;transport-class&gt;</code>	Specifies the fully qualified class name for the class that implements the transport; in this case, UCF. Value is: <code>com.documentum.web.contentxfer.ucf.UcfContentTransport</code>

---

## exportcontainer (type dmr\_content, version 5.2.5)

### Purpose

This is the WDK 5.2.5 exportcontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to display the export component to export `dm_sysobject` and `dmr_content` objects from the repository to the client file system. This component must run within the component

exportcontainer. Export of multiple objects creates one export container and an inner export component instance for each object.

## Configuration File

webcomponent\config\library\export\exportcontainer\_component.xml

## Parent Definition

contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml

## Scope

---

type	dmr_content
version	5.2.5

---

## Caller

Cannot be called directly. Must be extended.

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: 5.2.5
---------------------------	---

## exportcontainer (type dmr\_content)

**Purpose** This is a WDK 5.3 exportcontainer component. It is used to export dm\_sysobjects and dmr\_content objects from the repository to the client file system. This component must run within the component exportcontainer. Export of multiple objects creates one export container and an inner export component instance for each object.

### Configuration File

webcomponent\config\library\contenttransfer\export\exportcontainer\_component.xml

### Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

### Scope

type	dmr_content
------	-------------

### Caller

This component is called by the action [exportrendition \(type dmr\\_content\), page 299](#), and displays the component [export \(type dmr\\_content\), page 719](#).

## Parameters

These parameters are inherited from combocontainer:wdk/config/combocontainer\_component.xml.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<service-class>	Specifies the fully qualified class name for the class that implements the service. Value is: com.documentum.web.contentxfer.impl.ExportService
<transport-class>	Specifies the fully qualified class name for the class that implements the transport; in this case, UCF. Value is: com.documentum.web.contentxfer.ucf.UcfContentTransport

---

# extendedaddaccessor

**Purpose** Adds or edits individual accessors and accessor groups, including their permissions, for inclusion in access control lists. Requires Trusted Content Services for accessor groups.

## Configuration File

webcomponent\config\admin\acl\extendedaddaccessor\_component.xml

## Scope

All.

## Caller

This component is called by the action [editaceaction \(type dm\\_ace\)](#), page 286.

## Parameters

---

accessorName	(Required) The name of the individual or group to be added.
basicpermit	(Optional) The name of the basic permit for the accessor. Valid values are 1 (none), 2 (browse), 3 (read), 4 (relate), 5 (version), 6 (write), 7 (delete). Default = 3 (read)
extpermit	(Optional) The extended permission for the accessor. Valid values are EXECUTE_PROCEDURE_BIT, CHANGE_LOCATION_BIT, CHANGE_STATE_BIT, CHANGE_PERMISSIONS_BIT, TAKE_OWNERSHIP_BIT. Default = 0 (no extended permissions)
extpermit_readonly	(Optional) Set to true to disable extended permission controls
objectName	(Optional) Accessor name passed from container
permit	(Optional) String representation of the permission that is being applied to the accessor
permitType	(Optional) Valid types of IDfPermit: DF_ACCESS_PERMIT=0DF_ EXTENDED_PERMIT = 1DF_ APPLICATION_PERMIT = 2DF_ACCESS_RESTRICTION = 3DF_EXTENDED_RESTRICTION = 4DF_APPLICATION_RESTRICTION = 5DF_REQUIRED_GROUP = 6DF_REQUIRED_GROUP_SET = 7

---

# externalresultimportcontainer

**Purpose** Container for components that enable importing local files into repositories.

## Configuration File

webcomponent\config\library\contenttransfer\importcontent\externalresultimportcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [importexternalresult](#) (type `dm_externalresult`), [page 317](#) and displays one or more instances of the component [import](#), [page 804](#).

## Parameters

---

component	(Required) Supplied by the action execution class). The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.
entryId	(Required) Internal key to the query results.
queryId	(Required) Internally used to identify the current search context

---



## Elements

<control-propagation>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<control-propagation>.<property name="enabled"/>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<control-propagation>.<property name="visible"/>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<service-class>	Specifies the fully qualified class name for the class that implements the service. Value is: com.documentum.web.contentxfer.impl.ImportService
<transport-class>	Specifies the fully qualified class name for the class that implements the transport; in this case, HTTP. Value is: com.documentum.web.contentxfer.http.HttpContentTransport

## fileplanlocator (entitlement recordsmanager)

**Purpose** Enables selection of a file plan in which to place a record. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\records\locator\fileplan\_locator\_component.xml

## Parent Definition

allfolderlocator:webcomponent/config/library/locator/folderlocators\_component.xml

## Scope

---

entitlement	recordsmanager
-------------	----------------

---

## Caller

Runs in the [fileplanlocatorcontainer \(entitlement recordsmanager\), page 730](#) container.

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

# fileplanlocatorcontainer (entitlement recordsmanager)

**Purpose** Container for the fileplanlocator component. Requires a Records Manager license.

## Configuration File

webcomponent\config\library\records\locator\fileplan\_locator\_component.xml

## Parent Definition

folderlocatorcontainer:webcomponent/config/library/locator/folderlocators\_component.xml

## Scope

---

entitlement	recordsmanager
-------------	----------------

---

## Caller

Runs the [fileplanlocator \(entitlement recordsmanager\), page 729](#), [fileplanmyobjectlocator \(entitlement recordsmanager\), page 732](#), and [fileplansubscriptionlocator \(entitlement recordsmanager\), page 733](#) component.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) The ID number of the folder in which to begin the search.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) The doctype in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## fileplanmyobjectlocator (entitlement recordsmanager)

**Purpose** Displays all file plan objects that are owned by the user. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\records\locator\fileplan\_locator\_component.xml

### Parent Definition

myobjectlocator:webcomponent/config/library/locator/myobjectlocator\_component.xml

### Scope

---

entitlement	recordsmanager
-------------	----------------

---

### Caller

Runs in the [fileplanlocatorcontainer \(entitlement recordsmanager\)](#), page 730 container.

### Parameters

These parameters are inherited from myobjectlocator:webcomponent/config/library/locator/myobjectlocator\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

# fileplansubscriptionlocator (entitlement recordsmanager)

**Purpose** Displays a user's subscribed file plans. Requires a Records Manager license.

## Configuration File

webcomponent\config\library\records\locator\fileplan\_locator\_component.xml

## Parent Definition

subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml

## Scope

entitlement	recordsmanager
-------------	----------------

## Caller

Runs in the [fileplanlocatorcontainer \(entitlement recordsmanager\)](#), page 730 container.

## Parameters

These parameters are inherited from subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml.

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## finishworkflowtask

**Purpose** The finishworkflowtask component completes non-manual workflow tasks (for which the user is not expected to specify the next task).

### Configuration File

webcomponent\config\library\workflow\taskmanager\finishworkflowtask\_  
component.xml

### Scope

All.

### Caller

This component is invoked from the taskmanager component.

### Parameters

---

objectId	(Required) ID of the workflow task
userCost	(Optional) Currency cost, such as dollars, euros, or other type of cost defined by customer. Value is stored as a double.
userTime	(Optional) Time in units defined by the customer. Value is stored as an integer.

---

## folderlocatorcontainer

**Purpose** Extends the container [locatorcontainer](#), page 839 and selects the folders to be searched. This container is used by [advsearch](#), page 522 and

contains [allfolderlocator](#), page 532, [foldersubscriptionlocator](#), page 736, and [recentfolderlocator](#), page 923.

## Configuration File

webcomponent\config\library\locator\folderlocators\_component.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

# foldersubscriptionlocator

**Purpose** Extends the component [subscriptionlocator, page 1024](#) and locates subscribed folders.

## Configuration File

webcomponent\config\library\locator\folderlocators\_component.xml

## Parent Definition

subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

The configurable elements for this component are inherited from [sysobjectlocator, page 1029](#).



# foldertree

**Purpose** The folder tree component provides basic folder navigation functionality. The user can browse cabinets and folders. Failover is enabled for foldertree. This component is extended by the Webtop browsertree component.

## Configuration File

```
webcomponent\config\navigation\foldertree\foldertree_component.xml
```

## Scope

All.

## Parameters

folderPath	(Optional) Specifies the path to the folder that serves as the base for navigation. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder
------------	---

## Usage

You can set the limit for number of folders to be displayed in the tree in the file `Environment.properties`, located in `/WEB-INF/classes/com/documentum/web/formext`. Larger numbers will be replaced with a More Folders link.

The folder tree component class `FolderTree` has a single method: `onInit(ArgumentList args)`. This method gets the folder path parameter if it is supplied as an argument, and sets the base for navigation to the supplied path or to the root path as supplied by the `DocbaseFolderTree` control. The control [doibasefoldertree](#), page 85 must be present in the foldertree component JSP page.

# formalrecordcombocontainer (entitlement recordsmanager)

**Purpose** Enables a user to perform multi-select operations on formal record objects. Requires a Records Manager license.

## Configuration File

webcomponent\config\library\records\declarerecord\formalrecordcombocontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

---

entitlement

recordsmanager

---

## Caller

Runs in the [fileplanlocatorcontainer \(entitlement recordsmanager\)](#), page 730 container.

## Parameters

These parameters are inherited from combocontainer:wdk/config/combocontainer\_component.xml.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

## format\_preferences

**Purpose** Enables the user to set format preferences for preferred renditions.

### Configuration File

webcomponent\config\environment\preferences\format\format\_preferences\_component.xml

### Scope

All.

## Elements

<base_type>	The base type to search from when building the list of docbase types.
<ucfrequired>	If this element is present, then the component requires UCF to be installed.

# formatattributes

**Purpose** Displays attributes of objects or creates new objects of type dm\_format. This component is contained within the container [formatproperties](#), page 741.

## Configuration File

webcomponent\config\admin\format\formatattributes\_component.xml

## Scope

All.

## Parameters

objectId	(Required) ID of the object for which properties are displayed. Set to newobject to create a new format.
----------	--

# formatlist

**Purpose** Displays a list of formats in the repository (objects of type dm\_format).

**Configuration File**

webcomponent\config\admin\format\formatlist\_component.xml

**Scope**

All.

**Caller**

This component is called from the component UI of [checkin \(type dm\\_sysobject\)](#), page 620 and [commitchanges](#), page 646.

## formatproperties

**Purpose**

Extends the container [adminpropertycontainer](#), page 515 and the attributes of format objects or creates a new format object. Contains the component [formatattributes](#), page 740.

**Configuration File**

webcomponent\config\admin\format\formatproperties\_component.xml

**Parent Definition**

adminpropertycontainer:webcomponent/config/admin/container/  
adminpropertycontainer\_component.xml

**Scope**

All.

**Caller**

This component is called by the actions [properties \(type dm\\_format\)](#), page 366 and [newformat \(type dm\\_format\)](#), page 343.

**Parameters**

---

objectId	(Required) ID of the object for which properties are displayed. Set to newobject to create a new format.
----------	--

---

**Elements**

---

<objecttype>	Type of object that is supported by the container (placeholder). Default = dm_format.
--------------	---

---

## forwardroutertask

**Purpose** Forwards router tasks. It is invoked from the taskmanager component.

**Configuration File**

webcomponent\config\library\workflow\taskmanager\forwardroutertask\_component.xml

**Scope**

All.

**Parameters**

---

objectId	(Required) ID of the router task
----------	----------------------------------

---

---

# forwardworkflowtask

**Purpose** Forwards manual workflow tasks. It is invoked from the taskmanager component.

## Configuration File

webcomponent\config\library\workflow\taskmanager\forwardworkflowtask\_  
component.xml

## Scope

All.

## Parameters

---

objectId	(Required) ID of the workflow task
----------	------------------------------------

---

# general\_preferences

**Purpose** Enables the user to select a UI theme and the startup location.

## Configuration File

webcomponent\config\environment\preferences\general\general\_preferences\_  
component.xml

## Scope

All.

## Caller

This component is called by the action [preferences, page 359](#) and displayed in the component [preferences, page 898](#).

# getcontent

## Purpose

Streams Documentum object content to the browser without the use of applets. Login is requested if the user has not already logged in. Where possible, the content is displayed within the browser and not with an external application.

## Configuration File

wdk\config\getcontent\_component.xml

## Scope

All.

## Parameters

---

format	(Optional) If a value is provided, the component attempts to stream the content in the specified format. If not, the object's native format is used.
objectId	(Required) ID of the selected object
pageModifier	(Optional) Specifies a user-defined attribute (page_modifier) value that distinguishes between renditions.
pageNumber	(Optional) Specifies the page number of a document

---



## Usage

The start page is specified as the URL (/wdk5-download) that is mapped to the streaming content servlet.

Following is an example of a URL to stream content:

```
/APP_HOME/component/getcontent?objectId=xxxxyyyyzzzzaaaa&format=ppt8
```

## govern (type dm\_folder)

**Purpose** Enables the user to specify whether governing should be applied to the children of a selected object to govern. Requires Collaborative Edition.

### Configuration File

```
webcomponent\config\library\rooms\govern_component.xml
```

### Scope

type	dm_folder
------	-----------

### Parameters

folderId	(Required) Object ID of the folder that contains the object specified by the objectId parameter
objectId	(Required) ID of the object to govern
sourceFolderId	(Optional) ID of the folder that will be added to the governed folder or room

## groupattributes

**Purpose** Creates a new user profile and edits existing user groups.

### Configuration File

webcomponent\config\admin\group\groupattributes\_component.xml

### Scope

All.

### Parameters

---

objectId	(Required) The ID of the selected group.
objectname	(Required) The name of the selected group.

---

## groupdelete

**Purpose** Deletes an existing user group. Warns the administrator about the repercussions of performing this task and offers alternative solutions.

### Configuration File

webcomponent\config\admin\group\groupdelete\_component.xml

### Scope

All.

---

## Parameters

---

objectId	(Required) The ID of the group to delete.
objectname	(Required) The name of the group to delete.

---

# grouplist

**Purpose** Displays a list of all user groups present in the repository. It provides the ability to drill down in a group to view all the users and groups within the group.

## Configuration File

webcomponent\config\admin\group\grouplist\_component.xml

## Scope

All.

## Usage

To preserve repository consistency, do not remove groups from the repository. Remove all members of the group and leave the empty group in the repository. When you delete a group, the server does not remove the group's name from objects in the repository, such as ACLs and other groups. You can delete a group, and then create a group with the same name. If you create a new group with the same name as a deleted group, the new group inherits the group memberships and object permissions belonging to the deleted group.

# grouplocator

**Purpose** Displays a list of groups in the repository.

## Configuration File

webcomponent\config\library\locator\grouplocator\_component.xml

## Parent Definition

userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml.

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of groups in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements

---

<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.

<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## grouplocatorcontainer

**Purpose** Extends the container [locatorcontainer](#), page 839 and contains one or more [grouplocator](#), page 747.

### Configuration File

webcomponent\config\library\locator\grouplocator\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) ID of the repository in which to perform the search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

# groupproperties

**Purpose** Extends the container [adminpropertycontainer, page 515](#) and displays properties components for the selected group. It overrides some of the behavior from the adminpropertycontainer component. and contains [groupattributes, page 746](#).

## Configuration File

webcomponent\config\admin\group\groupproperties\_component.xml

## Parent Definition

adminpropertycontainer:webcomponent/config/admin/container/  
adminpropertycontainer\_component.xml

## Scope

All.

## Parameters

---

objectId	(Required) The ID of the group to delete.
objectname	(Required) The name of the group to delete.

---

## Elements

---

<contains>	One or more <component> elements specify components are displayed by the formatproperties container component. Default = groupproperties.
<objecttype>	The type of the group object. Default = dm_group.

---

# grouprename

**Purpose** Reassigns an existing group. The administrator can reassign all the objects owned by the current group and generate an exhaustive report.

## Configuration File

webcomponent\config\admin\group\grouprename\_component.xml



**Scope**

All.

**Parameters**


---

objectId	(Required) The ID of the group to delete.
objectname	(Required) The name of the group to delete.

---

## groupwhereused

**Purpose**

Lists the locations where the selected group is being used; for example, it lists all sysobjects owned by the group, acls in which the group exists, and workflows for which the group is a designated performer.

**Configuration File**

```
webcomponent\config\admin\group\groupwhereused_component.xml
```

**Scope**

All.

**Parameters**


---

objectId	(Required) Object ID of the group to locate.
objectname	(Required) Object name of the group to locate.

---

## halttask (type dm\_task)

**Purpose** Halts (pauses) a task in a workflow. This component is defined for dm\_task objects.

### Configuration File

webcomponent\config\library\workflow\halt\halttask\_component.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

This component is launched by the action [halttask \(type dm\\_task\), page 312](#) and displayed in the component [combocontainer, page 644](#).

### Parameters

---

objectId	(Required) ID of the work item.
objectName	(Required) Task name.

---

## haltworkflow (type dm\_workflow)

**Purpose** Halts (pauses) workflow objects. This component is defined for dm\_workflow objects.

### Configuration File

webcomponent\config\library\workflow\halt\haltworkflow\_component.xml

**Scope**

---

type	dm_workflow
------	-------------

---

**Parameters**

---

objectId	(Required) ID of the workflow
----------	-------------------------------

---

## help-index

**Purpose** Help files are mapped in the help-index component definition. The help\_component.xml configuration file stores a list of reference entries for HTML help pages. This definition is overridden by the Webtop help component, which has entries for help topics.

**Configuration File**

webcomponent\config\help\help\_component.xml

**Scope**

All.

## historicalactivityreportresults

**Purpose** Displays the results of a historical activity report query.

**Configuration File**

webcomponent\config\library\workflow\historicalreport\activityresults\_component.xml

**Scope**

All.

**Parameters**

---

process_id	(Required) ID of the workflow template (process) on which to report
query	(Required) When the querytype is string, this value is the search string. If the querytype is not string, this value is the query string.
title	(Optional) Title to be displayed at the top of the report

---

## historicalprocessreport

**Purpose** This component displays a workflow process historical report.

**Configuration File**

webcomponent\config\library\workflow\historicalreport\processreport\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [historicalprocessreport](#), page 313 and displayed in the component [historicalprocessreportcontainer](#), page 757.

---

# historicalprocessreportcontainer

**Purpose** This component is used to display historical workflow reports for processes and performers.

## Configuration File

webcomponent\config\library\workflow\historicalreport\processreportcontainer\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [historicalprocessreport, page 313](#) and displays the component [historicalprocessreport, page 756](#).

## Parameters

---

component	(Optional) ID of the component to be displayed
-----------	--

---

# historicalprocessreportresults

**Purpose** This component is used to display reports on workflow processes that have been completed.

## Configuration File

webcomponent\config\library\workflow\historicalreport\processresults\_component.xml

## Scope

All.

## Parameters

---

query	(Required) When the querytype is string, this value is the search string. If the querytype is not string, this value is the query string.
title	(Optional) The title that will appear at the top of the report.

---

# historicalreportcolumnselector

**Purpose** This component provides UI and behavior for selecting the columns and column order for view components such as DocList and Drilldown.

## Configuration File

webcomponent\config\library\workflow\historicalreport\columnselector\_component.xml

## Parent Definition

columnselector:webcomponent/config/environment/preferences/display/columnselector\_component.xml

**Scope**

All.

**Parameters**


---

baseType	(Optional) Specifies the repository types, in addition to the 'default' option, to populate the attribute selector's doctype drop-down list.
columns	(Required) A serialized list of items to populate the available items list box. See multiselector control for details on the list format.
defaultSelectedColumns	(Optional) A serialized list of items to populate the selected items list box before the user performs an initial selection.
selectedColumns	(Optional) A serialized list of items to populate the selected columns list box.

---

**historicaluserreport**

**Purpose** This component displays a workflow process historical report for a performer.

**Configuration File**

```
webcomponent\config\library\workflow\historicalreport\userreport_component.xml
```

**Scope**

All.

## Caller

This component is called by the action [historicaluserreport, page 314](#) and displayed in the component [historicaluserreportcontainer, page 760](#).

# historicaluserreportcontainer

**Purpose** This component is a container for components that display historical user reports.

## Configuration File

webcomponent\config\library\workflow\historicalreport\userreportcontainer\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [historicaluserreport, page 314](#) and displays the component [historicaluserreport, page 759](#).

## Parameters

---

component	(Optional) ID of the component to display
-----------	---

---



---

## historicaluserreportresults

**Purpose** This component displays the report results of a historical workflow process report for a performer.

### Configuration File

webcomponent\config\library\workflow\historicalreport\userresults\_component.xml

### Scope

All.

### Parameters

---

query	(Required) When the querytype is string, this value is the search string. If the querytype is not string, this value is the querystring.
title	(Optional) The title to display at the top of the report.

---

## history

**Purpose** Extends the component [objectgrid, page 884](#) and displays an object history (audit trail) including events, users, event source, and time stamp.

### Configuration File

webcomponent\config\library\history\history\_component.xml

## Parent Definition

objectgrid:/webcomponent/config/navigation/objectgrid/objectgrid\_component.xml

## Scope

All.

## Caller

This component is called by the action [history](#) (type `dm_sysobject`), page 315 in the menubar component and is displayed in the component [properties](#), page 908.

## Parameters

---

objectId	(Required) ID of the selected object
----------	--------------------------------------

---

## Elements

For information on configurable columns, see [doclist](#), page 684.

---

<queryconditionstring>	Specify the query for events to be shown. The dql query format will be of the form "select ... from FROM dm_audittrail WHERE queryconditionstring. By default, the following histories will be displayed: dm_addpackage, dm_completedworkitem, dm_removepackage, and dm_signoff.
<header>	Set the visible attribute to true to show a header (using the component as standalone). Contains <showlabel> and <showclosebutton>.
<showlabel>	True to show a label for the object.
<showclosebutton>	True to show a Close button within the header
<columns>	Contains <column> elements that configure audit attributes for the object.

<column>	True to display the column. The attribute value must correspond to an attribute, for example, time_stamp.
<lookuptable>	Contains <lookup> elements that supply user-friendly textual descriptions for system and user defined events. NLS mappings are provided in the <nlsbundle> file.
<lookup>	Has two configurable attributes: key specifies the attribute name, and nlsid specifies the NLS key to match to a localized string for the attribute.

---

## homecabinet\_drilldown

**Purpose** Supports browsing of the user home cabinet contents. The user can select cabinets or folders and drill down into them to see their contents. The visibility of object types is configurable. The component is identical to the drilldown component except that the top-level navigation location is the user's home cabinet path instead of the repository root cabinet list.

### Configuration File

webcomponent\config\navigation\homecabinet\homecabinet\_drilldown\_component.xml

### Scope

All.

### Parameters

---

folderPath	(Optional) Specifies the ID of the folder in which to start browsing.
------------	---

---

## Elements

---

<showfilters>	Set to true to show the objectfilters drop-down control
<objectfilters>	Contains filters that define which objects should be shown in the objects selection list.
<objectfilter>	Specifies a filter for the items that are displayed. Contains <label>, <showfolders>, <type>
<objectfilter>.<label>	Displays a label such as Folders or All Can contain a string or <nlsid>.
<objectfilter>. <showfolders>   <type>	To show folders only, set <showfolders> to true and <type> to null (no value). To show objects only, set <showfolders> to false and <type> to dm_sysobject. To show all, set <showfolders> to true and <type> to dm_sysobject. <type> can take any value that is a valid Documentum type.
<objectfilter>. showallversions	Set to true to display all versions in the object list
<columns>. <loadinvisibleattribute>	Uncomment this element and set to true to get invisible attribute values for use in your component. The invisible attributes can then be passed by configuring a column in the <columns> element. See <i>Web Development Kit and Client Applications Development Guide</i> for details.
<columns><column>	Specifies columns to show or hide
<column><attribute>	<attribute> sets the attribute to be displayed in the column.
<attribute> <label>	The value of <label> sets a label for the column.
<column><attribute> <visible>	Set <visible> to true to show the column.

### Drag and Drop Elements

<dragdrop>	Supports the drag and drop feature.
------------	-------------------------------------

---

<sourceactions>	Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.
<targetactions>	Contains zero or more <targetaction> classes that support drag and drop actions on targets in the component.
<targetaction>	Contains a fully qualified class name of the class that implements IDragTargetAction, for example, com.documentum.web.formext.control.dragdrop.CopyToFolderTargetAction.
<dataproviders>	Contains one or more <dataprovider> elements.
<dataprovider>	Contains a <format> element and a <provider> element.
<dataprovider>.<format>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<dataprovider>.<provider>	Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

## homecabinet\_list

### Purpose

Supports browsing of the user home cabinet. Users can select cabinets and folders and view their contents. The visibility of object types is configurable. The component is identical to the doclist component with the exception that the top-level navigation location is the user's home cabinet path instead of the repository root cabinet list.

## Configuration File

webcomponent\config\navigation\homecabinet\homecabinet\_list\_component.xml

## Scope

All.

## Parameters

---

folderPath	(Optional) Specifies the ID of the folder in which to start browsing.
------------	---

---

## Elements

---

<showfilters>	Set to true to show the objectfilters drop-down control
<objectfilters>	Contains filters that define which objects should be shown in the objects selection list.
<objectfilter>	Specifies a filter for the items that are displayed. Contains <label>, <showfolders>, <type>
<objectfilter>.<label>	Displays a label such as Folders or All Can contain a string or <nlsid>.
<objectfilter>. <showfolders>   <type>	To show folders only, set <showfolders> to true and <type> to null (no value). To show objects only, set <showfolders> to false and <type> to dm_sysobject. To show all, set <showfolders> to true and <type> to dm_sysobject. <type> can take any value that is a valid Documentum type.
<objectfilter>. showallversions	Set to true to display all versions in the object list

---

<columns>. <loadinvisibleattribute>	Uncomment this element and set to true to get invisible attribute values for use in your component. The invisible attributes can then be passed by configuring a column in the <columns> element. See <i>Web Development Kit and Client Applications Development Guide</i> for details.
<columns><column> <column><attribute>	Specifies columns to show or hide <attribute> sets the attribute to be displayed in the column.
<attribute> <label>	The value of <label> sets a label for the column.
<column><attribute> <visible>	Set <visible> to true to show the column.

### Drag and Drop Elements

<dragdrop>	Supports the drag and drop feature.
<sourceactions>	Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.
<targetactions>	Contains zero or more <targetaction> classes that support drag and drop actions on targets in the component.
<targetaction>	Contains a fully qualified class name of the class that implements IDragTargetAction, for example, com.documentum.web.formext.control.dragdrop.CopyToFolderTargetAction.
<dataproviders>	Contains one or more <dataprovider> elements.
<dataprovider>	Contains a <format> element and a <provider> element.
<dataprovider>.<format>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<dataprovider>.<provider>	Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

## httpcancelcheckout (type dm\_sysobject)

**Purpose** This is the WDK 5.3 httpcancelcheckout component. It enables cancelling checkouts of objects.

### Configuration File

webcomponent\config\library\contenttransfer\cancelcheckout\httpcancelcheckout\_component.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

This component is called by the action [cancelcheckout \(type dm\\_sysobject\), page 240](#) and is displayed in the component [httpcancelcheckoutcontainer, page 769](#).

### Parameters

---

objectId	(Required) (Required) r_object_id of the object for which to cancel the checkout.
----------	---

---



## Elements

<service>.<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.CancelCheckoutProcessor
<init-controls>	Initializes the nodescendents radio button

# httpcancelcheckoutcontainer

**Purpose** This is the WDK 5.3 httpcancelcheckoutcontainer component. It is a container for components that enable cancelling the checkout of a repository object.

## Configuration File

webcomponent\config\library\contenttransfer\cancelcheckout\httpcancelcheckoutcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [cancelcheckout](#) (type dm\_sysobject), page 240 and is displays the component [httpcancelcheckout](#) (type dm\_sysobject), page 768.

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

---

<code>component</code>	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<code>&lt;bindingcomponentversion&gt;</code>	Specifies the version of components that are supported by the container component. Default: latest
<code>&lt;setrepositoryfromobjectid&gt;</code>	Set to true to execute the query against the source repository. Affects reference and foreign objects.
<code>&lt;service&gt;.&lt;service-class&gt;</code>	Specifies the fully qualified class name for the class that implements the cancel checkout service. Value is: <code>com.documentum.web.contentxfer.impl.CancelCheckoutService</code>
<code>&lt;service&gt;.&lt;transport-class&gt;</code>	Specifies the fully qualified class name for the class that implements the transport protocol (in this case, HTTP). Value is: <code>com.documentum.web.contentxfer.http.HttpContentTransport</code>

---

<control-propagation>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<control-propagation>.<property name="enabled"/>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<control-propagation>.<property name="visible"/>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

---

## httpcancelcheckoutcontainer (version 5.2.5)

### Purpose

This is the WDK 5.2.5 httpcancelcheckoutcontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to display components that enable cancelling the checkout of a repository object.

### Configuration File

webcomponent\config\library\httpcancelcheckout\httpcancelcheckoutcontainer\_component.xml

### Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

---

version	5.2.5
---------	-------

---

## Caller

Cannot be called directly. Must be extended.

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container
<service>.<service-class>	Specifies the fully qualified class name for the class that implements the cancel checkout service. Value is: com.documentum.web.contentxfer.impl.CancelCheckoutService)
<service>.<transport-class>	Specifies the fully qualified class name for the class that implements the transport protocol (in this case, HTTP). Value is: com.documentum.web.contentxfer.http.HttpContentTransport

---

## httpcheckin (type dm\_sysobject)

**Purpose** This is the WDK 5.3 httpcheckin component. It checks in non-XML objects one at a time using the HTTP input mechanism.

### Configuration File

webcomponent\config\library\contenttransfer\checkin\httpcheckin\_component.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

This component is called by the action [checkin \(type dm\\_sysobject\), page 244](#) for applications that use the HTTP content transfer mechanism and displayed in the component [httpcheckincontainer, page 774](#).

### Parameters

objectId	(Required) (Required) The ID of the object to check in.
----------	---

### Elements

<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.CheckinProcessor
<init-controls>	Initializes values for the minorversion, newversion, and branchrevision radio controls; the makecurrent and

	checkindescendants checkboxes; and the showhideoptions link
<property-name>	Specifies an attribute on the control. Value is value.
<property-value>	Specifies an attribute on the control. Value is true.
<control>	Names a control and sets its initial value. The value of the name attribute must match the name of a control in the component JSP page. The value of the type attribute must be a fully qualified class name for a display control. Name attribute value is branchrevision; type attribute value is com.documentum.web.form.control.Radio.

---

## httpcheckincontainer

**Purpose** This is the WDK 5.3 httpcheckincontainer. It checks in non-XML objects one at a time using the HTTP input mechanism. (XML objects must be checked in using the UCF checkin component, which contains a parser.) The httpcheckincontainer component extends [combocontainer](#), [page 644](#).

### Configuration File

webcomponent\config\library\contenttransfer\checkin\httpcheckincontainer\_component.xml

### Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

### Scope

All.

## Caller

This container is launched by the action [checkin](#) (type `dm_sysobject`), [page 244](#) and displays the component [httpcheckin](#) (type `dm_sysobject`), [page 773](#) for applications that use the HTTP content transfer mechanism.

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

<code>component</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) List of arguments and values to pass to each contained component.

## Elements

<code>&lt;bindingcomponentversion&gt;</code>	Specifies the version of components that are supported by the container component. Default: latest
<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom <code>checkin</code> component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.

<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;setrepositoryfromobjectid&gt;</code>	Set to true to execute the query against the source repository. Affects reference and foreign objects.
<code>&lt;service-class&gt;</code>	Specifies the fully qualified class name for the class that implements the service. Value is: <code>com.documentum.web.contentxfer.impl.CheckinService</code>
<code>&lt;transport-class&gt;</code>	Specifies the fully qualified class name for the class that implements the transport; in this case, UCF. Value is: <code>com.documentum.web.contentxfer.http.HttpContentTransport</code>

---

## Usage

The parameters are added by the action execution class. Do not call the container by URL.

## httpcheckincontainer (version 5.2.5)

### Purpose

This is the WDK 5.2.5 httpcheckincontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to display the httpcheckin component to check in non-XML objects one at a time using the HTTP input mechanism. (XML objects must be checked in using the checkin applet, which contains a parser.) The httpcheckincontainer component extends [combocontainer](#), page 644.

### Configuration File

`webcomponent\config\library\httpcheckin\httpcheckincontainer_component.xml`



**Parent Definition**

combocontainer:wdk/config/combocontainer\_component.xml

**Scope**

version	5.2.5
---------	-------

**Caller**

Cannot be called directly. Must be extended.

**Parameters**

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

**Elements**

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
---------------------------	--

**httpcheckout (type dm\_sysobject)****Purpose**

This is the WDK 5.3 httpcheckout component. It checks out objects from the repository without launching an editing application. The httpcheckout component is available in all views for objects that are checked out on the user's current

machine. This component must be used within the [httpcheckoutcontainer, page 780](#) container. Checking out multiple objects creates one checkout container and an inner checkout component instance for each object.

## Configuration File

webcomponent\config\library\contenttransfer\checkout\httpcheckout\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [checkout \(type dm\\_sysobject\), page 247](#) and is displayed in the component [httpcheckoutcontainer, page 780](#).

## Parameters

---

isIgnoreDescendants	(Optional) Boolean value that indicates whether or not to check out descendants in a virtual document.
isLockOnly	(Optional) Boolean value that indicates whether or not to check out objects that are locked.
nodeId	(Optional) r_object_id of a virtual document node to check out.
objectId	(Required) The ID of the object to check out.
vdmRootObjectId	(Optional) r_object_id of the virtual document root.

---

## Elements

<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.CheckoutProcessor
<init-controls>	Initializes the nodescendents, alldescendents, and xmlrootnodownload radio button values

## httpcheckout (version 5.2.5)

**Purpose** This is the WDK 5.2.5 httpcheckout component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility.

### Configuration File

webcomponent\config\library\httpcheckout\httpcheckout\_action\_component.xml

### Scope

version	5.2.5
---------	-------

### Caller

Cannot be called directly. Must be extended.

### Parameters

objectId	The ID of the object to check out.
----------	------------------------------------

# httpcheckoutcontainer

**Purpose** This is the WDK 5.3 httpcheckoutcontainer component. It is the container for components that enable checking out objects from a repository.

## Configuration File

webcomponent\config\library\contenttransfer\checkout\httpcheckoutcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [checkout](#) (type `dm_sysobject`), page 247 and is displays the component [httpcheckout](#) (type `dm_sysobject`), page 777.

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
<control-propagation>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<control-propagation>.<property name="enabled"/>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<control-propagation>.<property name="visible"/>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<service-class>	Specifies the fully qualified class name for the class that implements the service. Value is: com.documentum.web.contentxfer.impl.CheckoutService
<transport-class>	Specifies the fully qualified class name for the class that implements the transport; in this case, HTTP. Value is: com.documentum.web.contentxfer.http.HttpContentTransport

---

## httpedit (type dm\_sysobject)

### Purpose

This is the WDK 5.3 httpedit component. It enables checking out objects from repositories and opening them for editing in an associated application on the local machine.

## Configuration File

webcomponent\config\library\contenttransfer\edit\httpedit\_component.xml

## Parent Definition

httpcheckout:webcomponent/config/library/contenttransfer/checkout/httpcheckout\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [editfile](#) (type [dm\\_sysobject](#)), [page 292](#) and is displayed in the component [httpeditcontainer](#), [page 783](#).

## Parameters

These parameters are inherited from [httpcheckout:webcomponent/config/library/contenttransfer/checkout/httpcheckout\\_component.xml](#).

---

isIgnoreDescendents	(Optional) Boolean value that indicates whether or not to check out descendants in a virtual document.
isLockOnly	(Optional) Boolean value that indicates whether or not to check out objects that are locked.
nodeId	(Optional) <code>r_object_id</code> of a virtual document node to check out.
objectId	(Required) The ID of the object to check out.
vdmRootObjectId	(Optional) <code>r_object_id</code> of the virtual document root.

---

## Elements

---

<processor-class>	Specifies the fully qualified class name for the class that implements a processor class; in this case a class that starts an associated application on the local machine. Value is: com.documentum.web.contentxfer.impl.EditProcessor
-------------------	--

---

# httpeditcontainer

**Purpose** This is the WDK 5.3 httpeditcontainer component. It is used to display components that enable checking out objects and opening them for editing in an associated application via the HTTP protocol.

## Configuration File

webcomponent\config\library\contenttransfer\edit\httpeditcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [editfile](#) (type `dm_sysobject`), [page 292](#) and displays the component [httpedit](#) (type `dm_sysobject`), [page 781](#).

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
<control-propagation>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<control-propagation>.<property name="enabled"/>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<control-propagation>.<property name="visible"/>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<service-class>	Specifies the fully qualified class name for the class that implements the service. Value is: com.documentum.web.contentxfer.impl.EditService
<transport-class>	Specifies the fully qualified class name for the class that implements the transport; in this case, HTTP. Value is: com.documentum.web.contentxfer.http.HttpContentTransport

---



## httpexport (type dm\_sysobject)

**Purpose** This is the WDK 5.3 httpexport component. It is used to export dm\_sysobjects from the repository to the client file system. This component must run within the 5.3 exportcontainer (see [export \(type dm\\_sysobject\)](#), page 716).

### Configuration File

webcomponent\config\library\contenttransfer\export\httpexport\_component.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

This component is called by the action [export \(type dm\\_sysobject\)](#), page 298 and is displayed in the component [httpexportcontainer](#), page 788.

### Parameters

contentType	(Optional) File format to use when opening the object (passed to the operating system)
objectId	(Required) ID of the object to be exported
pageModifier	(Optional) A valid value for the page_modifier attribute of the dmr_content object. You should not pass in your own value for PDF renditions.

## Elements

---

<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.ExportProcessor
<init-controls>	Initializes values for the downloadDesc-Checkbox

---

## httpexport (type dmr\_content)

**Purpose** This is the WDK 5.3 httpexport component. It is used to export dmr\_content objects from the repository to the client file system. This component must run within the 5.3 exportcontainer (see [export \(type dmr\\_content\)](#), page 719). Export of multiple objects creates one export container and an inner export component instance for each object.

### Configuration File

webcomponent\config\library\contenttransfer\export\httpexport\_component.xml

### Scope

---

type	dmr_content
------	-------------

---

### Caller

This component is called by the action [exportrendition \(type dmr\\_content\)](#), page 299, and is displayed in the component [httpexportcontainer](#), page 788.

## Parameters

contentType	(Optional) File format to use when opening the object (passed to the operating system)
objectId	(Required) ID of the object to be exported
pageModifier	(Optional) A valid value for the page_modifier attribute of the dmr_content object. You should not pass in your own value for PDF renditions.

## Elements

<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.ExportProcessor
<init-controls>	Top-level element for the init-controls feature.
<control>	Initializes a value for the downloadDesc-Checkbox

# httpexport (version 5.2.5)

**Purpose** This is the WDK 5.2.5 httpexport component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility and must run within the 5.2.5 exportcontainer.

## Configuration File

webcomponent\config\library\httpexport\httpexport\_action\_component.xml

## Scope

---

version	5.2.5
---------	-------

---

## Caller

Cannot be called directly. Must be extended.

## Parameters

---

contentType	(Optional) File format to use when opening the object (passed to the operating system)
objectId	(Required) ID of the object to be exported

---

# httpexportcontainer

**Purpose** This is the WDK 5.3 httpexportcontainer component. It is the container for components that enable exporting objects from repositories to local machines.

## Configuration File

webcomponent\config\library\contenttransfer\export\httpexportcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [export](#) (type `dm_sysobject`), [page 298](#) and [exportrendition](#) (type `dmr_content`), [page 299](#), and displays the components [httpexport](#) (type `dm_sysobject`), [page 785](#) and [httpexport](#) (type `dmr_content`), [page 786](#) .

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

<code>component</code>	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

## Elements

<code>&lt;bindingcomponentversion&gt;</code>	Specifies the version of components that are supported by the container component. Default: latest
<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.

<code>&lt;control-propagation&gt;&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;service-class&gt;</code>	Specifies the fully qualified class name for the class that implements the service. Value is: <code>com.documentum.web.contentxfer.impl.ExportService</code>
<code>&lt;transport-class&gt;</code>	Specifies the fully qualified class name for the class that implements the transport; in this case, UCF. Value is: <code>com.documentum.web.contentxfer.ucf.UcfContentTransport</code>

---

## httpimport

**Purpose** This is the WDK 5.3 httpimport component. It must be used within the httpimportcontainer component. Using the HTTP protocol, it enables the import of local files into repositories.

### Configuration File

```
webcomponent\config\library\contenttransfer\importcontent\httpimport_
component.xml
```

### Scope

All.

### Parameters

---

<code>baseDocbaseType</code>	(Optional) Parent <code>object_type</code> of the <code>object_type</code> of the object to be imported.
<code>docbaseType</code>	(Optional) <code>object_type</code> that the file is to be imported as.

---

filenameWithPath	(Required) True to display the file path with the file name
localFilePath	(Optional) Full path to the file to be imported.
objectId	(Required) ID of the target folder for import

---

## Elements

---

<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.ImportProcessor
<init-controls>	Initializes values of the xmlCategoryListLabel label control and the xmlCategoryList datadropdownlist control
<document-dochbase-type>	Default object type of the object to be imported.
<preserve-file-extension>	Specify True to keep the file extension as part of the object name when imported into the repository; False to strip the file extension from the object name.

---

# httpimportcontainer

**Purpose** This is the WDK 5.3 httpimportcontainer component. It is the container for components that use the HTTP protocol to import local files into repositories.

## Configuration File

```
webcomponent\config\library\contenttransfer\importcontent\httpimportcontainer_
component.xml
```

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.
filePath	(Optional) Path to the file on the local machine.
isDirectory	(Optional) Boolean that indicates whether the object being imported is a directory.
parentPath	(Optional) Repository path into which the file is imported.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
<control-propagation>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls



---

<control-propagation>.<property name="enabled"/>	are propagated to the JSP shown for each object being checked in.
<control-propagation>.<property name="visible"/>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<service-class>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<transport-class>	Specifies the fully qualified class name for the class that implements the service. Value is: com.documentum.web.contentxfer.impl.ImportService Specifies the fully qualified class name for the class that implements the transport; in this case, HTTP. Value is: com.documentum.web.contentxfer.http.HttpContentTransport

---

## httpimportcontainer (version 5.2.5)

**Purpose** This is the WDK 5.2.5 httpimportcontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility and must be used within the 5.2.5 httpimporttrenditioncontainer.

### Configuration File

webcomponent\config\library\httpimportcontent\httpimportcontainer\_component.xml

### Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

**Scope**

---

version	5.2.5
---------	-------

---

**Caller**

Cannot be called directly. Must be extended.

**Parameters**

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## httpimportrendition

**Purpose**

This is the WDK 5.3 httpimportrendition component. It extends the component [import, page 804](#) and is used to import a single rendition into the repository using the HTTP content transfer. The component must run within the 5.3 container httpimportrenditioncontainer.

**Configuration File**

webcomponent\config\library\contenttransfer\importrendition\httpimportrendition\_component.xml

**Scope**

All.

## Caller

This component is called by the actions [importrendition \(type dm\\_sysobject\)](#), [page 318](#) and [importrendition \(type dmr\\_content\)](#), [page 319](#), and is displayed in the component [httpimportrenditioncontainer](#), [page 795](#).

## Parameters

objectId	(Required) ID of the target folder for import
----------	---

## Elements

<service>.<processor-class>	Specifies the fully qualified class name for the class that processes the rendition
<updatereplicasource>	Set to true to add a rendition to both the source and replica document. Set to false to add the rendition to the selected document only (either a source or replica).
<init-controls>.<control>	Initializes a value for the formatList datadropdownlist control

# httpimportrenditioncontainer

**Purpose** This is the WDK 5.3 httpimportrenditioncontainer component. It is the container for components that are used to import renditions into repositories.

## Configuration File

```
webcomponent\config\library\contenttransfer\importrendition\httpimportrenditioncontainer_
component.xml
```

## Parent Definition

```
combocontainer:wdk/config/combocontainer_component.xml
```

## Scope

All.

## Caller

This component is called by the actions [importrendition \(type dm\\_sysobject\), page 318](#) and [importrendition \(type dmr\\_content\), page 319](#) , and displays the component [httpimportrendition, page 794](#).

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

---

<code>component</code>	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<code>&lt;bindingcomponentversion&gt;</code>	Specifies the version of components that are supported by the container component. Default: latest
<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.

---

<code>&lt;control-propagation&gt;&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

---

## httpimportrenditioncontainer (version 5.2.5)

**Purpose** This is the WDK 5.2.5 httpimportrenditioncontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to display components that enable importing renditions into repositories.

### Configuration File

webcomponent\config\library\httpimportrendition\httpimportrenditioncontainer\_component.xml

### Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

### Scope

---

version	5.2.5
---------	-------

---

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: 5.2.5
---------------------------	---

---

# httpmultifiledownload

## Purpose

Displays a list of files to be downloaded over HTTP content transfer. The user will not have to open multiple windows at the same time to do simultaneous downloads. It is used by all outbound HTTP content transfer operations when more than one file is selected.

## Configuration File

wdk\config\httpmultifiledownload\_component.xml

## Scope

All.

## Parameters

componentArgs	Not used
contentId	Array of IDs of selected objects for download

## httpview

**Purpose** This is the WDK 5.3 httpview component. It downloads repository content to the browser local file system and opens a viewing application on the client. The httpview component must run in the [httpviewcontainer, page 801](#) container.

### Configuration File

webcomponent\config\library\contenttransfer\view\httpview\_component.xml

### Scope

All.

### Caller

This component is called by the action [view \(type dm\\_sysobject\), page 465](#) and [viewcontent \(type dm\\_sysobject\), page 472](#), and is displayed in the component [httpviewcontainer, page 801](#).

## Parameters

contentType	(Optional) File format to use when opening the object (passed to the operating system)
objectId	(Required) ID of the object to be viewed

## httpview (type dmr\_content)

**Purpose** Downloads repository content to the browser local file system and opens a viewing application on the client. The view component must run in the [view \(type dmr\\_content\), page 1106](#) container.

### Configuration File

webcomponent\config\library\contenttransfer\view\httpview\_component.xml

### Scope

---

type	dmr_content
------	-------------

---

### Caller

This component is called by the action [view \(type dmr\\_content\), page 468](#) and is displayed in the component [httpviewcontainer, page 801](#).

### Parameters

---

contentType	(Optional) File format to use when opening the object (passed to the operating system)
objectId	(Required) ID of the object to be viewed

---

## httpview (version 5.2.5)

**Purpose** This is the WDK 5.2.5 httpview component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to download repository content to the browser local file system and opens a viewing application



on the client. The view component must run in the [view](#) (type `dm_sysobject`), [page 1104](#) container.

## Configuration File

`webcomponent\config\library\httpview\httpview_component.xml`

## Scope

version	5.2.5
---------	-------

## Caller

Cannot be called directly. Must be extended.

## Parameters

contentType	(Optional) File format to use when opening the object (passed to the operating system)
objectId	(Required) ID of the object to be viewed

# httpviewcontainer

**Purpose** 5.3 container for components that enable (via the HTTP protocol) viewing objects in an associated application on local machines

## Configuration File

`webcomponent\config\library\contenttransfer\view\httpviewcontainer_component.xml`

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [view \(type dm\\_sysobject\), page 465](#), [view \(type dmr\\_content\), page 468](#), and [viewcontent \(type dm\\_sysobject\), page 472](#), and displays the components [httpview, page 799](#) and [httpview \(type dmr\\_content\), page 800](#).

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
<control-propagation>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the

---

<control-propagation>.<property name="enabled"/>	enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<control-propagation>.<property name="visible"/>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.  (Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

---

## iconwelltest

**Purpose** Tests the iconwell control in the dmf tag library.

### Configuration File

wdk\config\iconwelltest\_component.xml

### Scope

All.

### Parameters

---

component	(Required) At least one component must be specified as the value of <iconwell>.<id> in the component definition.
-----------	--

---

# import

**Purpose** This is the WDK 5.3 import component. Imports objects from the client file system to the repository. The component must run within the 5.3 importcontainer or httpimportcontainer.

## Configuration File

webcomponent\config\library\contenttransfer\importcontent\import\_component.xml

## Scope

All.

## Caller

This component is called by the action [importexternalresult](#) (type `dm_externalresult`), [page 317](#).

## Parameters

---

baseDocbaseType	(Optional) Parent object_type of the object_type of the object to be imported.
defaultAttributesValues	(Optional) Default attribute values for the object that is to be imported.
docbaseType	(Optional) object_type that the file is to be imported as.
filenameWithPath	(Required) True to display the file path with the file name
format	(Optional) dm_format of the object to be imported.
isDirectory	(Optional) Boolean that indicates whether the object being imported is a directory.
localFilePath	(Optional) Full path to the file to be imported.

---

objectId	(Required) ID of the target folder for import
parentPath	(Optional) Repository path into which the file is imported.

---

## Elements

---

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<processor-class>	Specifies the fully qualified class name for the class that implements the processor. Value is: com.documentum.web.contentxfer.impl.ImportProcessor
<init-controls>	Initializes values of the xmlCategoryListLabel label control and the xmlCategoryList datadropdownlist control
<document-docbase-base-type>	Default base for retrieving a type list to be displayed on import. Default: dm_document
<document-docbase-type>	Default selected object type of an object to be imported. Must be the type or subtype of <document-docbase-base-type>. Default: dm_document
<folder-docbase-type>	Default folder type of a folder to be imported
<preserve-file-extension>	Specify True to keep the file extension as part of the object name when imported into the repository; False to strip the file extension from the object name.

---

## Usage

When the user imports a document and selects a custom type for the document, a format dropdown list is presented for selecting the associated document format. The custom type must have a template in order to be associated with a format. You must have a template in the Templates cabinet for each format that will apply to your custom type.

To customize the import layout, use the [doibaseattributelist, page 76](#) control. First, create a definition file that is scoped to each type for which you wish to provide a different set of import attributes, as described in [doibaseattributelist, page 76](#). In the JSP page for the custom type, add a tag for the custom attribute list that you wish to display on import. Add a tag similar to the following:

```
<dmfx:doibaseattributelist  
  name="attrlist" object="obj" attrconfigid="import"/>
```

All custom attributes that are in a category appear unless they are listed in the `<ignore_attributes>` element. Attributes are displayed in order of category. If the value of `showpagesastabs`

You can have different attribute lists for checkin, properties, import, and other components.

## import (version 5.2.5)

**Purpose** This is the WDK 5.2.5 import component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility and must be used within the 5.2.5 importcontainer.

### Configuration File

webcomponent\config\library\importContent\import\_component.xml

### Scope

---

version	5.2.5
---------	-------

---

### Caller

Cannot be called directly. Must be extended.

## Parameters

filenameWithPath	(Required) True to display the file path with the file name
objectId	(Required) ID of the target folder for import

## Elements

<default_formats>	Contains <format> elements that should be matched to DOS file extensions. For example, there can be 7 dm_format objects that have the extension .doc.
<format>	Each format describes a default format that is used when a user imports a file with a filename extension that matches the value of the dos_extension attribute. If a configured format cannot be found, the import component will try to find a format with the file's dos extension. If no format can be found, the user can select a format. Attributes: dos_extension (for example, 'doc') or name (for example, 'msw8').

# importcontainer

**Purpose** This is the WDK 5.3 importcontainer component. It is the container used to display components that enable the import of local files into repositories.

## Configuration File

webcomponent\config\library\contenttransfer\importcontent\importcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Parameters

---

component	(Required) Supplied by the action execution class). The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.
filePath	(Optional) Path to the file on the local machine.
isDirectory	(Optional) Boolean that indicates whether the object being imported is a directory.
parentPath	(Optional) Repository path into which the file is imported.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
<control-propagation>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls



---

<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	are propagated to the JSP shown for each object being checked in.  (Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;ucfrequired&gt;</code>	If this element is present, then the component requires UCF to be installed.
<code>&lt;max-import-file-count&gt;</code>	Specifies the maximum number of files that can be imported in a single operation.

---

## importcontainer (version 5.2.5)

**Purpose** This is the WDK 5.2.5 importcontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to display components that enable importing local files into repositories.

### Configuration File

webcomponent\config\library\importContent\importcontainer\_component.xml

### Parent Definition

contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml

### Scope

---

version	5.2.5
---------	-------

---

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
---------------------------	--

---

# importrendition

**Purpose** This is the WDK 5.3 importrendition component. It extends the component [import, page 804](#) and is used to import a rendition into the repository. The component must run within the 5.3 container importrenditioncontainer.

## Configuration File

```
webcomponent\config\library\contenttransfer\importrendition\importrendition_
component.xml
```

## Scope

All.

## Caller

This component is called by the actions [importrendition \(type dm\\_sysobject\)](#), page 318 and [importrendition \(type dmr\\_content\)](#), page 319, and is displayed in the component [importrenditioncontainer](#), page 812.

## Parameters

objectId	(Required) ID of the target folder for import
----------	---

## Elements

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
<service>.<processor-class>	Specifies the fully qualified class name for the class that processes the rendition
<updatereplicasource>	Set to true to add a rendition to both the source and replica document. Set to false to add the rendition to the selected document only (either a source or replica).
<init-controls>.<control>	Initializes a value for the formatList data dropdownlist control

# importrendition (version 5.2.5)

## Purpose

This is the WDK 5.2.5 importrendition component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility and must be used within the 5.2.5 importrendition container.

## Configuration File

webcomponent\config\library\importrendition\importrendition\_component.xml

### Parent Definition

import:webcomponent/config/library/importContent/import\_component.xml

### Scope

---

version	5.2.5
---------	-------

---

### Parameters

---

objectId	(Required) ID of the target folder for import
----------	---

---

## importrenditioncontainer

**Purpose** This is the WDK 5.3 importrenditioncontainer component. It is the container for components that enable the import of renditions into repositories.

### Configuration File

webcomponent\config\library\contenttransfer\importrendition\importrenditioncontainer\_component.xml

### Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

### Scope

All.

## Caller

This component is called by the actions [importrendition \(type dm\\_sysobject\)](#), [page 318](#) and [importrendition \(type dmr\\_content\)](#), [page 319](#) , and displays the component [importrendition](#), [page 810](#).

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

## Elements

<code>&lt;bindingcomponentversion&gt;</code>	Specifies the version of components that are supported by the container component. Default: latest
<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.

<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;ucfrequired&gt;</code>	If this element is present, then the component requires UCF to be installed.

---

## importrenditioncontainer (version 5.2.5)

**Purpose** This is the WDK 5.2.5 importrenditioncontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to display components that enable importing renditions into repositories.

### Configuration File

webcomponent\config\library\importrendition\importrenditioncontainer\_component.xml

### Parent Definition

contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml

### Scope

---

version	5.2.5
---------	-------

---

### Parameters

These parameters are inherited from contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
---------------------------	--

---

# inboxdrilldown

**Purpose** Lists the user's inbox tasks, using a streamline-style layout

## Configuration File

webcomponent\config\navigation\inbox\inboxdrilldown\_component.xml

## Scope

All.

## Elements

---

<failoverenabled>	Specifies that this component is enabled for failover. Set to false to disable.
<showattachmentinfo>	Boolean: true to show the attachment info

<taskmanagerid>	ID of the task manager component accessible from inbox
<showfilters>	True to show object type filters
<workflowstatusaction name>	Specifies the action that calls the workflow status page
<workflowreportaction name>	Specifies the action that calls the workflow report page
<showautogettask>	Set to true to display a checkbox to get next task automatically
<autogettaskdefault>	Default value for auto get next task checkbox (default = unchecked)

---

A default set of columns is configured for initial display. Users can also set column preferences in the `display_preferences` component UI. For information on configuring the initial columns, see [doclist, page 684](#).

## inboxlist

**Purpose** Lists the user's inbox tasks, using a classic-style layout.

### Configuration File

`webcomponent\config\navigation\inbox\inboxlist_component.xml`

### Scope

All.

### Elements

---

<failoverenabled>	Specifies that this component is enabled for failover. Set to false to disable.
<showattachmentinfo>	Boolean: true to show the attachment info



---

<taskmanagerid>	ID of the task manager component accessible from inbox
<showfilters>	True to show object type filters
<workflowstatusactionname>	Specifies the action that calls the workflow status page
<workflowreportactionname>	Specifies the action that calls the workflow report page
<showautogettask>	Set to true to display a checkbox to get next task automatically
<autogettaskdefault>	Default value for auto get next task checkbox (default = unchecked)

---

A default set of columns is configured for initial display. Users can also set column preferences in the display\_preferences component UI. For information on configuring the initial columns, see [doclist, page 684](#).

## index

**Purpose** Not used.

### Configuration File

webcomponent\config\navigation\index\index.xml

### Scope

All.

## informinvalidactionforreference

**Purpose** Launched by certain actions that use a dynamic filter to display a message to the user that the action is invalid for the selected reference object.

## Configuration File

webcomponent\config\library\informinvalidaction\informinvalidactionforreference\_  
component.xml

## Scope

All.

## Caller

This component is invoked by many actions that use a dynamic filter to display an error message when an action is invalid.

## Usage

For actions that should not be performed on reference objects, add a dynamic filter to the action definition and list the action as notdefined for the scope foreign. To create a dynamic filter in the action definition, see "Dynamic Component Launching" in *Web Development Kit and Client Applications Development Guide*. To specify the action as notdefined, create an XML file in the /custom/config directory or a subdirectory with the following type of entries:

```
<scope type="foreign">
  <!-- disabled actions -->
  <action id="versions" notdefined="true"></action>
  <action id="relationships" notdefined="true"></action>
  ...
</scope>
```

# informinvalidactionforreplica

## Purpose

Launched by certain actions that use a dynamic filter to display a message to the user that the action is invalid for the selected replicated object.

## Configuration File

webcomponent\config\library\informinvalidaction\informinvalidactionforreplica\_  
component.xml

## Scope

All.

## Caller

This component is invoked by many actions that use a dynamic filter to display an error message when an action is invalid for the selected replicated object.

## Usage

For actions that should not be performed on replica objects, add a dynamic filter to the action definition and list the action as notdefined for the scope foreign. To create a dynamic filter in the action definition, see *Dynamic Component Launching in Web Development Kit and Client Applications Development Guide*. To specify the action as notdefined, create an XML file in the /custom/config directory or a subdirectory with the following type of entries:

```
<scope type="foreign">  
  <!-- disabled actions -->  
  <action id="versions" notdefined="true"></action>  
  <action id="relationships" notdefined="true"> </action>  
  ...
```

# installedrecentwftemplatelocator

**Purpose** Extends [recentwftemplatelocator](#), page 931 and allows the user to locate a recently used workflow template that has been installed.

## Configuration File

webcomponent\config\library\workflow\startworkflow\startworkflowtemplatelocatorcontainer\_  
component.xml

## Parent Definition

recentwftemplatelocator:webcomponent/config/library/locator/wftemplatelocators\_ component.xml

## Scope

All.

## Parameters

These parameters are inherited from recentssystemobjectlocator:webcomponent/config/ library/locator/recentssystemobjectlocator\_ component.xml.

---

contentType	(Optional) The type of content
multiselect	(Optional) Boolean: set to true to support multiple selection.
objecttype	(Optional) Object type of the items to display in the list.

---

## Elements

The configurable elements are inherited from [sysobjectlocator](#), page 1029.

# installedwftemplatefolderlocator

**Purpose** Extends [unfilteredcontainersysobjectlocator](#), page 1064 and allows the user to locate a template folder.

## Configuration File

webcomponent\config\library\workflow\startworkflow\startworkflowtemplatelocatorcontainer\_ component.xml

## Parent Definition

unfilteredcontainersysobjectlocator:webcomponent/config/library/locator/  
unfilteredcontainersysobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/  
locator/sysobjectlocator\_component.xml.

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) The folder in which to search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

For information on configurable columns, see [doclist](#), page 684. Inherited from [sysobjectlocator](#), page 1029.

# installedwftemplatesubscriptionlocator

## Purpose

Extends [wftemplatesubscriptionlocator](#), page 1133 and locates and displays installed workflow templates to which the user has subscribed.

## Configuration File

webcomponent\config\library\workflow\startworkflow\startworkflowtemplatelocatorcontainer\_  
component.xml

## Parent Definition

wftemplatesubscriptionlocator:webcomponent/config/library/locator/  
wftemplatelocators\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from subscriptionlocator:webcomponent/config/library/  
locator/subscriptionlocator\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements

<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.

<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## jobabort

**Purpose** Called by the action [abortjob](#), page 204 to abort a running asynchronous action or component job.

### Configuration File

webcomponent\config\library\async\jobabort\_component.xml

### Scope

All.



## Caller

This component is called by the action [abortjob](#), [page 204](#) and displayed in the component [dialogcontainer](#), [page 669](#).

## Parameters

---

jobId	(Required) ID of the job to abort.
-------	------------------------------------

---

# jobprogressmonitor

**Purpose**        Displays the progress of an asynchronous job

## Configuration File

webcomponent\config\library\progress\progress\_component.xml

## Scope

All.

## Caller

This component is nested to by every container that extends the 5.3 ContentTransferServiceContainer class.

## Parameters

---

jobId	(Optional) ID of the asynchronous job
pending	(Optional) Boolean flag that signals to the JobProgressMonitor class that the job is already pending. This flag is set by the content transfer container class

when nesting to the jobprogressmonitor component.

---

## jobstatus

**Purpose** This component displays status and details about every running asynchronous action or component job.

### Configuration File

webcomponent\config\library\async\jobstatus\_component.xml

### Scope

All.

### Caller

This component is called by the action [showjobstatus, page 423](#) in the Webtop statusbar UI.

## jobstatushistory

**Purpose** This component displays job history and messages for a selected job.

### Configuration File

webcomponent\config\library\async\jobstatushistory\_component.xml

## Scope

All.

## Caller

This component is called by the action [showjobstatusthistory, page 423](#) in the jobstatus component UI. It is displayed in the component [dialogcontainer, page 669](#).

## Parameters

---

jobId	(Required) ID of the selected job.
jobName	(Required) Name of the selected job.

---

# launchrepositoryselector

**Purpose** This component is nested to from the locatorcontainer component to call the launchauthenticate action.

## Configuration File

wdk\config\launchrepositoryselector\_component.xml

## Scope

All.

## Parameters

---

suppressLoginEvent	(Optional) Specifies whether to fire the loginSuccess or loginCancel client events after the login dialog has been dismissed
--------------------	--

---

# lifecyclefolderlocator

**Purpose** Extends the component [sysobjectlocator, page 1029](#) and displays in a folder view all objects with attached lifecycles.

## Configuration File

webcomponent\config\library\locator\lifecyclelocators\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

For information on configurable columns, see [doclist](#), page 684. The configurable elements are inherited from [objectlocator](#), page 886.

# lifecyclelocatorcontainer

**Purpose** Extends the container [locatorcontainer](#), page 839 and contains lifecyclelocator components: [alllifecyclelocator](#), page 533, [lifecyclesubscriptionlocator](#), page 830, [lifecyclefolderlocator](#), page 828, and [recentlifecyclelocator](#), page 925.

## Configuration File

webcomponent\config\library\locator\lifecyclelocators\_component.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## lifecyclesubscriptionlocator

**Purpose** Extends [subscriptionlocator, page 1024](#) and displays subscribed lifecycle objects.

### Configuration File

webcomponent\config\library\locator\lifecyclelocators\_component.xml

### Parent Definition

subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml

### Scope

All.

### Parameters

These parameters are inherited from subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

For information on configurable columns, see [doclist, page 684](#). The configurable elements are inherited from [objectlocator, page 886](#).

# link

**Purpose** This component is used to create a link to a local file or to create a reference object.

## Configuration File

webcomponent\config\library\link\link\_component.xml

## Scope

All.

## Parameters

folderId	(Required) (Required) ID of the target folder for the link
objectId	(Required) (Required) Object for which to create a link

# linkcontainer

**Purpose** This component displays the link component, which is used to create a link to a local object or to create a reference object.

## Configuration File

webcomponent\config\library\link\linkcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Parameters

---

component	(Required) Supplied by the action execution class). The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.
localItems	(Optional) This is a comma-separated list of IDs of linked items from the local repository. Local items are linked differently than items linked from external sources. IDs of items not listed in the localItems parameter are assumed to be from an external repository.

---

# locatecheckinlocalfile

**Purpose** Enables browsing the local filesystem to find files to check in.

## Configuration File

webcomponent\config\library\contenttransfer\checkin\locatecheckinlocalfile\_component.xml



## Parent Definition

locatelocalfile:webcomponent/config/library/contenttransfer/locatelocalfile\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from locatelocalfile:webcomponent/config/library/contenttransfer/locatelocalfile\_component.xml.

fileName	(Optional) The local path and name of the file to check in.
----------	---

## Elements

<control-propagation>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<control-propagation>.<property name="enabled"/>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<control-propagation>.<property name="visible"/>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

## locatecheckinlocalfilecontainer

**Purpose** Container for components that enable browsing the local filesystem to find files to check in.

### Configuration File

webcomponent\config\library\contenttransfer\checkin\locatecheckinlocalfilecontainer\_component.xml

### Parent Definition

locatelocalfilecontainer:webcomponent/config/library/contenttransfer/  
locatelocalfilecontainer\_component.xml

### Scope

All.

### Parameters

These parameters are inherited from locatelocalfilecontainer:webcomponent/config/  
library/contenttransfer/locatelocalfilecontainer\_component.xml.

---

component	(Required) (Required) The name of the component to contain.
fileName	(Optional) The local path and name of the file to check in.

---

## locateimportlocalfile

**Purpose** Displays local files for import into repositories.

## Configuration File

```
webcomponent\config\library\contenttransfer\importcontent\locateimportlocalfile_
component.xml
```

## Parent Definition

```
locatelocalfile:webcomponent/config/library/contenttransfer/locatelocalfile_
component.xml
```

## Scope

All.

## Parameters

These parameters are inherited from locatelocalfile:webcomponent/config/library/contenttransfer/locatelocalfile\_component.xml.

---

fileName	(Optional) Name of the file to import.
----------	--

---

# locateimportlocalfilecontainer

**Purpose** Container for components that display local files to import.

## Configuration File

```
webcomponent\config\library\contenttransfer\importcontent\locateimportlocalfilecontainer_
component.xml
```

## Parent Definition

```
locatelocalfilecontainer:webcomponent/config/library/contenttransfer/
locatelocalfilecontainer_component.xml
```

**Scope**

All.

**Parameters**

These parameters are inherited from locatelocalfilecontainer:webcomponent/config/library/contenttransfer/locatelocalfilecontainer\_component.xml.

---

component	(Required) Supplied by the action execution class). The name of the component to contain.
fileName	(Optional) Name of the file to import.

---

## locatelocalfile

**Purpose**      Displays local files.**Configuration File**

webcomponent\config\library\contenttransfer\locatelocalfile\_component.xml

**Scope**

All.

**Parameters**

---

fileName	(Optional) Name of the file to import.
----------	--

---

# locatelocalfilecontainer

**Purpose** Container for components that display local files.

## Configuration File

webcomponent\config\library\contenttransfer\locatelocalfilecontainer\_component.xml

## Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml

## Scope

All.

## Parameters

component	(Required) Supplied by the action execution class). The name of the component to contain.
fileName	(Optional) Name of the file to import.

# locations

**Purpose** Extends [objectgrid, page 884](#) and lists the folders and virtual documents to which the selected object is directly linked.

## Configuration File

webcomponent\config\library\locations\locations\_component.xml

## Parent Definition

objectgrid:/webcomponent/config/navigation/objectgrid/objectgrid\_component.xml

## Scope

All.

## Caller

This component is called by the action [locations](#) (type `dm_sysobject`), [page 327](#) and displayed in the component [navigationcontainer](#), [page 864](#).

## Parameters

---

objectId	(Required) ID of the object
----------	-----------------------------

---

## Elements

---

<dragdrop>	Supports the drag and drop feature.
<sourceactions>	Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.
<dataproviders>	Contains one or more <dataprovider> elements.
<dataprovider>	Contains a <format> element and a <provider> element.
<dataprovider>.<format>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<dataprovider>.<provider>	Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

---

# locatorcontainer

**Purpose** Extends [propertysheetcontainer](#), page 914 and serves as a base locator container component.

## Configuration File

webcomponent\config\library\locator\locatorcontainer\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

All.

## Parameters

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) The ID number of the folder in which to begin the search.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) The docbase in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## Elements

---

<multidocbasesupport>	Set value to true to enable the locator to search across repositories
-----------------------	---

---

# login

**Purpose** Performs an explicit login to a single repository or a trusted login to a J2EE application server. The trusted login will log in to the repositories accessible to the application.

## Configuration File

wdk\config\login\_component.xml

## Scope

All.

## Caller

The login component is called by the component dispatcher if a component is called without a valid session.

## Parameters

---

docbase	(Optional) Specifies the repository to log into
startComponent	(Optional) If an entry component name is specified, the login component will forward to that component after successful login.



---

startPage	(Optional) This parameter sets the component page that will be displayed on forward.
startUrl	(Optional) URL to forward to. If a startURL is specified, the login component will forward to that URL after successful login. This value takes precedence over startComponent. The URL must be relative to the root context, not relative to the directory location of the current JSP page.

---

## Elements

---

<failoverenabled>	Set to true to enable failover. Requires that you also enable failover in app.xml.
<serveroptionenabled>	Set to true to display an additional login option of any running Content Server
<showoptions>	Set to true to display the additional options panel in the login UI.

---

## Usage

For more information on the types of application login, see the section on application login and authentication in *Web Development Kit and Client Applications Development Guide*.

In addition to the showoptions element in the login component definition, the Web application's definition file (app.xml) contains an <authentication> element whose child elements <domain> and <docbase> specify the default domain and repository, respectively. You can configure these values in the app.xml file in your custom directory (/custom/app.xml).

These values for default domain and repository are used to initialize the login dialog. The user can change the domain and repository in the login screen, and the user's selections will be stored in a cookie for the next login.

Several types of login connections are provided in the WDK authentication framework:

- Login dialog
- Skip authentication (for example, to the login or password component)
- J2EE principal user login

Each Documentum component requires a Documentum connection unless skip authentication is configured (for example, for the login component). If a Documentum connection is not available when the component is called, the component dispatcher calls the authentication service to obtain a session. If the authentication service does not obtain a session, the dispatcher calls the login component to create one. The login component then forwards to the original request after authentication.

## logoff

**Purpose** The logoff component is one of the components specified in the file Environment.properties (located in /WEB-INF/classes/com/documentum/web/formext) as a non\_docbase\_component, which means that it does not require a Documentum session.

### Configuration File

wdk\config\logoff\_component.xml

### Scope

All.

### Caller

The logoff component is called by the logout action in the menubar and titlebar components.

### Parameters

---

afterLogoff	(Optional) Accepts as a value the following JavaScript functions on the Window object: closeWindow   forward
forwardUrl	(Optional) URL to which to forward the browser

---

---

# main

**Purpose** The main component is an abstract application entry point. The default Web page in the application virtual root should redirect to this component. Override the main component to handle all timeouts and history releases, which by default are redirected to this component.

## Configuration File

wdk\config\main\_component.xml

## Scope

All.

## Usage

The default URL for Webtop is `http://server:port/webtop`. This URL redirects to the main component. To redirect your default application page to the main component, use the following JavaScript function in your `default.html` or `index.html` page:

```
<script>
function redirect()
{
//Get the virtual dir via the window.location.pathname
var strPath = window.location.pathname;
var nIndex1 = strPath.indexOf("/");
var nIndex2 = strPath.indexOf("/", nIndex1 + 1);
var strVirtualDir = strPath.substring(
    nIndex1 + 1, nIndex2);
window.location.replace("/"
    + strVirtualDir + "/component/main");
}
</script>
```

# managequeueinbox

**Purpose** Lists all uncompleted tasks in a work queue, both assigned and unassigned.

## Configuration File

webcomponent\config\library\queuemanagement\managequeueinbox\_component.xml

## Scope

All.

## Caller

This component is called by the action [managequeueinbox](#) (type `dmc_workqueue`), [page 333](#).

## Parameters

---

categoryPath	(Optional) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs.
queueName	(Required) Name of the work queue.

---

## Elements

This component has configurable columns. (See [doclist](#), [page 684](#) for a description.)

---

<failoverenabled>	This component supports failover.
-------------------	-----------------------------------

---

---

## menubar

**Purpose** Adds a bar of menu controls

### Configuration File

webcomponent\config\library\menubar\menubar\_component.xml

### Scope

All.

### Parameters

---

accessible	(Optional) Set to true to render menu selections accessible (Section 508 compliant)
------------	---

---

## messages

**Purpose** Displays a data grid of all the user's messages for a session. The error message service passes non-fatal error messages to the messages component for display.

### Configuration File

webcomponent\config\library\messages\messages\_component.xml

### Scope

All.

## Usage

The messages component uses the message service to get the result set of messages and initialize a data grid with the result set. For example:

```
Datagrid msgGrid = (Datagrid)getControl(  
    CONTROL_GRID, Datagrid.class);  
ResultSet res = MessageService.getResultSet();  
msgGrid.getDataProvider().setScrollableResultSet(  
    ScrollableResultSet)res)
```

# modalerrordialog

**Purpose** This component provides the UI used to present modal error dialogs to the user.

## Configuration File

webcomponent\config\library\modalerrordialog\modalerrordialog\_component.xml

## Scope

All.

## Parameters

---

forceModal	(Optional) Boolean: If this component is being directed to a non-modal frame, set to true to set up the forced modality infrastructure.
------------	---

---

# modifyversionlabels

**Purpose** This component provides the UI and functionality to allow users to modify version labels on objects.

## Configuration File

webcomponent\config\library\modifyversionlabels\modifyversionlabels\_component.xml

## Scope

All.

## Caller

This component is called by the action [modifyversionlabels](#) (type `dm_sysobject`), page 335 and displayed in the component [combocontainer](#), page 644.

## Parameters

nodeId	(Optional) ID of the node if it is part of a virtual document
objectId	(Required) (Required) ID of the object being modified

# movetoqueuecontainer

**Purpose** Moves one or more selected queued tasks to another work queue.

## Configuration File

webcomponent\config\actions\movetoqueuecontainer\_component.xml

## Parent Definition

multiargdialogcontainer:webcomponent/config/library/multiargdialogcontainer/multiargdialogcontainer\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [move\\_queued\\_task](#) (type `dm_task`), [page 336](#) and displays the component [assignqueuedtask](#), [page 571](#).

**Parameters**

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## multiargdialogcontainer

**Purpose**

Extends the container [dialogcontainer](#), [page 669](#) and allows multiple sets of arguments (following a multi-select in classic view, for example) to be used by a single component contained in this container. Contains [sendtodistributionlist](#), [page 1009](#).

**Configuration File**

webcomponent\config\library\multiargdialogcontainer\multiargdialogcontainer\_component.xml

**Parent Definition**

dialogcontainer:wdk/config/dialogcontainer\_component.xml



**Scope**

All.

**Caller**

This component is called by the action [sendtodistributionlist \(type dm\\_sysobject\), page 421](#) and displays the component [sendtodistributionlist, page 1009](#).

**Parameters**


---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## multidocbasetreetest

**Purpose**

One of a set of components and actions that test controls and actions. This component is used for internal testing at Documentum.

**Configuration File**

wdk\config\fxtest\_component.xml

**Scope**

All.

# mydocumentlocator

**Purpose** Extends [myobjectlocator, page 851](#) and displays all objects of type `dm_document` or its subtypes that are owned by the user. The scope is not defined, but the `<objecttype>` element restricts this component to `dm_document` objects.

## Configuration File

`webcomponent\config\library\locator\documentlocators_component.xml`

## Parent Definition

`myobjectlocator:webcomponent/config/library/locator/myobjectlocator_component.xml`

## Scope

All.

## Parameters

These parameters are inherited from `myobjectlocator:webcomponent/config/library/locator/myobjectlocator_component.xml`.

---

<code>flatlist</code>	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<code>multiselect</code>	(Optional) Boolean: Set to true to support multiple object selection.
<code>objecttype</code>	(Optional) Type of the selected object

---

## Elements

The configurable elements for this component are inherited from [sysobjectlocator, page 1029](#).

# myobjectlocator

**Purpose** Extends [sysobjectlocator](#), page 1029 and displays all sysobjects that are owned by the user.

## Configuration File

webcomponent\config\library\locator\myobjectlocator\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Parameters

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.

<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document

---

<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<columns>	Contains <column> elements to display or hide attributes

<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

---

## myobjects\_drilldown

**Purpose** Displays a folder frame and a folder contents frame for folders that contain objects owned by the user.

### Configuration File

webcomponent\config\library\myobjects\myobjects\_drilldown\_component.xml

### Scope

All.

### Elements

This component has configurable columns. (See [doclist, page 684](#) for a description.)

---

<objecttype>	Sets the user preference for object type to be displayed
<showfolders>	Sets the user preference for whether to display folders
<modifiedwithindays>	Integer that specifies the number of days to search for modified files that belong to the user

---

## myobjects\_list

**Purpose** Lists objects that are owned by the user.

### Configuration File

webcomponent\config\library\myobjects\myobjects\_list\_component.xml

### Scope

All.

### Elements

This component has configurable columns. (See [doclist](#), page 684 for a description.) The following additional preferences are defined:

---

<objecttype>	Sets the user preference for object type to be displayed
<showfolders>	Sets the user preference for whether to display folders
<modifiedwithindays>	Integer that specifies the number of days to search for modified files that belong to the user

---

## myrooms\_drilldown

**Purpose** My Rooms Streamline view component, used to display the rooms to which the user belongs.

### Configuration File

webcomponent\config\library\rooms\myrooms\_drilldown\_component.xml

### Parent Definition

myobjects\_drilldown:/webcomponent/config/library/myobjects/myobjects\_drilldown\_component.xml

### Scope

All.

## myrooms\_list

**Purpose** My Rooms Classic view component, used to list rooms of which the user is a member.

### Configuration File

webcomponent\config\library\rooms\myrooms\_list\_component.xml

### Parent Definition

myobjects\_list:/webcomponent/config/library/myobjects/myobjects\_list\_component.xml



**Scope**

All.

## mysavedsearches

**Purpose** Lists all dm\_smartlist objects that are owned by the user. The user can revise , execute, view properties, and delete.

**Configuration File**

webcomponent\config\library\search\searchex\mysavedsearches\_component.xml

**Scope**

All.

## myuserwftemplatelocator

**Purpose** Extends [allwftemplatelocator, page 540](#) and lists all non-system workflow templates owned by the current user.

**Configuration File**

webcomponent\config\library\workflow\myuserwftemplatelocator\_component.xml

**Parent Definition**

allwftemplatelocator:webcomponent/config/library/locator/wftemplatelocators\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.

<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)

<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<columns>	Contains <column> elements to display or hide attributes
<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'>. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

---

## mywftemplatelocator

**Purpose** Extends [myobjectlocator](#), page 851 and displays a flat list of workflow templates that have been checked out or recently modified by the current user.

## Configuration File

webcomponent\config\library\locator\wftemplatelocators\_component.xml

## Parent Definition

myobjectlocator:webcomponent/config/library/locator/myobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from myobjectlocator:webcomponent/config/library/locator/myobjectlocator\_component.xml.

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

For information on configurable columns, see [doclist, page 684](#).

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements

<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.

---

<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<columns>	Contains <column> elements to display or hide attributes
<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

---

# navigationcontainer

**Purpose** Displays a breadcrumb, title, and drop-down list above an included component to support navigation to an object from the included component. This container component is a base component and must be extended in order to add components.

## Configuration File

```
webcomponent\config\navigation\navigationcontainer\navigationcontainer_
component.xml
```

## Scope

All.

## Caller

This component is called by the actions [locations](#) (type `dm_sysobject`), [page 327](#), [relationships](#) (type `dm_sysobject`), [page 381](#), [renditions](#) (type `dm_sysobject`), [page 397](#), and [versions](#) (type `dm_sysobject`), [page 456](#), and displays the components [locations](#), [page 837](#), [relationships](#), [page 940](#), [renditions](#) (type `dm_sysobject`), [page 955](#), and [versions](#) (type `dm_sysobject`), [page 1098](#).

## Parameters

---

component	(Required) Contained component.
folderId	(Optional) Specifies the primary location of the specified folder Id
folderPath	(Optional) Specifies the path to the folder in which to start browsing. This path must be a complete path from the repository root, for example, /Documentation/Library/subfolder
objectId	(Required) Selected object.

---



---

# nestedcomponentcontainer

**Purpose** Extends the container [dialogcontainer](#), page 669 and displays **Close** and **Help** buttons for the component that is nested within this container.

## Configuration File

wdk\config\nestedcomponentcontainer\_component.xml

## Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from container:wdk/config/container\_component.xml.

---

component	(Required) The contained component. Set the requiresVisit attribute to true to require the component to be visited before an OK button is displayed.
-----------	--

---

# newassembly

**Purpose** Creates a new virtual document snapshot. The user must have a contributor role assigned in the repository. The container extends newdoccontainer and contains the newassembly component as well as the attributes and permissions components, to set attributes and permissions on the new snapshot.

## Configuration File

webcomponent\config\library\vdm\newassembly\newassembly\_component.xml

## Scope

All.

## Caller

This component is launched by the action [newassembly \(type dm\\_sysobject\), page 340](#) and is displayed in the component [newassemblycontainer, page 866](#).

## Parameters

---

folderId	(Optional) ID of the folder in which to create the assembly
nodeId	(Optional) ID of root node in the virtual document
objectId	(Required) ID of the virtual document.

---

# newassemblycontainer

**Purpose** Contains component used to create a new assembly and set its attributes and permissions.

## Configuration File

webcomponent\config\library\vdm\newassembly\newassemblycontainer\_component.xml

## Parent Definition

newdoccontainer:webcomponent/config/library/create/newdoccontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [newassembly](#) (type `dm_sysobject`), page 340 and displays the component [newassembly](#), page 865.

## Parameters

These parameters are inherited from `newdoccontainer:webcomponent/config/library/create/newdoccontainer_component.xml`.

<code>component</code>	(Required) Specifies the component to have initial focus
<code>contentType</code>	(Optional) Content type of the document being created.
<code>editAfterCreate</code>	(Optional) Boolean: This parameter is ignored by the implementation class and should always be set to false.

## Elements

<code>&lt;newcomponentname&gt;</code>	Specifies the name of the component that will be launched to create the new object
---------------------------------------	--

# newcabinet

## Purpose

Creates a new cabinet. This component must run within the component [newcabinetcontainer](#), page 868.

## Configuration File

`webcomponent\config\library\create\newcabinet_component.xml`

**Scope**

All.

**Caller**

This component is called by the action newcabinet.

**Elements**

---

<code>&lt;base_type&gt;</code>	Sets the base type for a new cabinet. Subtypes will also be displayed. Default: dm_cabinet
<code>&lt;default_type&gt;</code>	Sets the default selection for new cabinet type. Must be the base type or one of its subtypes. Default: dm_cabinet

---

## newcabinetcontainer

**Purpose**

Extends [propertywizardcontainer](#), page 915 and contains attributes, newcabinet, and permissions. The newcabinet component requires a visit.

**Configuration File**

webcomponent\config\library\create\newcabinetcontainer\_component.xml

**Parent Definition**

propertywizardcontainer:webcomponent/config/library/  
propertywizardcontainer/propertywizardcontainer\_component.xml

**Scope**

All.

## Caller

This component is called by the action newcabinet.

## Parameters

component	(Required) Supplied by the action execution class). The name of the component to contain.
-----------	---

## Elements

<newcomponentname>	Specifies the name of the component that will be launched to create the new object
--------------------	--

## Usage

The container parameters are added by the LaunchComponent action execution class in order to support actions on multiple objects. Do not call the container by URL.

# newdoccontainer

**Purpose** Extends [propertywizardcontainer](#), page 915 and contains attributes, the component [newdocument](#), page 871, and permissions. The component [newdocument](#), page 871 requires a visit.

## Configuration File

```
webcomponent\config\library\create\newdoccontainer_component.xml
```

## Parent Definition

```
propertywizardcontainer:webcomponent/config/library/  
propertywizardcontainer/propertywizardcontainer_component.xml
```

## Scope

All.

## Caller

This component is called by the [newdocument](#) (type `dm_folder`), [page 342](#)action.

## Parameters

---

<code>component</code>	(Required) (Required) ID of the component displayed
<code>contentType</code>	(Optional) Content type of document being created.
<code>editAfterCreate</code>	(Optional) Boolean: Set to true to open the new document object in edit mode after it has been created.

---

## Elements

<code>&lt;newcomponentname&gt;</code>	Specifies the name of the component that will be launched to create the new object
---------------------------------------	--

## Usage

For performance reasons, validation is not performed on new document creation. The validation is performed only when the document is checked in. If the user creates a document with no format (zero content), then validation is done, because the document can not be edited and is not checked out.

Both `NewContainer` and `NewDocContainer` determine the component name by calling a protected member function, `getNewComponentName()`. You can override this function to return the component name. For example, the `NewAssemblyContainer` java class, which derives from `NewDocContainer`, overrides the `getNewComponentName()` function to return "newassembly" as the contained component name.

The parameters for `combocontainer` (parent of `newdoccontainer`) are added by the `LaunchComponent` action execution class in order to support actions on multiple objects. Do not call the container by URL.

# newdocument

**Purpose** Creates new objects. This component must run within the component [newdoccontainer, page 869](#) or the component [newhttpdoccontainer, page 874](#). Each custom object must have a template in the repository Templates cabinet in order for the template and its associated formats to show up in the newdocument UI.

## Configuration File

webcomponent\config\library\create\newdocument\_component.xml

## Scope

All.

## Caller

This component is called by the actions [newdocument \(type dm\\_folder\), page 342](#) and [newvirtualdocumentnode \(type dm\\_sysobject\), page 349](#) and displayed in the containers [newdoccontainer, page 869](#) and [newhttpdoccontainer, page 874](#).

## Parameters

---

objectId	(Required) ID of the folder that will contain the new document
----------	--

---

## Elements

You can configure the preferred default selected object type and format and whether to display templates for supertypes of formats that do not have a template in the current repository. The following table describes the elements that you can configure within the <combo\_defaults> element.

---

type	Default type to display in type combo box. Valid values: dm_sysobject or subtype
format	Default format to display in format combo box. A template of this format must be available in the /Templates cabinet or subfolder. Valid values: 'name' attribute values for dm_format object

---

## Usage

Refer to [newdoccontainer](#), page 869.

# newfolder

**Purpose** Creates a new folder. This component must run within the component [newfoldercontainer](#), page 873.

## Configuration File

webcomponent\config\library\create\newfolder\_component.xml

## Scope

All.

## Caller

This component is called by the action [newfolder](#) (type dm\_folder), page 343.

## Parameters

---

objectId	(Required) ID of the folder or cabinet that will contain the new folder
----------	---

---



## Elements

<base_type>	Sets the base type for a new folder. Subtypes will also be displayed. Default: dm_folder
<default_type>	Sets the default selection for new folder type. Must be the base type or one of its subtypes. Default: dm_folder

# newfoldercontainer

**Purpose** Extends [propertywizardcontainer](#), page 915 and contains attributes, newfolder, and permissions. The newfolder container requires a visit.

## Configuration File

webcomponent\config\library\create\newfoldercontainer\_component.xml

## Parent Definition

propertywizardcontainer:webcomponent/config/library/  
propertywizardcontainer/propertywizardcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [newfolder](#) (type dm\_folder), page 343.

## Parameters

---

component	(Required): Supplied by the action execution class). The name of the component to contain.
-----------	--

---

## Elements

---

<newcomponentname>	Specifies the name of the component that will be launched to create the new object
--------------------	--

---

# newhttpdoccontainer

**Purpose** Extends [newdoccontainer](#), page 869.

## Configuration File

webcomponent\config\library\httpcreate\newhttpdocument\_component.xml

## Parent Definition

newdoccontainer:webcomponent/config/library/create/newdoccontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [newdocument](#) (type `dm_folder`), page 342 and [newvirtualdocumentnode](#) (type `dm_sysobject`), page 349, and displays the component [newdocument](#), page 871.

## Parameters

These parameters are inherited from `newdoccontainer:webcomponent/config/library/create/newdoccontainer_component.xml`.

---

<code>component</code>	(Required) (Required) ID of the component displayed
<code>contentType</code>	(Optional) Content type of the document being created.
<code>editAfterCreate</code>	(Optional) Boolean: Set to true to open the new document object in edit mode after it has been created.

---

# newnotepage

**Purpose** This component is used to create a new notepage object. Requires Collaborative Edition.

## Configuration File

`webcomponent\config\library\notepage\newnotepage_component.xml`

## Scope

All.

## Caller

This component is called by the action.

## Parameters

---

<code>objectId</code>	(Required) ID of the folder in which to create the new notepage object
-----------------------	--

---

## Elements

---

<objecttype>	Type of notepage object to create
--------------	-----------------------------------

---

# newnotepagecontainer

**Purpose** This component is used to display the newnotepage component. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\notepage\newnotepagecontainer\_component.xml

## Parent Definition

propertywizardcontainer:webcomponent/config/library/  
propertywizardcontainer/propertywizardcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action .

## Parameters

---

component	(Required) ID of the component to be displayed
-----------	--

---

## Elements

<newcomponentname>	Specifies the name of the component that will be launched to create the new object
--------------------	--

## newroom

**Purpose** Creates a new room: its name, description, and default access controls. Requires Collaborative Edition. This component must run within the container [newroomcontainer](#), page 877.

### Configuration File

webcomponent\config\library\rooms\newroom\_component.xml

### Scope

All.

### Caller

This component is called by the action newroom.

### Parameters

objectId	(Optional) (Required) Object ID of the cabinet in which the new room is created.
----------	--

## newroomcontainer

**Purpose** Container component for creating new eRooms. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\rooms\newroomcontainer\_component.xml

## Parent Definition

propertywizardcontainer:webcomponent/config/library/  
propertywizardcontainer/propertywizardcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action newroom.

## Parameters

These parameters are inherited from propertywizardcontainer:webcomponent/  
config/library/propertywizardcontainer/propertywizardcontainer\_  
component.xml.

---

component	(Required) Name of the contained component to display first.
-----------	--

---

## Elements

---

<newcomponentname>	Specifies the name of the component that will be launched to create the new object
--------------------	--

---

# newroomgroup

## Purpose

Creates a private group in a room. Requires Collaborative Edition.

**Configuration File**

webcomponent\config\library\rooms\newroomgroupcontainer\_component.xml

**Parent Definition**

objectlocator:webcomponent/config/library/locator/objectlocator\_component.xml

**Scope**

All.

**Parameters**

---

groupName	(Optional) ID of group to which the selection will be restricted
objectId	(Optional) (Required) Object ID of the room in which the private group is created

---

## newroomgroupcontainer

**Purpose** Container component for creating private groups in a room. Requires Collaborative Edition.

**Configuration File**

webcomponent\config\library\rooms\newroomgroupcontainer\_component.xml

**Parent Definition**

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

**Scope**

All.

**Parameters**

---

component	(Required) Name of the contained component to display first.
groupName	(Optional) ID of group to be created
objectId	(Required) Object ID of the room in which the private group is created.

---

## newwindow

**Purpose** Opens a second WDK application browser window.**Configuration File**

webcomponent\config\library\newwindow\newwindow\_component.xml

**Scope**

All.

**Caller**This component is called by the action [newwindow](#), page 350.**Elements**

---

<windowparams>	Specifies configuration for the new window. The parameters specified for this element are used when opening a new window using JavaScript.
----------------	--

---



## newxforms

**Purpose** This component is used to create a new form. The user must have the role `form_user`.

### Configuration File

`webcomponent\config\xforms\create\newxforms_component.xml`

### Scope

All.

### Caller

This component is called by the action [newxforms \(type dm\\_folder\), page 354](#) and displayed within the component [newxformscontainer, page 881](#).

### Parameters

---

<code>objectId</code>	(Required) Object ID of the current folder in which the form will be created.
-----------------------	---

---

## newxformscontainer

**Purpose** Displays the component [newxforms, page 881](#) to create a new form. It extends [wizardcontainer, page 1148](#).

### Configuration File

`webcomponent\config\xforms\newxformscontainer_component.xml`

## Parent Definition

wizardcontainer:wdk/config/wizardcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [newxforms](#) (type `dm_folder`), page 354 and [saveasxforms](#) (type `dm_sysobject`), page 412, and displays the components [newxforms](#), page 881 and [saveasxforms](#), page 986.

## Parameters

These parameters are inherited from `container:wdk/config/container_component.xml`.

---

<code>component</code>	(Required) The contained component. Set the <code>requiresVisit</code> attribute to <code>true</code> to require the component to be visited before an OK button is displayed.
------------------------	--

---

# nodemanagement

**Purpose** Handles nodes on the tree that have subnodes. It reproduces the nodes as a list in the content area.

## Configuration File

`webcomponent\config\admin\nodemanagement\nodemanagement_component.xml`

## Scope

All.

**Caller**

This component is not called from WDK actions or from other WDK components.

**Parameters**


---

nodeIds	(Optional) Target nodes in the tree
---------	-------------------------------------

---

## nomaclsupport\_permissions

**Purpose**

This is a backup ID for WDK 5.3 that passes the arguments to the standard permissions component.

**Configuration File**

webcomponent\config\library\permissions\extendedpermissions\_component.xml

**Parent Definition**

permissions:webcomponent/config/library/permissions/permissions\_component.xml

**Scope**

All.

**Parameters**

These parameters are inherited from permissions:webcomponent/config/library/permissions/permissions\_component.xml.

---

objectId	(Required) ID of the object
----------	-----------------------------

---

## notepage (type dmc\_notepage)

**Purpose** This component is used to display a note object. A note is a content file that can be displayed directly in the browser. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\notepage\notepage\_component.xml

### Scope

---

type	dmc_notepage
------	--------------

---

### Caller

This component is called by the action .

### Parameters

---

folderId	(Optional) ID of the folder in which the notepage is located
folderPath	(Optional) Specifies the ID of the primary location for the folder in which the notepage is located
objectId	(Required) ID of the notepage to be displayed

---

## objectgrid

**Purpose** Renders the result of a query attached to a data grid.

## Configuration File

webcomponent\config\navigation\objectgrid\objectgrid\_component.xml

## Scope

All.

## Elements

---

<header>	Contains header settings <showpagingcontrol> and <showclosebutton>. Set the visible attribute to true to display a header when the objectgrid component is used as standalone. Sub-components can override this element to configure the header.
<showpagingcontrol>	True to display a paging control within the header
<showclosebutton>	True to display a close button within the header
<nondocbasecolumns>	Contains <column> elements that are databound or otherwise provided by the component class but are not included in the query

---

The default columns to be displayed are configured in the <columns> element. For information on this configuration, see [doclist, page 684](#).

## Usage

The objectgrid component and the datagrid control can both display objects. The objectgrid component renders attributes for a single object type. Some examples of components that extend objectgrid to display objects are locations, relationships, renditions, versions, and history.

In Webtop, components that extend object grid component display a subset of the dm\_sysobject attributes and the folder path for the objects in the result set. If your component doesn't need to display attributes other than those of dm\_sysobject, you can use the default object grid JSP pages. To display other dm\_sysobject attributes, you must extend this component and modify the JSP page data fields to display the attributes you wish.

Components that extend ObjectGrid provide the query string by implementing `getQuery(String strVisibleAttrs, ArgumentList args)`. You must also provide custom JSP pages that display the custom attributes from your query. For an example of a customized attribute display of the query resultset, see `/webcomponent/library/relationships/relationships.jsp`.

Sample override of `getQuery` (error handling removed):

```
protected String geQuery(String strVisibleAttrs, ArgumentList args)
{
    String strQuery = null;
    String strCustomAttr = args.get("customAttr");
    strQuery = new StringBuffer(512)
        .append("SELECT" " ")
        .append(strVisibleAttrs)
        .append(INTERNAL_attrs)
        .append(" from mycustomobject(all) ")
        .append("WHERE mycustomattr = ")
        .append(strCustomAttr)
        .append("'') "
        .toString();
    return strQuery;
}
```

## objectlocator

### Purpose

The locator components locate objects in the repository. You can include locators in your component class when you need to locate Documentum objects. Locator components locate objects by type. Some locators search on specific criteria within the type, such as objects that are subscribed to or belong to the user. Two types of views are supported: *flatlist*, in which all selectable objects in the repository are shown, and *hierarchical*, in which the user can drill down from a list of root container (cabinets).

### Configuration File

`webcomponent\config\library\locator\objectlocator_component.xml`

### Scope

All.

## Parameters

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.

<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)



---

<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute <code>dqlformatted='false'</code> to quote and escape a value. Use the attribute <code>casesensitive='true'</code> to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## Usage

The basic behavior of the locator is defined in `com.documentum.webcomponent.library.locator.ObjectLocator`. Each locator is driven by a query builder object defined in `com.documentum.webcomponent.library.locator.ObjectLocatorQuery`, which provides a base query for the selectable styles. The locator can refine the query by applying name-matching and attribute filters.

The object found by the locator is represented by `com.documentum.webcomponent.library.locator.LocatorItemResultSet`. To provide alias attributes, such as container path, you can use a query-build object to redefine the result set.

The `locatoritemresultset` component represents the objects returned by the locator query. The constructor constructs the result set and specifies extra columns for dummies and aliases.

## permissions

**Purpose** Sets extended permissions on objects using Trusted Content Services when that feature is enabled for the repository

### Configuration File

`webcomponent\config\library\permissions\extendedpermissions_component.xml`

## Scope

All.

## Parameters

---

objectId	(Required) ID of the object
----------	-----------------------------

---

## Elements

---

<nomacsupport>	Specifies the name of the component to use when Trusted Content Services is not enabled for the repository. Default: permissions component
----------------	--

---

# permissions (version 5.2.5)

**Purpose** Sets permissions on objects. This component is the 5.2.5 version and cannot be launched directly. If your custom component extends this component, it will continue to work until you migrate it to 5.3.

## Configuration File

webcomponent\config\library\permissions\permissions\_component.xml

## Scope

---

version	5.2.5
---------	-------

---

## Caller

This component is called by the action [permissions](#) (type `dm_sysobject`), [page 359](#) and displayed in the component [properties](#), [page 908](#).

## Parameters

objectId	(Required) ID of the ACL
----------	--------------------------

# persistentobjectlocator

## Purpose

Locates and selects any persistent object in the repository. It is a generic locator which can be configured to locate any object in the repository.

## Configuration File

`webcomponent\config\library\locator\persistentobjectlocator_component.xml`

## Scope

All.

## Parameters

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

For information on configurable columns, see [doclist](#), [page 684](#).

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document

<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## policyattributes

**Purpose** Sets the attributes for a work queue policy.

## Configuration File

webcomponent\config\library\queuemanagement\policyattributes\_component.xml

## Scope

All.

## Caller

This component is called by the actions [attributes](#) (type `dmc_workqueue_policy`), page 237, [newWorkQueuePolicy](#) (type `dmc_workqueue_policy`), page 353, and [properties](#) (type `dmc_workqueue_policy`), page 372, and displayed in the component [workqueuepolicyproperties](#), page 1157.

## Parameters

---

objectId	(Required) ID of the work queue policy object whose attributes to set.
----------	--

---

# policylist

**Purpose** Displays a list of the available work queue policies.

## Configuration File

webcomponent\config\library\queuemanagement\list\policylist\_component.xml

## Scope

All.

## Elements

This component has configurable columns. (See [doclist](#), page 684 for a description.)

---

## portal\_login

**Purpose** Enables logging in to a portal.

### Configuration File

webcomponent\config\library\portallogin\portallogin\_component.xml

### Scope

All.

### Parameters

---

docbase	(Optional) For internal use only.
startComponent	(Optional) For internal use only.
startPage	(Optional) For internal use only.
startUrl	(Optional) For internal use only.

---

## portal\_preferences

**Purpose** Enables users to set preferences for portals..

### Configuration File

webcomponent\config\environment\preferences\portal\portal\_preferences\_component.xml

### Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

**Scope**

All.

**Parameters**

These parameters are inherited from propertycontainer:wdk/config/propertycontainer\_component.xml.

---

component	(Required) For internal use only.
-----------	-----------------------------------

---

## portalsearch

**Purpose** Enables searching repositories for portal users. For internal use only.

**Configuration File**

webcomponent\config\library\search\searchex\portalsearch\_component.xml

**Scope**

All.

**Parameters**

---

drilldown	(Optional) For internal use only.
showwait	(Optional) For internal use only.

---

## portalsearchcontainer

**Purpose** Container for the portalsearch component. For internal use only.



**Configuration File**

webcomponent\config\library\search\searchex\portalsearchcontainer\_component.xml

**Parent Definition**

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

**Scope**

All.

**Parameters**

drilldown	(Optional) For internal use only.
showwait	(Optional) For internal use only.

**portlet\_preferences**

**Purpose** Enables users to set preferences for portlets. For internal use only.

**Configuration File**

webcomponent\config\environment\preferences\portal\portlet\portlet\_preferences\_component.xml

**Parent Definition**

dialogcontainer:wdk/config/dialogcontainer\_component.xml

**Scope**

All.

## Parameters

These parameters are inherited from container:wdk/config/container\_component.xml.

---

component	(Required) For internal use only.
-----------	-----------------------------------

---

# preferences

**Purpose** Extends the property sheet container. Launched by the preferences action to contain several preferences components.

## Configuration File

webcomponent\config\environment\preferences\preferences\_component.xml

## Parent Definition

property sheetcontainer:wdk/config/property sheetcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [preferences, page 359](#) and displays the component [general\\_preferences, page 743](#).

## Parameters

These parameters are inherited from property sheetcontainer:wdk/config/property sheetcontainer\_component.xml.

---

component	(Required) The contained component. Set the required Visit attribute to true to require the component to be visited before an OK button is displayed.
-----------	---

---

## Elements

---

<failoverenabled>	Controls serialization of the component state, and notifies the component in case of a failover recovery. Failover is enabled if this element is set to true.
<ucfrequired>	If this element is present, then the component requires UCF to be installed.

---

# processdetailreportresults

**Purpose** Displays the results of a workflow process

## Configuration File

webcomponent\config\library\workflow\historicalreport\processdetailresults\_  
component.xml

## Scope

All.

## Parameters

---

process_id	(Required) ID of the workflow process (template)
query	(Required) When the querytype is string, this value is the search string. If the querytype is not string, this value is the query string.
title	(Optional) Title for the report

---

# processorattributes

**Purpose** Sets the queue-related attributes for a work queue processor.

## Configuration File

webcomponent\config\library\queuemanagement\processorattributes\_component.xml

## Scope

All.

## Parameters

---

objectId	(Required) ID of the user acting as a work queue processor.
objectname	(Required) Name of the user.

---

---

# processorauthenticationlocatorcontainer

**Purpose** Container component for specifying the work queue document profile, skill level, and work queues for a newly defined work queue processor.

## Configuration File

webcomponent\config\library\queuemanagement\processorauthenticationlocator\_component.xml

## Parent Definition

workqueuelocatorcontainer:webcomponent/config/library/queuemanagement/workqueuelocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a navigation tree.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the object to include in the list

repository	(Optional) Name of the repository to include objects from
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## processorauthenticationselect

**Purpose** Sets the skill level and document profile settings as part of adding a new work queue processor. The component saves the work queue processor profile.

### Configuration File

webcomponent\config\library\queuemanagement\processorauthenticationlocator\_component.xml

### Scope

All.

## processorproperties

**Purpose** Sets the properties for a work queue processor.

### Configuration File

webcomponent\config\library\queuemanagement\processorproperties\_component.xml

**Parent Definition**

adminpropertycontainer:webcomponent/config/admin/container/  
adminpropertycontainer\_component.xml

**Scope**

All.

**Parameters**

component	(Required) (Required) ID of the component to display
objectId	(Required) ID of the user object for the work queue processor.
objectname	(Required) Name of the user

## processorworkloadinbox

**Purpose** Provides a queue manager with a list of the tasks awaiting a work queue processor.

**Configuration File**

webcomponent\config\actions\processorworkloadinbox\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [processorworkloadinbox](#), page 363.

## Parameters

---

categoryPath	(Optional) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs.
objectname	(Required) ID of the user object for the work queue processor.
queueName	(Required) Name of the work queue.

---

## Elements

---

<showfilters>	Set to false to suppress the objectfilters drop-down control.
<showattachmentinfo>	Set to true to show the names of attachments as part of the task name.
<columns>	Configures the columns displayed in the Inbox. The first column, displaying taskName, is always visible.

---

For information on configurable columns, see [doclist, page 684](#).

# prompt

## Purpose

Provides the functionality of a message box. A prompt has a title, a message, and a selection of buttons: OK, Cancel, Continue, Yes, No, Yes to all, No to all, and Help, any combination of which can be displayed. There is also an optional icon and "Don't show this again" checkbox. The don't show again feature is enabled only when the prompt component is used within the combocontainer component.

## Configuration File

wdk\config\prompt\_component.xml



## Scope

All.

## Parameters

button	(Required) The name of the button that will be displayed. Multiple values may be passed.
dontshowagain	Boolean flag to display a checkbox that will hide the prompt the next time the context of the prompt occurs. The actual handling of this flag must be implemented in the caller component.
icon	Name of icon class
message	String message in the prompt frame
title	String title for the prompt frame

## Usage

The prompt component can either be extended, in order to set the message and title or modify its behavior, or the component can be driven directly through arguments. In the following example, the prompt component is nested within another component, and the user is returned to the parent component after responding to the prompt:

```
ArgumentList args = new ArgumentList();
args.add(Prompt.ARG_TITLE, getString(strTitleId));
args.add(Prompt.ARG_MESSAGE, getString(strMessageId));
args.add(Prompt.ARG_ICON, Prompt.ICON_WARNING);
args.add(Prompt.ARG_BUTTONS, Prompt.CONTINUE + "," +
    Prompt.CANCEL);
args.add(Prompt.ARG_DONTSHOWAGAIN,
    "true");
setComponentNested("prompt", args, getContext());
```

For an example of the usage of the prompt component, see the combo container. The prompt component is displayed when a user clicks OK before the last page has been displayed.

### Handling DONTSHOWAGAIN

If you want to give the user the option of turning off the prompt when the prompt context arises, your calling component must set the dontshowagain argument and then handle the user's selection. You should store the selection as a preference and retrieve it in your component.

The following example sets up the prompt component based on the user's preference:

```
public void onEvent(Control control, ArgumentList args)
{
    ...
    // display prompt conditionally
    if (showPrompt())
    // see helper below
    {
        ArgumentList args = new ArgumentList();
        args.add(Prompt.ARG_TITLE, getString(TITLE));
        args.add(Prompt.ARG_MESSAGE, getString(MESSAGE));
        args.add(Prompt.ARG_ICON, Prompt.ICON_WARNING);
        args.add(Prompt.ARG_BUTTONS, Prompt.CONTINUE + "," +
            Prompt.CANCEL);
        args.add(Prompt.ARG_DONTSHOWAGAIN, "true");
        setComponentNested("prompt", args, getContext());
    }
}
```

The helper method gets the user preference:

```
private boolean showPrompt()
{
    IPreferenceStore preferences = PreferenceService.
        getPreferenceStore();
    return (preferences.readBoolean(
        INHIBIT_PROMPT_PREFERENCE) == null);
}
// unique preference string
final private static String INHIBIT_PROMPT_PREFERENCE
    = "    com.documentum...";
```

The component onRender() method checks the prompt return and whether to inhibit or show the prompt:

```
public void onRender()
{
    ...
    // check whether the prompt has returned
    String strButton = (String)getReturnedValue(
        Prompt.RTN_BUTTON);
    String strDontShowAgain = (String)getReturnedValue(
        Prompt.RTN_DONTSHOWAGAIN);

    if (strButton != null || strDontShowAgain != null)
    {
        // remove return values
        removeReturnedValue(Prompt.RTN_BUTTON);
        removeReturnedValue(Prompt.RTN_DONTSHOWAGAIN);
        if ( strButton.equals(Prompt.CONTINUE) )
        {
            // inhibit prompt if 'dont show again' was selected
            if ( strDontShowAgain != null && Boolean.valueOf(
                strDontShowAgain).booleanValue() )
            {
                inhibitPrompt();
                // see inhibitPrompt() helper below
            }
        }
    }
}
```

```

        }
        // perform task
    }
}

```

Another helper method writes the preference:

```

private void inhibitPrompt()
{
    IPreferenceStore preferences = PreferenceService.
        getPreferenceStore();
    preferences.writeBoolean(
        INHIBIT_PROMPT_PREFERENCE, Boolean.TRUE);
}

```

## promptinput

**Purpose** Prompts users for input.

### Configuration File

webcomponent\config\library\contenttransfer\promptinput\_component.xml

### Scope

All.

### Parameters

input	(Required) Text that users enter.
message	(Optional) Text that is displayed requesting input from users.
values	(Optional) Valid values that users can choose.

# promptinputcontainer

**Purpose** Container for components that prompt users for input.

## Configuration File

webcomponent\config\library\contenttransfer\promptinputcontainer\_component.xml

## Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml

## Scope

All.

## Parameters

---

component	(Required) Supplied by the action execution class). The name of the component to contain.
input	(Required) Text that users enter.
message	(Optional) Text that is displayed requesting input from users.
values	(Optional) Valid values that users can choose.

---

# properties

**Purpose** Extends the container [propertysheetcontainer](#), [page 914](#) and displays property sheets that show general information about the selected object, such as name, type, format and owner. The properties component contains attributes, history, and

---

permissions. The properties component for dm\_category objects extends this properties component.

## Configuration File

webcomponent\config\library\properties\properties\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [attributes \(type dm\\_sysobject\), page 234](#), [history \(type dm\\_task\), page 316](#), [permissions \(type dm\\_sysobject\), page 359](#), and [properties \(type dm\\_sysobject\), page 369](#). It displays the components [attributes \(type dm\\_sysobject\), page 590](#), [history, page 761](#), and [permissions, page 889](#).

## Parameters

---

component	(Required) The ID of the first contained component to be shown
objectId	(Required) The ID of the object for which properties are to be shown

---

## Elements

You can configure the properties container to display other components. The configurable elements for this component are inherited from [container, page 652](#). Additional element:

---

<commitorder>	Contains <component> elements that specify the order in which properties changes will be saved
<component>	Value specifies one or more contained component for which changes will be saved in the order the components are listed within <commitorder>
<setrepositoryfromobjectid>	Set to true to execute the query against the source repository. Affects reference and foreign objects.

---

## Usage

The following example adds the locations component to the list of properties for an object. Copy the properties component configuration file from `/webcomponent/config/library/properties` to your custom config directory. Add the locations component to the list of contained components:

```
<contains>
  <component>attributes</component>
  <component>permissions</component>
  <component>history</component>
  <component>locations</component>
</contains>
```

# properties (type dm\_category)

**Purpose** Extends the container [propertysheetcontainer](#), [page 914](#) and displays property sheets that show general information about the selected object, such as name, type, format and owner. The properties component contains attributes, history, and permissions. The properties component for dm\_category objects extends this properties component.

## Configuration File

`webcomponent\config\library\properties\dm_category_properties_component.xml`

**Parent Definition**

properties:webcomponent/config/library/properties/properties\_component.xml

**Scope**


---

type	dm_category
------	-------------

---

**Parameters**

These parameters are inherited from properties:webcomponent/config/library/properties/properties\_component.xml.

---

component	(Required) The ID of the first contained component to be shown
objectId	(Required) The ID of the object for which properties are to be shown

---

**Elements**

You can configure the properties container to display other components. The configurable elements for this component are inherited from [container, page 652](#).

**properties (type dm\_externalresult)**

**Purpose** This component is used to display attributes components.

**Configuration File**

webcomponent\config\library\properties\dm\_externalresult\_properties\_component.xml

**Parent Definition**

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

---

type	dm_externalresult
------	-------------------

---

## Parameters

---

component	(Required) ID of the component displayed
entryId	(Required) Internal key to the query results.
queryId	(Required) Internally used to identify the current search context

---

## Elements

You can configure the properties container to display other components. The configurable elements for this component are inherited from [container](#), page 652.

# properties (type dmc\_rm\_formal\_record)

**Purpose** Enables users to view and change properties of dmc\_rm\_formal\_record objects. Requires a Records Manager license.

## Configuration File

webcomponent\config\library\records\declarerecord\formalrecord\_properties\_component.xml

## Parent Definition

properties:webcomponent/config/library/properties/properties\_component.xml



**Scope**

type	dmc_rm_formal_record
------	----------------------

**Caller**

Called by the [properties \(type dm\\_sysobject\), page 369](#) , [attributes \(type dm\\_sysobject\), page 234](#), [history \(type dm\\_sysobject\), page 315](#), [permissions \(type dm\\_sysobject\), page 359](#) actions and runs the [and history, page 761](#) components.

**Parameters**

These parameters are inherited from `properties:webcomponent/config/library/properties/properties_component.xml`.

component	(Required) The ID of the first contained component to be shown
objectId	(Required) The ID of the object for which properties are to be shown

**properties (type dmc\_room)****Purpose**

Displays property sheets that show general information about the selected room object, such as name, type, format and owner. Requires Collaborative Edition.

**Configuration File**

`webcomponent\config\library\rooms\room_properties_component.xml`

**Parent Definition**

`properties:webcomponent/config/library/properties/properties_component.xml`

## Scope

---

type	dmc_room
------	----------

---

## Parameters

These parameters are inherited from `properties:webcomponent/config/library/properties/properties_component.xml`.

---

component	(Required) Name of the component to display first.
objectId	(Required) ID of the room whose properties to display.

---

# propertysheetcontainer

**Purpose** Extends [wizardcontainer](#), [page 1148](#) and contains attributes, history, and permissions.

## Configuration File

`wdk\config\propertysheetcontainer_component.xml`

## Parent Definition

`wizardcontainer:wdk/config/wizardcontainer_component.xml`

## Scope

All.

## Parameters

component	(Required) The contained component. Set the requiresVisit attribute to true to require the component to be visited before an OK button is displayed.
-----------	--

## Usage

Set requiresVisit attribute to true to declare that a particular component must be visited before the container can commit changes. In the following example, the attributes component must be viewed before the user can commit changes:

```
<contains>
  <component>newFolder</component>
  <component requiresVisit='true'>attributes</component>
  <component>permissions</component>
</contains>
```

You can also declare the visit requirement in the individual component definition. Add the following tag to a component definition:

```
<requiresVisitBeforeCommit>true</requiresVisitBeforeCommit>
```

# propertysheetwizardcontainer

## Purpose

Extends the container [propertysheetcontainer, page 914](#) and contains [propertysheetcontainer, page 914](#) and [wizardcontainer, page 1148](#). This container has **Previous** and **Next** buttons that move through pages in the currently selected components and another set of buttons that move through the contained components. The contained components are presented as horizontal tabs.

## Configuration File

```
webcomponent\config\library\propertysheetwizardcontainer\propertysheetwizardcontainer_
component.xml
```

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

All.

## Parameters

---

component	(Required) The contained component. Set the requiresVisit attribute to true to require the component to be visited before an OK button is displayed.
-----------	--

---

## Elements

---

<numberedtabs>	Set to true to display each tab's title with a sequential number. Default value is true in base container, false in derived container for Application Connectors,.
----------------	--

---

## Usage

If your component extends propertysheetwizardcontainer and you wish to show the contained components as vertical tabs, call setcomponentpage("vertical\_tabs") in your component class.

# qualifiedqueueprocessorslocator

**Purpose** Displays a selection list of users who are qualified to process tasks from a specific work queue.

## Configuration File

webcomponent\config\actions\qualifiedqueueprocessorslocator\_component.xml

## Parent Definition

wfuserorgroupfromgrouplocator:webcomponent/config/library/workflow/  
wfuserorgroupfromgrouplocator\_component.xml

## Scope

All.

## Parameters

groupName	(Required) Name of the dm_group object representing the work queue.
multiselect	(Optional) Boolean: Set to true to enable users to select multiple processors.
objecttype	(Optional) Object type of the users to display in the selection list; default is dm_user.

## Elements

For information on configurable columns, see [doclist](#), page 684.

# qualifiedqueueprocessorslocatorcontainer

## Purpose

Container component for selecting qualified queue processors. Contains the component [qualifiedqueueprocessorslocator](#), page 916.

## Configuration File

webcomponent\config\actions\qualifiedqueueprocessorslocator\_component.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Parameters

---

arrangeselect	(Optional) Not used.
groupName	(Required) Name of the group to which users must belong to process tasks in the queue.
multiselect	(Optional) Boolean: Set to true to enable users to select multiple processors.
objecttype	(Optional) Object type of the users to display in the selection list.

---

# queueattributes

**Purpose** Sets the attributes for a work queue.

## Configuration File

webcomponent\config\library\queuemanagement\queueattributes\_component.xml

## Scope

All.

## Caller

This component is called by the actions [attributes](#) (type `dmc_workqueue`), [page 238](#), [newWorkQueue](#) (type `dmc_workqueue`), [page 350](#), and [properties](#) (type `dmc_workqueue`), [page 373](#), and displayed in the component [workqueueproperties](#), [page 1158](#).

## Parameters

<code>categoryPath</code>	(Optional) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs.
<code>objectId</code>	(Required) ID of the work queue object.
<code>objectname</code>	(Optional) Name of the work queue

# queuelist

**Purpose** Displays a navigable list of the work queues and work queue categories.

## Configuration File

`webcomponent\config\library\queuemanagement\list\queuelist_component.xml`

## Scope

All.

## Parameters

<code>category_path</code>	(Optional) Not used.
<code>object_name</code>	(Optional) Not used.
<code>objectId</code>	(Optional) ID of the work queue category to display as the initial starting point for the list.

## Elements

This component has configurable columns. (See [doclist](#), page 684 for a description.)

# queuemanagementclipboard

**Purpose** Displays contents of the Queue Management clipboard.

## Configuration File

```
webcomponent\config\library\queuemanagement\queuemanagementclipboard_
component.xml
```

## Parent Definition

```
clipboard:webcomponent/config/environment/clipboard/clipboard_component.xml
```

## Scope

All.

## Parameters

None.

# queuemonitorlist

**Purpose** Displays overview information for monitoring the contents of work queues.

## Configuration File

```
webcomponent\config\library\queuemanagement\queuemonitor_component.xml
```



**Scope**

All.

**Elements**

This component has configurable columns. (See [doclist](#), page 684 for a description.)

## queueusersandgroups

**Purpose**

Displays the users and groups assigned to work on a work queue.

**Configuration File**

```
webcomponent\config\library\queuemanagement\queueusersandgroups_component.xml
```

**Scope**

All.

**Caller**

This component is called by the action [queueusersandgroups](#) (type `dmc_workqueue`), page 375.

**Parameters**

categoryPath	(Optional) Navigation path through work queue categories used to reach the current queue; used for setting the breadcrumbs.
objectId	(Required) ID of the work queue object.
queuename	(Required) Name of the work queue.

## recentcategorylocator

**Purpose** Locator for all recently selected categories.

### Configuration File

webcomponent\config\library\locator\categorylocator\_component.xml

### Parent Definition

recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_component.xml

### Scope

All.

### Parameters

These parameters are inherited from recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display in the list.

---

## recentdocumentlocator

**Purpose** Displays all objects that have been selected within the user's current HTTP session. This component has no scope, but <objecttype> element is configured by default to restrict search to dm\_document.

## Configuration File

webcomponent\config\library\locator\documentlocators\_component.xml

## Parent Definition

recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_component.xml.

contentType	(Optional) Content type for filetype filter
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Type of document to locate (default = dm_document)

## Elements

The configurable elements are inherited from [sysobjectlocator](#), page 1029.

# recentfolderlocator

## Purpose

Extends [recentsysobjectlocator](#), page 926 and locates folders that were recently selected or used. This component has no scope, but the <objecttype> element restricts the search by default to dm\_folder objects.

## Configuration File

webcomponent\config\library\locator\folderlocators\_component.xml

## Parent Definition

recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display in the list.

---

## Elements

The configurable elements are inherited from [sysobjectlocator, page 1029](#).

# recentgrouplocator

## Purpose

Extends [recentuserorgrouplocator, page 928](#) and locates user groups that were recently selected or used. This component has no scope, but <objecttype> element restricts the search to dm\_group objects.

## Configuration File

webcomponent\config\library\locator\grouplocator\_component.xml

## Parent Definition

recentuserorgrouplocator:webcomponent/config/library/locator/  
recentuserorgrouplocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from recentuserorgrouplocator:webcomponent/config/  
library/locator/recentuserorgrouplocator\_component.xml.

multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display in the list.

## Elements

The configurable elements are inherited from [sysobjectlocator](#), page 1029.

# recentlifecyclelocator

**Purpose** Locates lifecycles that were recently selected or used. This component has no scope, but the <objecttype> element restricts the search to objects of type dm\_policy.

## Configuration File

webcomponent\config\library\locator\lifecyclelocators\_component.xml

## Parent Definition

recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display in the list.

---

## Elements

For information on configurable columns, see [doclist](#), page 684. The configurable elements are inherited from [sysobjectlocator](#), page 1029.

# recentsysobjectlocator

**Purpose** Extends [objectlocator](#), page 886 and locates objects that were recently selected or used.

## Configuration File

webcomponent\config\library\locator\recentsysobjectlocator\_component.xml

## Parent Definition

objectlocator:webcomponent/config/library/locator/objectlocator\_component.xml

## Scope

All.

## Parameters

contentType	(Optional) Content type for filetype filter
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display in the list

## Elements

For information on configurable columns, see [doclist, page 684](#). The configurable elements are inherited from [sysobjectlocator, page 1029](#).

# recentuseronlylocator

## Purpose

Extends [recentuserorgrouplocator, page 928](#) and locates users that were recently selected or changed. This component has no scope, but the <objecttype> element restricts the search to objects of type dm\_user.

## Configuration File

webcomponent\config\library\locator\useronlylocator\_component.xml

## Parent Definition

recentuserorgrouplocator:webcomponent/config/library/locator/  
recentuserorgrouplocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from recentuserorgrouplocator:webcomponent/config/library/locator/recentuserorgrouplocator\_component.xml.

---

multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display in the list.

---

## Elements

The configurable elements are inherited from [sysobjectlocator](#), page 1029.

# recentuserorgrouplocator

## Purpose

Extends [objectlocator](#), page 886 and locates users or groups that were recently selected or modified. This component has no scope, but the <objecttype> element restricts the search to objects of type dm\_user or dm\_group.

## Configuration File

webcomponent\config\library\locator\recentuserorgrouplocator\_component.xml

## Parent Definition

objectlocator:webcomponent/config/library/locator/objectlocator\_component.xml

## Scope

All.



## Parameters

multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display in the list.

## Elements

For information on configurable columns, see [doclist](#), page 684.

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...

<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)

---

<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute <code>dqlformatted='false'</code> to quote and escape a value. Use the attribute <code>casesensitive='true'</code> to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## recentwftemplatelocator

**Purpose** Extends [recentsysobjectlocator, page 926](#) and locates workflow templates that were recently used by the user. This component has no scope specified, but the `<objecttype>` element restricts the search to objects of type `dm_process` (templates).

### Configuration File

`webcomponent\config\library\locator\wftemplatelocators_component.xml`

### Parent Definition

`recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator_component.xml`

### Scope

All.

### Parameters

These parameters are inherited from `recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator_component.xml`.

---

contentType	(Optional) Content type for filetype filter
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type to display in the list.

---

## Elements

For information on configurable columns, see [doclist](#), page 684. The configurable elements are inherited from [sysobjectlocator](#), page 1029.

# recentworkqueuecategorylocator

**Purpose** Lists work queue categories that the user has recently selected.

## Configuration File

webcomponent\config\library\queuemanagement\workqueuecategorylocators\_  
component.xml

## Parent Definition

recentsysobjectlocator:webcomponent/config/library/locator/recentsysobjectlocator\_  
component.xml

## Scope

All.

## Parameters

---

arrangeselect	(Optional) Not used.
multiselect	(Optional) Boolean: Set to true to support multiple selection.

---

## Elements

---

<objecttype>	Sets the user preference for the object type to be displayed. The default is dmc_workqueue_category.
--------------	--

---

# recordattributes

**Purpose** Displays the attributes of formal record objects in a form based on a form template. Requires a Records Manager license.

## Configuration File

webcomponent\config\library\records\declarerecord\attribute\_rm\_formalrecord\_component.xml

## Parent Definition

xforms:webcomponent/config/xforms/xforms\_component.xml

## Scope

All.

## Caller

Called by the action.

## Parameters

These parameters are inherited from xforms:webcomponent/config/xforms/xforms\_component.xml.

---

objectId	(Required) Object ID of the form.
readOnly	(Optional) Specifies whether the form should be shown in edit or view mode. In edit mode, WDK buttons are not hidden (form is modal). In view mode, form buttons are disabled, and WDK buttons are displayed (Close and Help).

---

# recordertestcasevariables

**Purpose** For internal use only.

## Configuration File

webcomponent\config\testtool\recordertestcasevariables\_component.xml

## Scope

All.

## Parameters

---

nodeId	(Required) For internal use only.
ticket	(Required) For internal use only.

---

---

## recordingurlgenerator

**Purpose** For internal use only.

### Configuration File

webcomponent\config\testtool\recordingurlgenerator\_component.xml

### Scope

All.

### Parameters

---

target	(Required) For internal use only.
ticket	(Required) For internal use only.

---

## recordlocator (entitlement recordsmanager)

**Purpose** Locator for formal record objects. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\records\locator\rm\_record\_locator\_container.xml

### Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

---

entitlement	recordsmanager
-------------	----------------

---

## Caller

Called by the action and the [recordlocatorcontainer \(entitlement recordsmanager\)](#), page 936 container.

## Parameters

---

objectId	(Required) ID of the source formal record object of the create relationship action. This formal record will be excluded from the list of formal records, since an object cannot have a relationship to itself.
----------	--

---

# recordlocatorcontainer (entitlement recordsmanager)

## Purpose

Container for the recordlocator (entitlement recordsmanager), recordmyobjectlocator (entitlement recordsmanager), and recordsubscriptionlocator (entitlement recordsmanager) components. Requires a Records Manager license.

## Configuration File

webcomponent\config\library\records\locator\rm\_record\_locator\_container.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml



**Scope**

---

entitlement	recordsmanager
-------------	----------------

---

**Caller**

Called by the action and runs the [recordlocator \(entitlement recordsmanager\)](#), [page 935 recordmyobjectlocator \(entitlement recordsmanager\)](#), [page 937](#), and [recordsubscriptionlocator \(entitlement recordsmanager\)](#), [page 938](#) components.

**Parameters**

---

objectId	(Required) Object ID for the formal record that is the parent of the relationship to create.
----------	--

---

## recordmyobjectlocator (entitlement recordsmanager)

**Purpose** Displays all formal record objects owned by a user. Requires a Records Manager license.

**Configuration File**

webcomponent\config\library\records\locator\rm\_record\_locator\_container.xml

**Parent Definition**

myobjectlocator:webcomponent/config/library/locator/myobjectlocator\_component.xml

**Scope**

---

entitlement	recordsmanager
-------------	----------------

---

## Caller

Runs in the [recordlocatorcontainer \(entitlement recordsmanager\), page 936](#) component.

## Parameters

These parameters are inherited from `myobjectlocator:webcomponent/config/library/locator/myobjectlocator_component.xml`.

---

<code>flatlist</code>	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<code>multiselect</code>	(Optional) Boolean: Set to true to support multiple object selection.
<code>objecttype</code>	(Optional) Type of the selected object.

---

# recordsubscriptionlocator (entitlement recordsmanager)

**Purpose** Displays all formal record objects to which a user is subscribed. Requires a Records Manager license.

## Configuration File

`webcomponent\config\library\records\locator\rm_record_locator_container.xml`

## Parent Definition

`subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator_component.xml`

## Scope

---

<code>entitlement</code>	<code>recordsmanager</code>
--------------------------	-----------------------------

---

## Caller

Runs in the [recordlocatorcontainer \(entitlement recordsmanager\), page 936](#) component.

## Parameters

These parameters are inherited from `subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator_component.xml`.

---

<code>flatlist</code>	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<code>multiselect</code>	(Optional) Boolean: Set to true to support multiple object selection.
<code>objecttype</code>	(Optional) Type of the selected object

---

# rejectroutertask

**Purpose**      Rejects router tasks.

## Configuration File

`webcomponent\config\library\workflow\taskmanager\rejectroutertask_component.xml`

## Scope

All.

## Caller

This component is invoked from the task manager component.

**Parameters**

---

objectId	(Required) ID of the router task
----------	----------------------------------

---

## rejectworkflowtask

**Purpose** Displays a UI to allow the user to reject manual workflow tasks.

**Configuration File**

```
webcomponent\config\library\workflow\taskmanager\rejectworkflowtask_
component.xml
```

**Scope**

All.

**Caller**

It is invoked from the [rejectworkflowtask \(type dm\\_task\), page 380](#) action [taskmanager, page 1035](#) component container.

**Parameters**

---

objectId	(Required) ID of the workflow task
----------	------------------------------------

---

## relationships

**Purpose** Extends [objectgrid, page 884](#) and lists the sysobjects to which the source sysobject has a relationship (dm\_relation object).

## Configuration File

webcomponent\config\library\relationships\relationships\_component.xml

## Parent Definition

objectgrid:/webcomponent/config/navigation/objectgrid/objectgrid\_component.xml

## Scope

All.

## Caller

This component is called by the action [relationships](#) (type `dm_sysobject`), [page 381](#) and is displayed in the component [navigationcontainer](#), [page 864](#).

## Parameters

---

objectId	(Required) ID of the sysobject for which to locate relationships
----------	--

---

## Elements

This component has configurable columns (see [doclist](#), [page 684](#)).

# removeattachment

**Purpose** Removes attached objects from a workflow. It is part of the task manager component group.

## Configuration File

webcomponent\config\library\workflow\removeattachment\removeattachment\_component.xml

**Scope**

All.

**Parameters**

---

attachmentName	(Optional) Name of the workflow attachment
objectId	(Required) ID of the attachment.
packageName	(Required) Name of the workflow package.

---

## removecomponent

**Purpose**

Extends [checkout \(type dm\\_sysobject\)](#), [page 628](#) and removes a child node from a virtual document. The work of removing a component is done by the `removecomponentcontainer` component. The container checks out the parent of the selected node, if it is not already checked out, and removes any children of the virtual document. The container extends `checkoutcontainer`.

**Configuration File**

```
webcomponent\config\library\vdm\removecomponent\removecomponent_
component.xml
```

**Parent Definition**

```
checkout:webcomponent/config/library/checkout/checkout_component.xml
```

**Scope**

All.

## Caller

This component is called by the action [removecomponent](#) (type `dm_sysobject`), page 384 and is displayed in the component [removecomponentcontainer](#), page 943.

## Parameters

---

<code>nodeId</code>	(Optional) ID of the child node to be removed.
<code>objectId</code>	(Required) ID of the root node.

---

# removecomponentcontainer

**Purpose**      Contains one or more components that remove a node from a virtual document

## Configuration File

`webcomponent\config\library\vdm\removecomponent\removecomponentcontainer_component.xml`

## Parent Definition

`checkoutcontainer:webcomponent/config/library/checkout/checkoutcontainer_component.xml`

## Scope

All.

## Caller

This component is called by the action [removecomponent](#) (type `dm_sysobject`), page 384 and displays the component [removecomponent](#), page 942.

**Parameters**

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## removecredential

**Purpose** Removes the user credentials (login name, password, repository, domain) for the specifies repository by deleting the cookie. Saved credentials are turned on by default in /wdk/app.xml. Saved credentials are used for HTTP session timeout, DRLs, switching repositories, and accessing the application with a new browser session.

**Configuration File**

webcomponent\config\environment\preferences\credential\removecredential\_component.xml

**Scope**

All.

**Parameters**

---

docbase	(Required) Name of the repository for which the user credentials will be removed.
---------	---

---



## Usage

Username and password are encrypted and saved as browser cookies. This means that credentials saved with one browser, such as IE, will not be available if the user switches to another browser, such as Netscape.

To turn off the save credentials feature, add the following section to your /custom/app.xml file, within the <application> element:

```
<save_credential>
  <enabled>>false</enabled>
  <encryption_key>encryption key here</encryption_key>
  <disabled_docbases>
    <docbase></docbase>
  </disabled_docbases>
</save_credential>
```

To enable saved credentials in certain repositories, you must enable the credentials and then name the repositories that will not support saved credentials. In the following example from a /custom/app.xml file, all repositories except admin support saved credentials:

```
<save_credential>
  <enabled>>true</enabled>
  <encrypted_key>encryption key here</encrypted_key>
  <disabled_docbases>
    <docbase>admin</docbase>
  </disabled_docbases>
</save_credential>
```

**Note:** The encryption key must be identical across all WDK application instances on the application server. If the key is changed by an administrator, users will be prompted for login. You can use the trusted authenticator tool (com.documentum.web.formext.session.TrustedAuthenticatorTool). See *Web Development Kit and Client Applications Development Guide* for details on this tool.

## removestartwfattachment

**Purpose** Removes attached objects. It is part of the startworkflow container component.

## Configuration File

webcomponent\config\library\workflow\removeattachment\removeattachment\_ component.xml

## Scope

All.

## Parameters

---

attachmentName	(Optional) Name of the workflow attachment
objectId	(Required) ID of the attachment.
packageName	(Required) Name of the workflow package.

---

# removestartwfwattachment

**Purpose** This component is used with the startworkflow and taskmanager components. It is used to remove an attachment from a workflow.

## Configuration File

webcomponent\config\library\workflow\removeattachment\removewfattachment\_ component.xml

## Scope

All.

## Caller

This component is called by the action [removestartwfwattachment](#) (type dmi\_package), [page 389](#).

---

## Parameters

---

attachmentName	(Optional) Name of the attachment object being removed from the workflow
docId	(Required) (Required) ID of the attachment object being removed from the workflow
objectId	(Required) (Required) Process (template) ID

---

# removevirtualdocumentnode

**Purpose** This component is used to remove a node from a virtual document.

## Configuration File

webcomponent\config\library\vdm\removecomponent\removevirtualdocumentnode\_component.xml

## Scope

All.

## Caller

This component is called by the action [removevirtualdocumentnode](#) (type `dm_sysobject`), [page 392](#) and displayed in the component [removevirtualdocumentnodecontainer](#), [page 948](#).

## Parameters

---

nodeId	(Required) ID of the node to be removed from the virtual document
vdmRootObjectId	(Required) ID of the root document of the virtual document from which the node is being removed

---

# removevirtualdocumentnodecontainer

**Purpose** This component displays the removevirtualdocumentnode component and allows the user to remove a node from a virtual document.

## Configuration File

webcomponent\config\library\vdm\removecomponent\removevirtualdocumentnodecontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [removevirtualdocumentnode](#) (type `dm_sysobject`), [page 392](#) and displays the component [removevirtualdocumentnode](#), [page 947](#).

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) ID of the component being displayed
componentArgs	(Required) Supplied by the LaunchComponent class.) List of arguments and values to pass to each contained component.(Required) This parameter is supplied by the LaunchComponent class and consists of a list of arguments and values to pass to each contained component.
nodeId	(Required) (Required) ID of the node being removed from the virtual document(Required) ID of the node being removed from the virtual document
vdmRootObjectId	(Required) (Required) ID of the root node of the virtual document from which the node is being removed(Required) ID of the root node of the virtual document from which the node is being removed

---

## Elements

---

<control-propagation>	(Optional) Contains the property elements. Enables propogating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls
-----------------------	--

<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	are propagated to the JSP shown for each object being checked in.  (Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

---

## removewfattachment

**Purpose** This component is used to remove an attachment from a workflow.

### Configuration File

webcomponent\config\library\workflow\removeattachment\removewfattachment\_component.xml

### Scope

All.

### Caller

This component is called by the action [removewfattachment](#) (type `dmi_wf_attachment`), [page 393](#).

### Parameters

---

attachmentName	(Optional) Name of the attachment to be removed from the workflow
objectId	(Required) Workflow task ID

---

packageName	(Optional) ID of the package associated with the workflow from which the attachment is being removed
wf_attachment_id	(Required) ID of the attachment being removed from the workflow

---

## rename

**Purpose** Provides a simple UI for users to rename objects by entering a new name in a single text-edit field on a form. Webtop does not provide a menu to access this component.

### Configuration File

webcomponent\config\library\rename\rename\_component.xml

### Scope

All.

### Caller

This component is called by the action [rename \(type dm\\_sysobject\), page 395](#) and displayed by the component [combocontainer, page 644](#).

### Parameters

---

objectId	(Required) ID of the object to rename
----------	---------------------------------------

---

# renamergrouplist

**Purpose** Extends [renamergobrequestlist](#), page 952 and displays job requests for rename user or group.

## Configuration File

webcomponent\config\admin\renamelog\renamergrouplist\_component.xml

## Parent Definition

renamergobrequestlist:webcomponent/config/admin/renamelog/renamergobrequestlist\_component.xml

## Scope

All.

## Caller

This component is called by the action [grouprenamelog](#) (type `dm_group`), page 311 in the menubar component UI.

## Parameters

---

renamergobtarget	(Optional) Type of objects requested for rename. Valid values: USER   GROUP
------------------	---

---

# renamergobrequestlist

**Purpose** Displays jobs to that rename users or groups



---

## Configuration File

webcomponent\config\admin\renamelog\renamejobrequestlist\_component.xml

## Scope

All.

## Caller

This component is called by the action [delete \(type job\\_request\)](#), page 275.

## Parameters

---

renametarget	(Optional) Type of objects to be renamed. Valid values: user   group. Default = user
--------------	---

---

# renamereport

**Purpose** Displays job requests for rename user or group.

## Configuration File

webcomponent\config\admin\renamelog\renamereport\_component.xml

## Scope

All.

## Caller

This component is called by the action [viewreport \(type dm\\_rename\\_log\)](#), page 475.

## Parameters

---

objectId	(Required) ID of the selected report
oldname	(Optional) Name of the report before rename

---

# renameuserlist

**Purpose** Extends [renamejobrequestlist](#), page 952 and displays job requests for renaming users.

## Configuration File

webcomponent\config\admin\renamelog\renameuserlist\_component.xml

## Parent Definition

renamejobrequestlist:webcomponent/config/admin/renamelog/renamejobrequestlist\_component.xml

## Scope

All.

## Caller

This component is called by the action [userrenamelog](#) (type `dm_user`), page 452 in the menubar component UI.

## Parameters

---

renametarget	(Optional) Type of objects requested for rename. Valid values: USER   GROUP
--------------	---

---

## renditions (type dm\_sysobject)

**Purpose** Extends [objectgrid](#), [page 884](#) and lists the renditions of an object. This component is defined with dm\_sysobject scope. For dm\_folder objects it is specifically not defined, so that the renditions component is not available for folders.

### Configuration File

webcomponent\config\library\renditions\renditions\_component.xml

### Parent Definition

objectgrid:/webcomponent/config/navigation/objectgrid/objectgrid\_component.xml

### Scope

type	dm_sysobject
------	--------------

### Caller

This component is called by the action [renditions \(type dm\\_sysobject\)](#), [page 397](#) and displayed in the component [navigationcontainer](#), [page 864](#).

### Parameters

contentType	(Optional) Type of content to be rendered.
objectId	(Required) ID of the object to be rendered.

### Elements

This component has configurable columns (see [doclist](#), [page 684](#)).

# reordercomponents

**Purpose** Extends [checkout \(type dm\\_sysobject\), page 628](#) and changes the order of children in a virtual document. The actual work of reordering components is done in the [reordercomponentscontainer](#) component. The container extends [checkoutcontainer](#).

## Configuration File

webcomponent\config\library\vdm\reordercomponents\reordercomponents\_component.xml

## Parent Definition

checkout:webcomponent/config/library/checkout/checkout\_component.xml

## Scope

All.

## Caller

This component is called by the action [reordercomponents \(type dm\\_sysobject\), page 398](#) and displayed in the component [reordercomponentscontainer, page 957](#).

## Parameters

---

nodeId	(Optional) ID of root node
objectId	(Required) Root virtual document ID.

---

# reordercomponentscontainer

**Purpose**        Contains one or more components that reorder nodes in a virtual document

## Configuration File

webcomponent\config\library\vdm\reordercomponents\reordercomponentscontainer\_  
component.xml

## Parent Definition

checkoutcontainer:webcomponent/config/library/checkout/checkoutcontainer\_  
component.xml

## Scope

All.

## Caller

This component is called by the action [reordercomponents](#) (type `dm_sysobject`), page 398 and displays the component [reordercomponents](#), page 956.

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## reordervirtualdocumentnodes

**Purpose** This component is used to change the order of nodes in a virtual document.

### Configuration File

webcomponent\config\library\vdm\reordercomponents\reordervirtualdocumentnodes\_  
component.xml

### Scope

All.

### Parameters

---

nodeId	(Required) ID of node being reordered
vdmRootObjectId	(Required) ID of root node

---

## repeatworkflowtask

**Purpose** Completes workflow tasks and assign repeat performers for them. You can use it only for repeatable tasks.

### Configuration File

webcomponent\config\library\workflow\taskmanager\repeatworkflowtask\_  
component.xml

### Scope

All.

**Caller**

This component is invoked from the task manager.

**Parameters**

---

objectId	(Required) ID of the workflow task to be repeated.
----------	--

---

## reportdetailsauditdrilldown

**Purpose**

This component displays the audit events of a workflow instance. The user will see no events if the workflow is not audited.

**Configuration File**

webcomponent\config\library\workflow\reportdetails\reportdetailsauditdrilldown\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [reportdetailsauditdrilldown](#) (type `dm_workflow`), [page 401](#).

## Parameters

---

eventFilter	(Optional) Specifies the filter to use: 0 = show workflow events   1 = show workflow and doc events
hideEvents	(Optional) String of event names that should not be displayed. Event names are listed in the
objectId	(Required) ID of the selected workflow.

---

## Elements

For information on configurable columns, see [doclist](#), page 684.

# reportdetailsauditlist

**Purpose** This component displays the audit events of a workflow instance. The user will see no events if the workflow is not audited.

## Configuration File

```
webcomponent\config\library\workflow\reportdetails\reportdetailsauditlist_
component.xml
```

## Scope

All.

## Caller

This component is called by the action [reportdetailsauditlist](#) (type `dm_workflow`), page 402.



## Parameters

eventFilter	(Optional) Specifies the filter to use: 0 = show workflow events   1 = show workflow and doc events
hideEvents	(Optional) String of event names that should not be displayed. Event names are listed in the Content Server API Reference Manual.
objectId	(Required) ID of the selected workflow.

## Elements

For information on configurable columns, see [doclist](#), page 684.

# reportdetailscontainerdrilldown (type dm\_workflow)

**Purpose** Extends [propertysheetcontainer](#), page 914 and contains report detail components on a workflow.

## Configuration File

webcomponent\config\library\workflow\reportdetailscontainer\reportdetailscontainerdrilldown\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

type	dm_workflow
------	-------------

## Parameters

---

component	(Optional) The contained component.
objectId	(Required) ID of the selected workflow.

---

# reportdetailscontainerlist (type dm\_workflow)

**Purpose** Extends [propertysheetcontainer](#), page 914 and contains report detail components on a workflow.

## Configuration File

webcomponent\config\library\workflow\reportdetailscontainer\reportdetailscontainerlist\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

---

type	dm_workflow
------	-------------

---

## Parameters

---

component	(Optional) The contained component.
objectId	(Required) ID of the selected workflow.

---

## reportdetailsheader

**Purpose** This component is included in the UI of [reportdetailscontainerlist](#) (type `dm_workflow`), page 962 and [reportdetailscontainerdrilldown](#) (type `dm_workflow`), page 961. The header component displays a different page depending on the active component in the container.

### Configuration File

```
webcomponent\config\library\workflow\reportdetails\reportdetailsheader_
component.xml
```

### Scope

All.

### Parameters

component	(Optional) Specifies the component for which to display the header.
objectId	(Required) ID of the selected workflow.

## reportdetailsmap

**Purpose** This component displays the map view of the workflow instance in the Web Workflow Manager.

### Configuration File

```
webcomponent\config\library\workflow\reportdetails\reportdetailsmap_
component.xml
```

## Parent Definition

webwfm\_checkjvm:webcomponent/config/library/workflow/webwfm/webwfm\_component.xml

## Scope

All.

## Caller

This component is called by the action [reportdetailsmap](#) (type `dm_workflow`), page 402.

## Parameters

These parameters are inherited from webwfm:webcomponent/config/library/workflow/webwfm/webwfm\_component.xml.

---

objectId	(Required) ID of the path to the object for which to start a workflow
----------	---

---

# reportdetailssummarydrilldown

## Purpose

Displays the summary of a selected workflow's tasks and their attributes, such as task name, runtime status, action, performer, comments, and receive/complete/overdue days.

## Configuration File

webcomponent\config\library\workflow\reportdetails\reportdetailssummarydrilldown\_component.xml

## Scope

All.

## Parameters

objectId	(Required) ID of the selected workflow.
showAttachment	(Optional) Set to true to display attached documents with tasks
taskFilter	(Optional) Sets the selected filter. Valid values: 0 = all   1 = current   2 = overdue

## Elements

Configurable columns are described in [doclist, page 684](#) with the following additional element:

<taskmgractionname>	Specifies the name of the taskmanager component
---------------------	---

# reportdetailsummarylist

**Purpose** Displays the summary of a selected workflow's tasks and their attributes, such as task name, runtime status, action, performer, comments, and receive/complete/overdue days.

## Configuration File

webcomponent\config\library\workflow\reportdetails\reportdetailsummarylist\_component.xml

## Scope

All.

## Caller

This component is called by the action [reportdetailsummarylist \(type dm\\_workflow\)](#), [page 404](#). The component it displays is generated dynamically.

## Parameters

---

objectId	(Required) ID of the selected workflow.
showAttachment	(Optional) Set to true to display attached documents with tasks
taskFilter	(Optional) Sets the selected filter. Valid values: 0 = all   1 = current   2 = overdue

---

## Elements

Configurable columns are described in [doclist](#), page 684.

# reportdql

**Purpose** This component displays a workflow report based on a DQL query.

## Configuration File

webcomponent\config\library\workflow\historicalreport\reportdql\_component.xml

## Scope

All.

## Parameters

---

query	(Optional) When the querytype is string, this value is the search string. If the querytype is not string, this value is the query string.
-------	---

---

---

# reportmaindrilldown

**Purpose** Displays a list of workflow instances and its attributes

## Configuration File

webcomponent\config\library\workflow\reportmain\reportmaindrilldown\_  
component.xml

## Scope

All.

## Caller

This component is called by the component class [inboxdrilldown](#), page 815.

## Parameters

---

overdueDays	(Optional) Sets minimum number of days overdue (default = 0)
workflowFilter	(Optional) Sets the selected filter. Valid values: 0 = all   1 = completed   2 = running
workflowType	(Optional) Type of workflow query. Valid values: 0 = general   1 = supervisor   2 = template ID   3 = workflow name   4 = Document ID   5 = Workflow ID   6 = all   7 (default) = supervisor and performer
workflowTypeParam	(Optional) Second workflow type for query (see workflowType for valid values)

---

## Elements

Configurable columns are described in [doclist, page 684](#). The following additional elements are configurable:

---

<savedetailsreport>	Set to true to save the workflow details report in CSV format. Default = false
---------------------	--

---

# reportmainlist

**Purpose** Displays a list of workflow instances and its attributes. This component is extended by the Webtop reportmainclassic component.

## Configuration File

webcomponent\config\library\workflow\reportmain\reportmainlist\_component.xml

## Scope

All.

## Caller

This component is called by the Webtop action reportmainclassic.

## Parameters

---

overdueDays	(Optional) Sets minimum number of days overdue (default = 0)
workflowFilter	(Optional) Sets the selected filter. Valid values: 0 = all   1 = completed   2 = running



---

workflowType	(Optional) Type of workflow query. Valid values: 0 = general   1 = supervisor   2 = template ID   3 = workflow name   4 = Document ID   5 = Workflow ID   6 = all   7 (default) = supervisor and performer
workflowTypeParam	(Optional) Second workflow type for query (see workflowType for valid values)

---

## Elements

Configurable columns are described in [doclist, page 684](#). The following additional elements are configurable:

---

<savedetailsreport>	Set to true to save the workflow details report in CSV format. Default = false
---------------------	--

---

# reportmainsettings

**Purpose** This component allows the user to set criteria for the workflow instances to be displayed in the reportmain component.

## Configuration File

webcomponent\config\library\workflow\reportmain\reportmainsettings\_component.xml

## Scope

All.

## Caller

This component is launched by the action [reportmainsettings, page 405](#) and displayed in the component [combocontainer, page 644](#).

## Parameters

---

documentId	(Optional) Specifies a document for which workflows will be reported
filter	(Required) Sets the selected filter. Valid values: 0 = all   1 = completed   2 = running
overdueDays	(Optional) Specifies the minimum number of days overdue for which to display a report
overdueOnly	(Required) True to filter for overdue task status.
templateId	(Optional) Specifies a template for which workflows will be reported
userName	(Required) Name of the user for which to display workflows.

---

# reportsysobjectlocator

**Purpose** Extends [sysobjectlocator](#), page 1029 and locates reports. This locator is nested to from the component class [reportmainsettings](#), page 969. The container extends [locatorcontainer](#), page 839.

## Configuration File

webcomponent\config\library\workflow\reportsysobjectlocator\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from `sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator_component.xml`.

<code>contentType</code>	(Optional) Content type of the selected object.
<code>flatlist</code>	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<code>folderId</code>	(Optional) Object ID of the folder where the search should commence.
<code>multiselect</code>	(Optional) Boolean: Set to true to support multiple object selection.
<code>objecttype</code>	(Optional) Type of the selected object

## Elements

The configurable elements are inherited from [sysobjectlocator, page 1029](#). Additional element:

<code>&lt;setrepositoryfromobjectid&gt;</code>	Set to true to execute the query against the source repository. Affects reference and foreign objects.
<code>&lt;allversionsvisible&gt;</code>	Set to true to display all versions

# reportsysobjectlocatorcontainer

**Purpose** This component displays a simple list of sysobjects including the folder and the findtype.

## Configuration File

`webcomponent\config\library\workflow\reportsysobjectlocatorcontainer_component.xml`

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

# repositionvirtualdocumentnode (type dm\_sysobject)

**Purpose** Contains one or more components that move a virtual document node to a different part of the virtual document tree

## Configuration File

webcomponent\config\library\vdm\repositionnode\repositionvirtualdocumentnode\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [repositioncomponent](#) (type `dm_sysobject`), page 406.

## Parameters

---

insertAfterNodeId	(Optional) Node ID of the node after which the object should be added
nodeId	(Required) Node ID of the object to operate on, when it is part of a virtual document.
targetParentNodeId	(Required) Node ID of the parent node to which the object will be added
vdmRootObjectId	(Required) Object ID of the root of a virtual document (when operating on a child node).

---

# repositoryselector

## Purpose

This component allows the user to select a repository to be added to a list of preferred repositories. Like the login component, this component and its container are listed in `Environment.properties` as `non_docbase_components`. Displays visible repositories for the list of connection brokers. The user can manually enter the name of a connection broker to select a repository that connects to that broker.

## Configuration File

`wdk\config\repositoryselector_component.xml`

## Scope

All.

## Caller

This component is called by the action [selectrepository](#), page 416 from the login dialog.

# repositoryselectorcontainer

## Purpose

A container that enables selection of a content repository.

## Configuration File

wdk\config\repositoryselectorcontainer\_component.xml

## Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from container:wdk/config/container\_component.xml.

---

component	(Required) The contained component. Set the requiresVisit attribute to true to require the component to be visited before an OK button is displayed.
-----------	--

---

## rerunfailedautoworkflowtask

**Purpose** Retries failed automatic workflow tasks. You can use it only for failed automatic tasks. It is invoked from the task manager.

### Configuration File

webcomponent\config\library\workflow\taskmanager\rerunautotask\_component.xml

### Scope

All.

### Parameters

objectId	(Required) ID of failed autoworkflow task.
----------	--

## resumetask (type dm\_task)

**Purpose** Resumes a task in a workflow that was previously halted. This component is defined for dm\_task objects.

### Configuration File

webcomponent\config\library\workflow\resume\resumetask\_component.xml

### Scope

type	dm_task
------	---------

## Caller

This component is launched by the action [resumetask \(type dm\\_task\), page 408](#) and displayed in the component [combocontainer, page 644](#).

## Parameters

---

objectId	(Required) ID of the work item.
objectName	(Required) Task name.

---

# resumeworkflow (type dm\_workflow)

**Purpose** Resumes halted workflow objects.

## Configuration File

webcomponent\config\library\workflow\resume\resumeworkflow\_component.xml

## Scope

---

type	dm_workflow
------	-------------

---

## Caller

This component is called by the action [resumeworkflow \(type dm\\_workflow\), page 409](#) and is displayed in the component [combocontainer, page 644](#).

## Parameters

---

objectId	(Required) ID of the workflow.
----------	--------------------------------

---



## richtextimagepicker

**Purpose** Uploads an image into the Rich Text Editor, resulting in the insertion of an image tag `<img>` in the rich text HTML. The image file is stored in a temporary location on the application server until the rich text is saved, at which point the image file is saved to the repository. Requires Collaborative Edition on the Content Server.

### Configuration File

`webcomponent\config\library\richtext\imagepicker_component.xml`

### Scope

All.

## richtextinsertlink

**Purpose** Inserts a link in the rich text editor. Requires Collaborative Edition on the Content Server.

### Configuration File

`webcomponent\config\library\richtext\insertlink_component.xml`

### Scope

All.

# roleattributes

**Purpose** Extends the component [groupattributes, page 746](#) and displays a list of attributes of a role (dm\_group object with attributes group\_class="role" and group\_name="role\_name").

## Configuration File

webcomponent\config\admin\role\roleattributes\_component.xml

## Parent Definition

groupattributes:webcomponent/config/admin/group/groupattributes\_component.xml

## Scope

All.

## Caller

This component is called within the [roleproperties, page 979](#) container by the [newrole \(privilege creategroup, type dm\\_role\), page 347](#) action.

## Parameters

These parameters are inherited from groupattributes:webcomponent/config/admin/group/groupattributes\_component.xml.

---

objectId	(Required) Object ID of the role whose attributes to display.
objectname	(Required) Object name of the role whose attributes to display.

---

## rolelist

**Purpose** Extends the component [grouplist](#), page 747 and displays a list of objects of type `dm_group` with attributes `group_class="role"` and `group_name="role_name"`. This component is a node in the [usermanagement](#), page 1074 component.

### Configuration File

`webcomponent\config\admin\role\rolelist_component.xml`

### Parent Definition

`grouplist:webcomponent/config/admin/group/grouplist_component.xml`

### Scope

All.

### Parameters

None.

## roleproperties

**Purpose** Extends the container [adminpropertycontainer](#), page 515 and displays a list of object attributes of type `dm_group` with attributes `group_class="role"` and `group_name="role_name"`. Contains the `roleattributes` component.

### Configuration File

`webcomponent\config\admin\role\roleproperties_component.xml`

## Parent Definition

adminpropertycontainer:webcomponent/config/admin/container/  
adminpropertycontainer\_component.xml

## Scope

All.

## Caller

This component is launched by the [newrole](#) (privilege creategroup, type dm\_role), page [347](#) action.

## Parameters

---

objectId	(Required) ID of the object for which properties are displayed. Set to newobject to create a new role.
objectname	(Required) Name of the object.

---

## Elements

---

<objecttype>	Type of object that is supported by the container (placeholder).
--------------	--

---

# roomgrouplocator

**Purpose** Displays a selection list with all groups for a room.

## Configuration File

webcomponent\config\library\rooms\roomuserorgrouplocatorcontainer\_component.xml

## Parent Definition

grouplocator:webcomponent/config/library/locator/grouplocator\_component.xml

## Scope

All.

## Parameters

arrangeselect	(Optional) Boolean: Set to true to allow selection of members in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
roomId	(Required) Object ID of the room whose members to include in the list.

# roomgrouplocatorcontainer

**Purpose** Contains one or more roomgrouplocator components.

## Configuration File

webcomponent\config\library\rooms\roomuserorgrouplocatorcontainer\_component.xml

## Parent Definition

grouplocatorcontainer:webcomponent/config/library/locator/grouplocator\_component.xml

## Scope

All.

## Parameters

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of members in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
nobuiltingroups	(Optional) Boolean: Set to true to suppress display of built-in groups (Owners, Contributors, Members, Visitors)
objecttype	(Optional) Type of the selected object
roomId	(Required) Object ID of the room whose members to include in the list.
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## Elements

---

<multidocbasesupport>	Set to true to enable locator for multiple repositories
-----------------------	---

---

## roommembership

**Purpose** Displays a table listing all members of a room. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\rooms\room\_membership.xml

### Scope

All.

### Parameters

---

objectId	(Required) Object ID of the room whose members are listed.
----------	--

---

## roomoptions

**Purpose** Sets the options for a room, including default permissions for contributors. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\rooms\newroom\_component.xml

### Scope

All.

## Parameters

---

objectId	(Optional) ID of the room for which to set options
----------	--

---

# roomuserorgrouplocator

**Purpose** Displays a selection list with all members of a room. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\rooms\roomuserorgrouplocatorcontainer\_component.xml

## Parent Definition

userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml

## Scope

All.

## Parameters

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of members in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
groupName	(Optional) ID of group to which the selection will be restricted



multiselect	(Optional) Boolean: Set to true to support multiple selection.
nobuildinggroups	(Optional) Boolean: Set to true to suppress display of built-in groups (Owners, Contributors, Members, Visitors).
objecttype	(Optional) Object type to display.
roomId	(Required) Object ID of the room whose members to include in the list.
toplevelusers	(Optional) Boolean: Set to true to display only those users added directly to room.

---

## roomuserorgrouplocatorcontainer

**Purpose** Container component for displaying a list of room members. Requires Collaborative Edition.

### Configuration File

webcomponent\config\library\rooms\roomuserorgrouplocatorcontainer\_component.xml

### Parent Definition

userorgrouplocatorcontainer:webcomponent/config/library/locator/userorgrouplocator\_component.xml

### Scope

All.

## Parameters

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of members in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
groupName	(Optional) ID of a group to which the selection will be limited
multiselect	(Optional) Boolean: Set to true to support multiple selection.
nobuildinggroups	(Optional) Boolean: Set to true to display only custom groups.
objecttype	(Optional) Object type to display.
roomId	(Required) Object ID of the room whose members to include in the list.
toplevelusers	(Optional) Boolean: Set to true to display only top-level users.

---

## saveasxforms

**Purpose**        Creates a new form from a current form. The user must have the role `form_user`.

### Configuration File

`webcomponent\config\xforms\saveasxforms_component.xml`

### Scope

All.

## Caller

This component is called by the action [saveasxforms](#) (type `dm_sysobject`), page 412 and displayed in the component [newxformscontainer](#), page 881.

## Parameters

---

<code>objectId</code>	(Required) Object ID of the current folder in which the form will be created.
-----------------------	---

---

# savechanges

**Purpose** Provides the UI and functionality for saving changes made to virtual document structures.

## Configuration File

`webcomponent\config\library\vdm\savechanges\savechanges_component.xml`

## Scope

All.

## Caller

This component is called by the action [savechanges](#), page 413 and displayed in the component [dialogcontainer](#), page 669.

## Parameters

---

<code>closeUponCompletion</code>	(Optional) Set to true to close the virtual document after changes are saved.
<code>objectId</code>	(Optional) ID of the virtual document

---

## savecredential

**Purpose** Saves the user credentials (login name, password, repository, domain) as a cookie. This feature is turned on by default in /wdk/app.xml.

### Configuration File

webcomponent\config\environment\preferences\credential\savecredential\_  
component.xml

### Scope

All.

## savedsearches (version 5.2.5)

**Purpose** This is the WDK 5.2.5 savedsearches component that supports operations on saved searches. The user can execute the saved search by viewing it from a folder or object list or by selecting the savedsearches component within advanced search. This component cannot be called by URL, but it can be extended by a custom component.

### Configuration File

webcomponent\config\library\search\savedsearches\_component.xml

### Scope

---

version	5.2.5
---------	-------

---

### Caller

Cannot be called directly. Must be extended.

## Elements

<queryfilterset>	Defines the items (<queryfilter>) that are displayed in a drop-down list.
<queryfilter>	Sets up a default filter, such as My searches and All searches
<displayname>	Name of the filter to be displayed in a dropdown list. Can be a string or <nlsid>.
<dqlfilter>	DQL that searches for dm_query objects. The where clause is specified as the value of the DQL filter element.

## savereportlocator

**Purpose** Extends [folderlocatorcontainer](#), page 734 and allows a user to locate a folder in which to save a report. This locator is nested within the component class of several reporting components.

### Configuration File

```
webcomponent\config\library\workflow\savereport\savereportlocator_component.xml
```

### Parent Definition

```
folderlocatorcontainer:webcomponent/config/library/locator/folderlocators_component.xml
```

### Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display all objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## savereportname

**Purpose** This component is included in the component [savereportlocator, page 989](#) and allows the user to specify a report name for a saved report.

### Configuration File

```
webcomponent\config\library\workflow\savereport\savereportlocator_component.xml
```

### Scope

All.

# savesearch

**Purpose** Creates a `dm_smart_list` object. The user is presented with a dropdown list to select the content repository where the search will be stored. The repository is saved as a user preference. If the authentication for the selected repository fails or the user cancels authentication, the current repository is selected. The search is saved within a folder named Saved Searches in the user's home cabinet. (The folder is created if it does not exist.)

## Configuration File

`webcomponent\config\library\saveSearch\saveSearchex\saveSearch_component.xml`

## Scope

All.

## Caller

This component is called by the action [saveSearch](#), page 414 and displayed in the component [saveSearchcontainer](#), page 992.

## Parameters

---

<code>queryId</code>	(Required) The object id of the current search, an internal parameter to enable sharing of the query context between search components. If the parameter does not exist, the 5.2.5 save search component is called.
----------------------	---

---

## savesearch (version 5.2.5)

**Purpose** This is the 5.2.5 savesearch component, present for backward compatibility. Saves a performed search as a dm\_query object. The search is saved within a folder named Saved Searches in the user's home cabinet. (The folder is created if it does not exist.)

### Configuration File

webcomponent\config\library\savesearch\savesearch\_component.xml

### Scope

---

version	5.2.5
---------	-------

---

### Caller

Cannot be called directly. Must be extended.

### Usage

If your custom WDK 5.2.x component extends this component, it will use this component definition. If your application simply calls savesearch, you will get the 5.3 component.

## savesearchcontainer

**Purpose** This container extends dialogcontainer and contains the 5.3 savesearch component. It allows authentication in the container initialization.

### Configuration File

webcomponent\config\library\savesearch\savesearchex\savesearchcontainer\_component.xml



---

## Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [savesearch, page 414](#) and displays the component [savesearch, page 991](#).

## Parameters

These parameters are inherited from container:wdk/config/container\_component.xml.

---

component	(Required) ID of the component being displayed
-----------	--

---

# search

## Purpose

The WDK 5.3 search component performs a search query on the current repository for the supplied string of keywords or other type of query input. This component displays the search results. Searches in indexed attributes as well as full-text.

## Configuration File

webcomponent\config\library\search\searchex\search\_component.xml

## Scope

All.

## Caller

This component is called by the actions [search](#), page 414, [view \(type dm\\_query\)](#), page 461, and [view \(type dm\\_smart\\_list\)](#), page 463.

## Parameters

---

drilldown	(Optional) Set to true to use the drilldown view by default to display the search results. This parameter is set to false in Webtop when the user selects classic view in preferences.
query	(Required) DQL query (for queryType = dql) or search string (for queryType = string)
queryType	(Optional) Specifies the type of query. Valid values: string (one or more full-text search keywords)   objectId (of a saved search)   querydef (string consisting of smartlist content)   queryId (internal ID for the current query to be revised)   null (DQL)
showwait	(Optional) Not used

---

## Elements

---

<failoverenabled>	The query, but not results, is persisted in the case of failover, and the query is reissued.
<defaultsources>	Sets the default search locations to be displayed for a search. Valid values: current_docbase (current repository)   visible_docbases (all visible repositories)   preferred_sources (all visible repositories and external sources set in the user's preferences)   all_sources (all visible repositories and external sources)

---

<displayresultspath>	Set to true to query and display the folder path to each result object. A value of false will speed performance.
<type>	Default type for the search
<columns>	Configures the default columns to be displayed in search results for both search and advsearch components. Only searchable attributes can be displayed. The user can select other columns in the display_preferences UI. For information on configuring columns, see "Configuring Columns in <i>Web Development Kit and Client Applications Development Guide</i> .
<highlight_matching_terms>	Set to true to highlight matches for search terms in the results
<dragdrop>	Supports the drag and drop feature.
<sourceactions>	Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.
<targetactions>	Contains zero or more <targetaction> classes that support drag and drop actions on targets in the component.
<targetaction>	Contains a fully qualified class name of the class that implements IDragTargetAction, for example, com.documentum.web.formext.control.dragdrop.CopyToFolderTargetAction.
<dataproviders>	Contains one or more <dataprovider> elements.
<dataprovider>	Contains a <format> element and a <provider> element.
<dataprovider>.<format>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<dataprovider>.<provider>	Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

## Usage

Results columns are configured in this component, even results for advanced search. You can scope this component to display results columns for various object types. Add a column element for each custom attribute to the scoped definition.

A faster query that overrides the full-text string query uses the DQL query type. If you supply a null value for `queryType` and supply a DQL string for the `query` attribute, the query is passed to the 5.2.5 search component. This component is limited to single-repository queries, and the user cannot configure search results columns. The results are limited to the display of `dm_sysobject` attributes, which are configurable in the 5.2.x search component definition.

# search (version 5.2.5)

**Purpose** This is the WDK 5.2.5 search component. It cannot be addressed directly by URL or ID, because it is versioned. It can be extended, for backward compatibility. Searches in object name, subject, and title attributes as well as full-text if configured for full-text search.

## Configuration File

`webcomponent\config\library\search\search_component.xml`

## Scope

---

version	5.2.5
---------	-------

---

## Caller

This component is called only when it is extended by a custom component by the actions [search](#), page 414 and [view](#) (type `dm_query`), page 461, and displayed in the component [searchcontainer](#), page 998.

## Parameters

---

query	(Required) DQL query (for queryType = dql) or search string (for queryType = string)
queryType	(Optional) Specifies whether the query is a string value or DQL. Valid values: string   dql
ranking	(Optional) Boolean flag. True to display the ranking of search results.
useColumnsFromQuery	(Optional) Set to true to display results of the columns specified in the query rather than the columns configured in the component definition.

---

## Elements

---

<showfolderpath>	True to display the folder path to each result
<columnsdrilldown>	Specifies the order and visibility of attribute columns in the search results
<column>.	Corresponds to a column in the search results. Contains <attribute>, <label>, and <visible> elements.
<attribute>	Corresponds to an attribute of the search result. Must be a sysobject attribute.
<label>	Label for the column in the search results. Contains string or <nlsid>.
<visible>	Boolean flag that sets the visibility of the column in the search results
<searchtypes>	Contains the types of objects (<searchtype> that are searched for. The first type is the default.
<searchtype>. <displayname>	Name that will be displayed for the object type. Contains string or <nlsid>.
<searchtype>. <docbasetype>	Type that is searched for. Value must be a valid object type in the repository.

<code>&lt;searchtype&gt;.</code> <code>&lt;fulltextsearch&gt;</code>	True to perform full-text search on the type, false to search on properties only. By default, full-text search is performed on <code>dm_document</code> and subtypes only. Set the <code>showranking</code> attribute to true to show the Verity ranking of results.
<code>&lt;searchtype&gt;.&lt;column&gt;</code>	Set to false to hide a column for the attribute that is specified. For example, a value of false and <code>&lt;column attribute = 'r_content_size'&gt;</code> hides the size column for the Documentum type.

---

## searchcontainer

**Purpose** This is the WDK 5.2.5 search container component. It cannot be addressed by URL, but it can be extended by your custom component for backward compatibility.

### Configuration File

`webcomponent\config\library\search\searchcontainer_component.xml`

### Parent Definition

`nestedcomponentcontainer:wdk/config/nestedcomponentcontainer_component.xml`

### Scope

All.

### Caller

This component is called by the actions [search, page 414](#) and [view \(type dm\\_query\), page 461](#), and displays the component [search, page 993](#).

## Parameters

These parameters are inherited from container:wdk/config/container\_component.xml.

component	(Required) Specifies the initial component to have focus
-----------	--

## searchsources\_preferences

**Purpose** Enables the user to select repositories for search. Users can select external sources if ECI Services is installed.

### Configuration File

webcomponent\config\environment\preferences\searchsources\searchsources\_preferences\_component.xml

### Scope

All.

## searchstatus

**Purpose** Lists each repository and external source that is used for the current search and gives the status of each source.

### Configuration File

webcomponent\config\library\search\searchex\searchstatus\_component.xml

### Scope

All.

## Parameters

---

queryId	(Required) The object id of the current search, an internal parameter to enable sharing of the query context between search components.
---------	---

---

# searchstatuscontainer

**Purpose**      Contains one or more search status components.

## Configuration File

webcomponent\config\library\search\searchex\searchstatuscontainer\_component.xml

## Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from container:wdk/config/container\_component.xml.

---

component	(Required) Specifies the component that will have initial focus
-----------	---

---



---

## select\_application

**Purpose** Enables the user to select an application from the file system to use as the standard application for a given rendition or format type. Clicking OK returns the full path of the application to the calling component as a return parameter.

### Configuration File

webcomponent\config\library\selectapplication\select\_application\_component.xml

### Scope

All.

### Parameters

---

path	(Optional) File system path to the folder displayed as the default starting location.
------	---

---

## selectsingleattributevalue

**Purpose** Provides UI and behavior for selecting a single (not repeating) attribute value.

### Configuration File

webcomponent\config\library\selectsingleattributevalue\selectsingleattributevalue\_component.xml

### Parent Definition

docbasesingleattribute:wdk/config/docbasesingleattribute\_component.xml

## Scope

All.

## Caller

This component should be extended for editing specific single-valued attributes.

## Parameters

---

control	(Required) ID of the doctypeattributevalue control.
locator	(Optional) Specifies a selector component to select a single attribute value
selectedAttribute	(Optional) Attribute to be passed to the locator

---

## Elements

---

<locator>	Locator component to be launched for selecting the attribute value. For example, the selectuserattributevalue component sets the value to useronlylocatorcontainer.
<selectedAttribute>	Selected attribute value. Set this value in the derived component. For example, the selectuserattributevalue component sets the value to object_name.

---

## Usage

Extend this component to provide a locator for an open-ended attribute value of a single-value attribute. Register your component as the value of the <editcomponent> element for the desired attribute in a doctypeobjectconfiguration file. For more information on doctypeobject configuration, refer to [doctypeobject](#), page 88.

# selectuserattributevalue

**Purpose** Provides UI and behavior for selecting a user to supply an attribute value.

## Configuration File

webcomponent\config\library\selectsingleattributevalue\selectuserattributevalue\_component.xml

## Parent Definition

selectsingleattributevalue:webcomponent/config/library/selectsingleattributevalue/selectsingleattributevalue\_component.xml

## Scope

All.

## Caller

This component is called by a docbaseobject control to edit the owner\_name attribute, as configured in docbaseobjectconfiguration\_dm\_sysobject.xml.

## Parameters

These parameters are inherited from selectsingleattributevalue:webcomponent/config/library/selectsingleattributevalue/selectsingleattributevalue\_component.xml.

control	(Required) ID of the docbaseattributevalue control.
locator	(Optional) Specifies a selector component to select a single attribute value
selectedAttribute	(Optional) Attribute to be passed to the locator

## Elements

---

<locator>	Locator component to be launched for selecting the attribute value. For example, the selectuserattributevalue component sets the value to useronlylocatorcontainer.
<selectedAttribute>	Selected attribute value. Set this value in the derived component. For example, the selectuserattributevalue component sets the value to object_name.

---

# sendlocator

**Purpose** Sends an email link (DRL, see [drl](#), page 698) to one or more selected documents, folders, tasks, notifications, or quickflow items using the HTML mailto command.

## Configuration File

webcomponent\config\library\sendlocator\sendlocator\_component.xml

## Scope

All.

## Caller

This component is launched within the sendlocator container by the [sendlocator](#) (type [dm\\_sysobject](#)), page 419 action, which is in turn called by the menubar component UI. The sendlocator component has no UI.

## Parameters

---

objectId	(Required) ID of the folder or object for which to send a locator.
----------	--

---

---

## sendlocator (type dm\_notification)

**Purpose** Sends an email link (DRL, see [drl](#), page 698) to one or more selected documents, folders, tasks, notifications, or quickflow items using the HTML mailto command.

### Configuration File

webcomponent\config\library\sendlocator\sendlocator\_workflow\_items\_component.xml

### Parent Definition

sendlocator:webcomponent/config/library/sendlocator/sendlocator\_component.xml

### Scope

---

type	dm_notification
------	-----------------

---

### Caller

This component is launched within the sendlocator container by the action [sendlocator \(type dm\\_notification\)](#), page 417, which is in turn called by the menubar component UI. The sendlocator component has no UI.

### Parameters

These parameters are inherited from sendlocator:webcomponent/config/library/sendlocator/sendlocator\_component.xml.

---

objectId	(Required) ID of the folder or object for which to send a locator.
----------	--

---

## sendlocator (type dm\_queued)

**Purpose** Sends an email link (DRL, see [drl](#), page 698) to one or more selected documents, folders, tasks, notifications, or quickflow items using the HTML mailto command.

### Configuration File

webcomponent\config\library\sendlocator\sendlocator\_workflow\_items\_component.xml

### Parent Definition

sendlocator:webcomponent/config/library/sendlocator/sendlocator\_component.xml

### Scope

---

type	dm_queued
------	-----------

---

### Caller

This component is launched within the sendlocator container by the action [sendlocator \(type dm\\_queued\)](#), page 418, which is called by the menubar component UI. The sendlocator component has no UI.

### Parameters

These parameters are inherited from sendlocator:webcomponent/config/library/sendlocator/sendlocator\_component.xml.

---

objectId	(Required) ID of the folder or object for which to send a locator.
----------	--

---

## sendlocator (type dm\_router\_task)

**Purpose** Sends an email link (DRL, see [drl](#), page 698) to one or more selected documents, folders, tasks, notifications, or quickflow items using the HTML mailto command.

### Configuration File

```
webcomponent\config\library\sendlocator\sendlocator_workflow_items_component.xml
```

### Parent Definition

```
sendlocator:webcomponent/config/library/sendlocator/sendlocator_component.xml
```

### Scope

type	dm_router_task
------	----------------

### Caller

This component is launched within the sendlocator container by the action [sendlocator \(type dm\\_router\\_task\)](#), page 419, which is in turn called by the menubar component UI. The sendlocator component has no UI.

### Parameters

These parameters are inherited from sendlocator:webcomponent/config/library/sendlocator/sendlocator\_component.xml.

objectId	(Required) ID of the folder or object for which to send a locator.
----------	--

## sendlocator (type dm\_task)

**Purpose** Sends an email link (DRL, see [drl](#), page 698) to one or more selected documents, folders, tasks, notifications, or quickflow items using the HTML mailto command.

### Configuration File

webcomponent\config\library\sendlocator\sendlocator\_workflow\_items\_component.xml

### Parent Definition

sendlocator:webcomponent/config/library/sendlocator/sendlocator\_component.xml

### Scope

---

type	dm_task
------	---------

---

### Caller

This component is called by the action [sendlocator \(type dm\\_task\)](#), page 420 and displayed by the component [sendlocatorcontainer](#), page 1009. The sendlocator action is called by the menubar component UI. The sendlocator component has no UI.

### Parameters

These parameters are inherited from sendlocator:webcomponent/config/library/sendlocator/sendlocator\_component.xml.

---

objectId	(Required) ID of the folder or object for which to send a locator.
----------	--

---



## sendlocatorcontainer

**Purpose** This component is used to display a sendlocator component, which is used to send an email link (DRL, see [drl](#), page 698) to one or more selected documents, folders, tasks, notifications, or quickflow items using the HTML mailto command.

### Configuration File

webcomponent\config\library\sendlocator\sendlocatorcontainer\_component.xml

### Scope

All.

### Caller

This component is called by the action [sendlocator](#) (type `dm_queued`), page 418 and displays the component [sendlocator](#), page 1004.

### Parameters

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

## sendtodistributionlist

**Purpose** Sends objects to users within an ad hoc workflow (also known as quickflow).

## Configuration File

webcomponent\config\library\workflow\sendto\sendtodistributionlist\_component.xml

## Scope

All.

## Caller

This component is called by the action [sendtodistributionlist](#) (type `dm_sysobject`), page 421 and displayed in the component [multiargdialogcontainer](#), page 848.

## Parameters

---

objectId	(Required) ID of the object to send
----------	-------------------------------------

---

# setbindingrule

## Purpose

Launched by the `setbindingrule` action, this component extends `checkout` and allows the user to set a binding rule on a virtual document node. Displays the binding rule and version for the selected node. The container extends `checkoutcontainer` and performs the actual setting of the binding rule.

## Configuration File

webcomponent\config\library\vdm\setbindingrule\setbindingrule\_component.xml

## Scope

All.

**Caller**

This component is called by the action [setbindingrule](#) (type dm\_sysobject), page 422.

**Parameters**

---

nodeId	(Required) (Required) ID of the selected node for which to display the binding rule
vdmRootObjectId	(Required) (Required) ID of the root node of the virtual document to which the node is bound

---

## setretentiondate (type dm\_sysobject)

**Purpose** Sets a retention date for the object in its storage location. The action is scoped to the administrator role and is executed asynchronously.

**Configuration File**

webcomponent\config\library\retentiondate\setretentiondate\_component.xml

**Scope**

---

type	dm_sysobject
------	--------------

---

**Caller**

The action is scoped to the administrator role and is executed asynchronously.

## Parameters

---

lockOwner	(Optional) Name of user who has checked out the document
objectId	(Required) (Required) ID of the object for which the retention date is set.

---

# showtestcaseresult

**Purpose** For internal use only.

## Configuration File

webcomponent\config\testtool\showtestcaseresult\_component.xml

## Scope

All.

## Parameters

---

file	For internal use only.
index	(Required) For internal use only.
ticket	For internal use only.

---

# showtestresult

**Purpose** For internal use only.

## Configuration File

wdk\config\test\showtestresult\_component.xml

## Scope

All.

# showtopic

## Purpose

Displays a standalone topic page for objects that do not have embedded topics. The topic is rendered on the page along with the controls needed to add, edit, and delete comments. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\discussion\topicpage\_component.xml

## Scope

All.

## Parameters

---

folderPath	(Optional) Folder path of the parent topic
nodeIds	(Optional) List of dot-separated object IDs used to build an absolute location for display
objectId	(Required) ID of the parent object
type	(Optional) Parent object type, for example, dm_folder

---

## Elements

---

<comments><comment>	Specifies comment attributes to include.
<comment><attribute>	Sets the name of the attribute to include
<attribute><visible>	Boolean: set to true to display the attribute.

---

# spellchecker

**Purpose** Checks the spelling in the rich text editor. Requires the ActiveX plug-in to be enabled in app.xml and a successful plug-in download to the client. Rich text editing or display requires Collaborative Edition on the Content Server.

## Configuration File

webcomponent\config\library\richtext\spellchecker\_component.xml

## Scope

All.

## Usage

The calling window needs to set the following variables:  
window.contentWindow window: the iFrame ContentWindow  
window.layerDoc: the iFrame document

# sso\_login

**Purpose** Enables a user to log in to a content repository using the single sign-on (SSO) feature.

---

## Configuration File

wdk\config\sso\_login\_component.xml

## Parent Definition

login:wdk/config/login\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from login:wdk/config/login\_component.xml.

---

docbase	(Optional) Specifies the repository to log into
startComponent	(Optional) If an entry component name is specified, the login component will forward to that component after successful login.
startPage	(Optional) This parameter sets the component page that will be displayed on forward.
startUrl	(Optional) URL to forward to. If a startURL is specified, the login component will forward to that URL after successful login. This value takes precedence over startComponent. The URL must be relative to the root context, not relative to the directory location of the current JSP page.

---

## Usage

The component dispatcher must invoke this component to authenticate the user against a particular repository.

# startwfattachment

**Purpose** Manages the attachment information during the workflow initialization process. It is part of the startworkflow container component.

## Configuration File

webcomponent\config\library\workflow\attachment\startwfattachment\_component.xml

## Scope

All.

## Parameters

---

attachmentIds	(Optional) IDs of attachments to workflow
objectId	(Required) ID of the workflow process

---

# startwfcomments

**Purpose** Manages the process of adding notes to attachments during the workflow initialization process. It is part of the startworkflow component group.

## Configuration File

webcomponent\config\library\workflow\comments\startwfcomments\_component.xml

## Scope

All.



## Caller

This component is called by the actions [comment \(type dm\\_process\), page 248](#) (for dm\_process objects) and [comment \(type dm\\_task\), page 250](#)(for dm\_task objects). It is displayed in the component [startwfcontainer \(type dm\\_process\), page 1017](#).

## Parameters

objectId	(Required) ID of the workflow process
----------	---------------------------------------

## Elements

<showheaderforsinglepackage>	True to display the attached document header information when there is only one attached document
------------------------------	---

# startwfcontainer (type dm\_process)

**Purpose** Extends [propertysheetcontainer, page 914](#) and operates on dm\_process (template) objects.

## Configuration File

webcomponent\config\library\workflow\startwfcontainer\startwfcontainer\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

---

type	dm_process
------	------------

---

## Caller

This component is called by the actions [comment \(type dm\\_process\), page 248](#) and [startworkflow \(type dm\\_process\), page 426](#), and displays the components [startwfcomments, page 1016](#) and [startwfperformers, page 1018](#).

## Parameters

---

attachmentIds	(Optional) IDs of attachments to workflow
component	(Optional) Start component
objectId	(Required) ID of the workflow process
startworkflowId	(Required) ID of the workflow
type	(Required) Object type of the attachment, passed to the contained components

---

# startwfperformers

**Purpose** Manages the process of performer assignment. It is part of the startworkflow component group.

## Configuration File

webcomponent\config\library\workflow\performers\startwfperformers\_component.xml

## Scope

All.

## Parameters

---

objectId	(Required) ID of the workflow process
----------	---------------------------------------

---

# startworkflow

**Purpose** Displays the info for a workflow. The component has a classic-style layout. It is part of the group of task manager components.

## Configuration File

webcomponent\config\library\workflow\startworkflow\startworkflow\_component.xml

## Scope

All.

## Parameters

---

objectId	(Required) ID of the workflow process
----------	---------------------------------------

---

## Elements

---

formsavefolderpath	Path in the repository to associated form, if any. For example, /System/Workflow/Processing Form
--------------------	--

---

## startworkflowdrilldown

**Purpose** Extends [startworkflow](#), page 1019 and displays the Info tab on a workflow. The component has a streamline-style layout. It is part of the group of task manager components.

### Configuration File

```
webcomponent\config\library\workflow\startworkflow\startworkflowdrilldown_
component.xml
```

### Parent Definition

```
startworkflow:/webcomponent/config/library/workflow/startworkflow/startworkflow_
component.xml
```

### Scope

All.

### Parameters

These parameters are inherited from startworkflow:/webcomponent/config/library/workflow/startworkflow/startworkflow\_component.xml.

---

objectId	(Required) (Required) ID of the workflow to be started.
----------	---

---

## startworkflowtemplatelocatorcontainer

**Purpose** Extends [wftemplatelocatorcontainer](#), page 1132 and launches the workflow operation after the user selects the workflow template. The contained components locate various kinds of workflows and templates.

## Configuration File

webcomponent\config\library\workflow\startworkflow\startworkflowtemplatorcontainer\_component.xml

## Parent Definition

wftemplatorcontainer:webcomponent/config/library/locator/wftemplatorcontainers\_component.xml

## Scope

All.

## Caller

This component is called by the action [startworkflowfromdoc](#) (type `dm_sysobject`), [page 427](#) and displays the component [alluserwftemplator](#), [page 537](#). It can also display with no template component, when called by the action [startworkflownotemplate](#), [page 428](#).

## Parameters

componentArgs	(Optional) (Required) Supplied by the LaunchComponent class.) List of arguments and values to pass to each contained component.
objectId	(Optional) Attachment ID
startworkflowId	(Required) ID of the workflow

## Elements

For information on configurable columns, see [doclist](#), [page 684](#).

<iconwell_icon>	Path to the file containing the icons for display
<objecttype>	Base type to be located

<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document

---

<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## submitforcategorization

**Purpose** Submits the selected object to a task queue for categorization by the category manager.

## Configuration File

webcomponent\config\library\submitfor categorization\submitfor categorization\_ component.xml

## Scope

All.

## Caller

This component is called by the action [submitfor categorization](#) (type `dm_document`), [page 431](#) in the `menubar` component.

## Elements

This component has no defined parameters.

# subscriptionlocator

**Purpose** Extends [sysobjectlocator](#), [page 1029](#) and locates and displays a user's subscriptions.

## Configuration File

webcomponent\config\library\locator\subscriptionlocator\_ component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_ component.xml

## Scope

All.



## Parameters

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

For information on configurable columns, see [doclist, page 684](#).

# subscriptions\_drilldown

**Purpose** Displays subscriptions of the current session user.

## Configuration File

webcomponent\config\library\subscription\subscriptions\_drilldown\_component.xml

## Scope

All.

## Elements

<failoverenabled>	Controls serialization of the component state, and notifies the component in case of a failover recovery. Failover is enabled if this element is set to true.
<columns>.<loadinvisibleattribute>	Uncomment this element and set to true to get invisible attribute values for use in your component. The invisible attributes can then be passed by configuring a

	column in the <columns> element. See <i>Web Development Kit and Client Applications Development Guide</i> for details.
<columns><column>	Specifies columns to show or hide
<column><attribute>	<attribute> sets the attribute to be displayed in the column.
<attribute><label>	The value of <label> sets a label for the column.
<column><attribute><visible>	Set <visible> to true to show the column.

### Drag and Drop Elements

<dragdrop>	Supports the drag and drop feature.
<sourceactions>	Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.
<targetactions>	Contains zero or more <targetaction> classes that support drag and drop actions on targets in the component.
<targetaction>	Contains a fully qualified class name of the class that implements IDragTargetAction, for example, com.documentum.web.formext.control.dragdrop.CopyToFolderTargetAction.
<dataproviders>	Contains one or more <dataprovider> elements.
<dataprovider>	Contains a <format> element and a <provider> element.
<dataprovider>.<format>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<dataprovider>.<provider>	Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

# subscriptions\_list

**Purpose** Displays subscriptions of the current session user.

## Configuration File

webcomponent\config\library\subscription\subscriptions\_list\_component.xml

## Scope

All.

## Elements

<failoverenabled>	Controls serialization of the component state, and notifies the component in case of a failover recovery. Failover is enabled if this element is set to true.
<columns>.<loadinvisibleattribute>	Uncomment this element and set to true to get invisible attribute values for use in your component. The invisible attributes can then be passed by configuring a column in the <columns> element. See <i>Web Development Kit and Client Applications Development Guide</i> for details.
<columns><column>	Specifies columns to show or hide
<column><attribute>	<attribute> sets the attribute to be displayed in the column.
<attribute><label>	The value of <label> sets a label for the column.
<column><attribute><visible>	Set <visible> to true to show the column.
<showfilters>	Set to true to show the objectfilters drop-down control.
<showfolders>	Sets the user preference that specifies whether to display folders or not.

<objecttype> Sets the user preference for the object type to be displayed.

### Drag and Drop Elements

<dragdrop> Supports the drag and drop feature.

<sourceactions> Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.

<targetactions> Contains zero or more <targetaction> classes that support drag and drop actions on targets in the component.

<targetaction> Contains a fully qualified class name of the class that implements IDragTargetAction, for example, com.documentum.web.formext.control.dragdrop.CopyToFolderTargetAction.

<dataproviders> Contains one or more <dataprovider> elements.

<dataprovider> Contains a <format> element and a <provider> element.

<dataprovider>.<format> Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.

<dataprovider>.<provider> Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

## suspendqueuedtask

**Purpose** Puts a task from a work queue into a suspended state.

**Configuration File**

webcomponent\config\library\queuemanagement\suspendqueuedtask\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [suspend\\_queued\\_task](#) (type `dm_task`), page 432 and displayed in the component [combocontainer](#), page 644.

**Parameters**

---

objectId	(Required) ID of the task to suspend.
----------	---------------------------------------

---

## sysobjectlocator

**Purpose** Displays all `dm_sysobject` objects in a locator.

**Configuration File**

webcomponent\config\library\locator\sysobjectlocator\_component.xml

**Parent Definition**

objectlocator:webcomponent/config/library/locator/objectlocator\_component.xml

**Scope**

All.

## Parameters

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

The configurable elements in this component are the same as those in [objectlocator](#), page 886 with additional configurable elements described in the following table:

---

<columns>	Contains <column> elements to display or hide attributes
<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'>. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

---

# taskattachment

**Purpose**      Manages the attachments to tasks in the inbox.

## Configuration File

webcomponent\config\library\workflow\attachment\taskattachment\_component.xml

## Scope

All.

## Parameters

---

objectId	(Required) ID of the workflow process
----------	---------------------------------------

---

# taskcomments

**Purpose** Adds notes to inbox tasks. It is part of the task manager component.

## Configuration File

webcomponent\config\library\workflow\comments\taskcomments\_component.xml

## Scope

All.

## Caller

This component is called by the action [comment \(type dm\\_task\), page 250](#) and displayed in the component [taskmanagercontainer \(type dm\\_task\), page 1039](#).

## Parameters

---

objectId	(Required) ID of the workflow process
----------	---------------------------------------

---

## Elements

---

<code>&lt;showheaderforsinglepackage&gt;</code>	True to display the attached document header information when there is only one attached document
---	---

---

# taskcomponentcontainer

**Purpose** Extends the component [wizardcontainer, page 1148](#) and displays task components for workflows.

## Configuration File

webcomponent\config\library\workflow\taskmanager\taskcomponentcontainer\_component.xml

## Parent Definition

wizardcontainer:wdk/config/wizardcontainer\_component.xml

## Scope

All.

## Caller

This container is called by the action associated with each contained component.

## Parameters

---

component	(Required) Start component.
objectId	(Required) ID of the workflow process.



---

type	(Required) Task type. Valid values: dm_notification   dm_queued   dm_task. The type is passed by the taskmanagercontainer component to the contained components.
userCost	(Optional) Currency cost, such as dollars, euros, or other type of cost defined by customer. Value is stored as a double.
userTime	(Optional) Time in units defined by the customer. Value is stored as an integer.

---

## taskheader

**Purpose** Displays header information for tasks. It is part of the task manager and start workflow component groups.

### Configuration File

webcomponent\config\library\workflow\taskheader\taskheader\_component.xml

### Scope

All.

### Parameters

---

objectId	(Required) ID of the task
taskName	(Optional) Name of the task

---

# taskhistory

**Purpose** Displays runtime or audit history of a workflow that generated a specific task. It is part of the task manager component group.

## Configuration File

webcomponent\config\library\workflow\taskhistory\taskhistory\_component.xml

## Scope

All.

## Parameters

---

objectId	(Required) ID of the task
----------	---------------------------

---

## Elements

---

<historytype>	Specifies audit or runtime. Valid values: audit   auditorruntime (shows audit if available)   runtime (default = runtime)
<events>	Contains <visibility> elements which correspond to individual audit events
<visibility>	Corresponds to an event. Set to true to display the specified event. Set the name attribute to correspond to an audit type such as dm_signoff.

---

## taskinfo

**Purpose** Displays next and previous tasks information for specific tasks. It is part of the task manager.

### Configuration File

webcomponent\config\library\workflow\taskinfo\taskinfo\_component.xml

### Scope

All.

### Parameters

objectId	(Required) ID of the task
----------	---------------------------

### Elements

<showtaskinfo>	Configures the type of information to display. Contains <next> and <previous> elements.
<next>	True to show next information
<previous>	True to show previous information

## taskmanager

**Purpose** Displays the info tab in the taskmanager component group. It is the main component in the group of task manager components, which are contained in the taskmanagercontainer component.

## Configuration File

webcomponent\config\library\workflow\taskmanager\taskmanager\_component.xml

## Scope

All.

## Parameters

---

objectId	(Required) (Required) ID of the task
----------	--------------------------------------

---

## Elements

---

formsavefolderpath	Path in the repository to associated XForm, if any. For example, /System/Workflow/Processing Form
showusertimeandcost	Set to true to display the user time and cost attributes on the workitem object; set to false to turn off the display. These attributes have units that are set and interpreted by the customer.

---

# taskmanagercontainer (type dm\_notification)

**Purpose** Extends [propertysheetcontainer](#), page 914. This component is defined for following scopes: dm\_notification, dm\_queued, dm\_router\_task, and dm\_task. The definitions for dm\_notification, dm\_queued, and dm\_router\_task do not contain components.

## Configuration File

webcomponent\config\library\workflow\taskmgrcontainer\taskmanagercontainer\_component.xml

**Parent Definition**

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

**Scope**


---

type	dm_notification
------	-----------------

---

**Parameters**


---

component	(Optional) (Required) Specifies the ID of the component to initialize
objectId	(Required) Task ID
taskmanagerId	(Required) ID of the taskmanager object
type	(Required) (Required) Not used (any value can be inserted)

---

**taskmanagercontainer (type dm\_queued)**

**Purpose** Extends [propertysheetcontainer, page 914](#). This component is defined for following scopes: dm\_notification, dm\_queued, dm\_router\_task, and dm\_task. The definitions for dm\_notification, dm\_queued, and dm\_router\_task do not contain components.

**Configuration File**

webcomponent\config\library\workflow\taskmgrcontainer\taskmanagercontainer\_component.xml

**Parent Definition**

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

**Scope**

---

type	dm_queued
------	-----------

---

**Parameters**

---

component	(Optional) (Required) Specifies the ID of the component to initialize
objectId	(Required) Task ID
taskmanagerId	(Required) ID of the taskmanager object
type	(Required) Not used (any value can be inserted)

---

## taskmanagercontainer (type dm\_router\_task)

**Purpose** Extends [propertysheetcontainer, page 914](#). This component is defined for following scopes: dm\_notification, dm\_queued, dm\_router\_task, and dm\_task. The definitions for dm\_notification, dm\_queued, and dm\_router\_task do not contain components.

**Configuration File**

webcomponent\config\library\workflow\taskmgrcontainer\taskmanagercontainer\_component.xml

**Parent Definition**

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

**Scope**

---

type	dm_router_task
------	----------------

---

## Caller

This component is called by the action [view \(type dm\\_router\\_task\)](#), page 463. The component displayed is generated dynamically.

## Parameters

component	(Optional) (Required) Specifies the ID of the component to initialize
objectId	(Required) Task ID
taskmanagerId	(Required) ID of the taskmanager object
type	(Required) Not Used (any value can be inserted)

# taskmanagercontainer (type dm\_task)

**Purpose** Extends [propertysheetcontainer](#), page 914. This component is defined for following scopes: dm\_notification, dm\_queued, dm\_router\_task, and dm\_task. The definitions for dm\_notification, dm\_queued, and dm\_router\_task do not contain components.

## Configuration File

webcomponent\config\library\workflow\taskmgrcontainer\taskmanagercontainer\_component.xml

## Parent Definition

propertysheetcontainer:wdk/config/propertysheetcontainer\_component.xml

## Scope

type	dm_task
------	---------

## Caller

This component is called by the actions [comment \(type dm\\_task\), page 250](#) and [history \(type dm\\_task\), page 316](#), and displays the components [taskcomments, page 1031](#) and [taskprogress, page 1041](#), respectively.

## Parameters

---

component	(Optional) (Required) Specifies the ID of the component to initialize
objectId	(Required) Task ID
taskmanagerId	(Required) ID of the taskmanager object
type	(Required) (Required) Not used (any value can be inserted)

---

# taskmgrabortworkflow (type dm\_task)

**Purpose** Extends the abortworkflow component and aborts workflows that generated failed automatic workflow tasks. It is invoked from the taskmanager component.

## Configuration File

webcomponent\config\library\workflow\taskmanager\taskmgrabortworkflow\_component.xml

## Parent Definition

abortworkflow:webcomponent/config/library/workflow/abort/abortworkflow\_component.xml

## Scope

---

type	dm_task
------	---------

---



**Caller**

This component is called by the taskmanager component.

**Parameters**

These parameters are inherited from abortworkflow:webcomponent/config/library/workflow/abort/abortworkflow\_component.xml.

---

objectId	(Required) (Required) ID of the workflow object to be terminated
----------	--

---

## taskprogress

**Purpose** Contains taskhistory and taskinfo components. It is part of the task manager component group.

**Configuration File**

webcomponent\config\library\workflow\taskprogress\taskprogress\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [history \(type dm\\_task\), page 316](#) and is displayed in the component [taskmanagercontainer \(type dm\\_task\), page 1039](#).

**Parameters**

---

objectId	(Required) ID of the workflow process
----------	---------------------------------------

---

# taxonomylocator

**Purpose** This component is used to search for taxonomies.

## Configuration File

webcomponent\config\library\locator\taxonomylocator\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.

---

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

---

# testbed

**Purpose** Provides a testing environment for individual actions and components. Any action or component from the WDK library, WDK client application library, or custom library built on WDK can be tested. Also provides a common GUI for test automation.

## Configuration File

webcomponent\config\testbed\testbed\_component.xml

## Scope

All.

## Parameters

---

startComp	(Optional) Specifies the component that will be loaded in the content frame initially. Overrides the value of the <startComp> element.
-----------	--

---

## Elements

---

<startComp>	Specifies the component that will be loaded in the content frame initially
<pageTitle>	Title displayed for the testbed component. This title can be used to distinguish different versions of the testbed component used by other WDK-based applications.

<homeURL>	Specifies the URL that should be displayed when the logout action is invoked. The default value ensures that the testbed component is loaded again after the session is released.
<startQueryString>	Specifies additional query string parameters to pass to the component loaded in the content frame, for example, folderPath.

---

## testbed\_generic

**Purpose** Provides actionlink controls for invoking the actions defined by the application. The component also includes links to navigate to particular components or individual JSP pages. This component works within the menubar frame of the component [testbed](#), page 1043.

### Configuration File

webcomponent\config\componentTestbed\testbed\_generic\_component.xml

### Scope

All.

### Parameters

---

cut	(Optional) Name of component to test. Default = attributes
-----	---

---

## testcaseparameters

**Purpose** For internal use only.

---

## Configuration File

webcomponent\config\testtool\testcaseparameters\_component.xml

## Scope

All.

## Parameters

---

nodeId	(Required) For internal use only.
target	(Required) For internal use only.
ticket	(Required) For internal use only.

---

# testiteminfo

**Purpose** For internal use only.

## Configuration File

webcomponent\config\testtool\testiteminfo\_component.xml

## Scope

All.

## Parameters

---

type	(Required) For internal use only.
value	(Required) For internal use only.

---

## testlauncher

**Purpose** For internal use only.

### Configuration File

webcomponent\config\testtool\testlauncher\_component.xml

### Scope

All.

### Parameters

---

docbase	(Optional) For internal use only.
folderPath	(Optional) For internal use only.
password	(Optional) For internal use only.
username	(Optional) For internal use only.

---

## testlaunchermonitorticker

**Purpose** For internal use only.

### Configuration File

webcomponent\config\testtool\testlaunchermonitorticker\_component.xml

### Scope

All.

---

**Parameters**

---

ticket	(Required) For internal use only.
--------	-----------------------------------

---

## testlaunchermonitortree

**Purpose** For internal use only.

**Configuration File**

webcomponent\config\testtool\testlaunchermonitortree\_component.xml

**Scope**

All.

**Parameters**

---

ticket	(Required) For internal use only.
--------	-----------------------------------

---

## testlinks

**Purpose** Provides actionlink controls for invoking the actions defined by the application. The component also includes links to navigate to particular components or individual JSP pages. This component works within the menubar frame of the component [testbed](#), page 1043.

**Configuration File**

webcomponent\config\testbed\testbed\_component.xml

**Scope**

All.

## testmessage

**Purpose**

Displays the last status message that was generated by a component in the testbed component. It also provides simple search capability. This component loads within the message frame of the component [testbed, page 1043](#) and cannot be used outside the testbed component.

**Configuration File**

webcomponent\config\testbed\testbed\_component.xml

**Scope**

All.

**Caller**

This component loads within the message frame of the component [testbed, page 1043](#) and cannot be used outside the testbed component.

**Usage**

To see the full list of messages, use the Message List link in the Navigation column of wdlinks.jsp. The testbed.jsp page contains the onNewMessage() handler which reloads the testmessage component when a new message should be displayed.

Simple search is implemented using a form with a text field and button. A JavaScript function is invoked when the form is submitted, calling the search component.



---

# testrecorderlauncher

**Purpose** For internal use only.

## Configuration File

webcomponent\config\testtool\testrecorderlauncher\_component.xml

## Scope

All.

## Parameters

---

classname	(Optional) For internal use only.
configpath	(Optional) For internal use only.
docbase	(Optional) For internal use only.
folderPath	(Optional) For internal use only.
packagename	(Optional) For internal use only.
password	(Optional) For internal use only.
path	(Optional) For internal use only.
testCaseId	(Optional) For internal use only.
url	(Optional) For internal use only.
username	(Optional) For internal use only.

---

# testrecorderlaunchermonitortree

**Purpose** For internal use only.

## Configuration File

webcomponent\config\testtool\testrecorderlaunchermonitortree\_component.xml

## Scope

All.

## Parameters

---

classTicket	(Required) For internal use only.
isRecording	(Optional) For internal use only.
setupTicket	(Required) For internal use only.
tearDownTicket	(Required) For internal use only.

---

# testresultsviewer

**Purpose** For internal use only.

## Configuration File

webcomponent\config\testtool\testresultsviewer\_component.xml

## Scope

All.

# testtool

**Purpose** For internal use only.

---

## Configuration File

webcomponent\config\testtool\testtool\_component.xml

## Scope

All.

# testxmleditor

**Purpose** For internal use only.

## Configuration File

webcomponent\config\testtool\testxmleditor\_component.xml

## Scope

All.

## Parameters

---

recordmode	(Optional) For internal use only.
ticket	(Optional) For internal use only.

---

# transformationapplylifecycle (type dm\_sysobject)

**Purpose** Applies a lifecycle (business policy) to the selected transformed dm\_sysobject object. Requires Media Transformation Services.

## Configuration File

webcomponent\config\library\transformation\transformationapplylifecycle\_  
component.xml

## Parent Definition

transformationapplylifecyclecontainer:webcomponent/config/library/transformation/  
transformationapplylifecyclecontainer\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Parameters

---

component	(Required) Name of the component to contain.
objectId	(Optional) ID of the transformed object to which a lifecycle will be applied.
objectType	(Required) Content type of the object.

---

# transformationapplylifecycle (type dmr\_content)

**Purpose** Applies a lifecycle (business policy) to the selected transformed dmr\_content object. Requires Media Transformation Services.

## Configuration File

webcomponent\config\library\transformation\transformationapplylifecycle\_  
component.xml

**Parent Definition**

transformationapplylifecyclecontainer:webcomponent/config/library/transformation/transformationapplylifecyclecontainer\_component.xml

**Scope**

type	dmr_content
------	-------------

**Parameters**

component	(Required) Name of the component to contain.
objectId	(Optional) ID of the transformed object to which a lifecycle will be applied.
objectType	(Required) Content type of the object.

**transformationapplylifecycleall (type dm\_sysobject)**

**Purpose** Applies a lifecycle (business policy) to all the selected transformed dm\_sysobject objects. Requires Media Transformation Services.

**Configuration File**

webcomponent\config\library\transformation\transformationapplylifecycle\_component.xml

**Parent Definition**

applylifecycleall:webcomponent/config/library/applylifecycle/applylifecycle\_all\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Parameters

---

objectId	(Optional) ID of the transformed object to which a lifecycle will be applied.
objectType	(Required) Content type of the object.

---

# transformationapplylifecycleall (type dmr\_content)

**Purpose** Applies a lifecycle (business policy) to all the selected transformed dmr\_content objects. Requires Media Transformation Services.

## Configuration File

webcomponent\config\library\transformation\transformationapplylifecycle\_component.xml

## Parent Definition

applylifecycleall:webcomponent/config/library/applylifecycle/applylifecycle\_all\_component.xml

## Scope

---

type	dmr_content
------	-------------

---

**Parameters**

objectId	(Optional) ID of the transformed object to which a lifecycle will be applied.
objectType	(Required) Content type of the object.

**transformationapplylifecyclecontainer**

**Purpose** Base container for the transformationapplylifecycle component. Requires Media Transformation Services.

**Configuration File**

webcomponent\config\library\transformation\transformationapplylifecyclecontainer\_component.xml

**Parent Definition**

applylifecycle:webcomponent/config/library/applylifecycle/applylifecyclecontainer\_component.xml

**Scope**

All.

**Parameters**

These parameters are inherited from applylifecycle:webcomponent/config/library/applylifecycle/applylifecyclecontainer\_component.xml.

component	(Required) Name of the component within the container
objectId	(Required) ID of the object to which a lifecycle shall be applied

# transformationcontainer

**Purpose** Base container for transformation. Requires Media Transformation Services.

## Configuration File

webcomponent\config\library\transformation\transformationcontainer\_component.xml

## Parent Definition

propertywizardcontainer:webcomponent/config/library/  
propertywizardcontainer/propertywizardcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [importrenditiongeneric](#) (type `dmr_content`), [page 321](#) and [transformation](#) (type `dm_sysobject`), [page 440](#), and displays the component [choosetransformationtype](#), [page 639](#).

## Parameters

---

component	(Required) Supplied by the action execution class). The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.
folderId	(Optional) Not used.



---

objectId	(Required) ID of the transformed object.
renditionId	(Optional) ID of the content selected for transformation.

---

## Elements

---

<transformation>	Contains transformation specifications for rendition, relation, and container.
<rendition value='rendition value'>	Name of the transformation to rendition.
<relation value='relation value'>	Relation of the transformation to the new object container.
<choose value='transformation container name'>	Name of the transformation container.

---

# transformationcontainernewobject

**Purpose** Transformation container for the new object. Requires Media Transformation Services.

## Configuration File

webcomponent\config\library\transformation\transformationcontainer\_newobject\_component.xml

## Parent Definition

transformationcontainer:webcomponent/config/library/transformation/transformationcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [transformationtonewrelation](#) (type `dm_sysobject`), [page 440](#), and displays the component [transformationselect](#), [page 1061](#).

## Parameters

These parameters are inherited from `transformationcontainer:webcomponent/config/library/transformation/transformationcontainer_component.xml`.

---

<code>component</code>	(Required) Supplied by the action execution class). The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.
<code>folderId</code>	(Optional) Not used.
<code>objectId</code>	(Required) ID of the selected object.
<code>renditionId</code>	(Optional) ID of the content selected for transformation.

---

# transformationcontainerrendition

**Purpose** Transformation container for the new rendition. Requires Media Transformation Services.

## Configuration File

`webcomponent\config\library\transformation\transformationcontainer_rendition_component.xml`

## Parent Definition

`transformationcontainer:webcomponent/config/library/transformation/transformationcontainer_component.xml`

## Scope

All.

## Caller

This component is called by the actions [transformationtonewrendition \(type dm\\_sysobject\)](#), [page 442](#) and [transformationtonewrendition \(type dmr\\_content\)](#), [page 443](#), and displays the components [transformationselect](#), [page 1061](#) and [choosetransformationtype](#), [page 639](#).

## Parameters

These parameters are inherited from transformationcontainer:webcomponent/config/library/transformation/transformationcontainer\_component.xml.

---

component	(Required) Supplied by the action execution class). The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.
folderId	(Optional) Not used.
objectId	(Required) ID of the selected object.
renditionId	(Optional) ID of the content selected for transformation.

---

# transformationdetails

**Purpose** Specifies transformation details for an object. Requires Media Transformation Services.

## Configuration File

webcomponent\config\library\transformation\transformationdetails\_component.xml

**Scope**

All.

**Elements**

---

<controlproperties>	Contains specifications for the control
<textareanumcols value='number'>	Number of columns of the text block
<textareanumrows value="number">	Number of rows of the text block
<thumbnailheight value="number">	Height of the content object thumbnail
<thumbnailwidth value="number">	Width of the content object thumbnail
<listboxsize value="number">	Size of the list box
<colorpickernumrows value="number">	Number of rows of the color picker
<colorpickerwidth value="number">	Width of the color picker
<colorpickerheight value="number">	Height of the color picker

---

## transformationnewobjectdef

**Purpose** Provides definition of the new transformation object. Requires Media Transformation Services.

**Configuration File**

webcomponent\config\library\transformation\transformationnewobjectdef\_  
component.xml

**Scope**

All.

## Parameters

---

objectId	(Required) ID of the selected object.
----------	---------------------------------------

---

# transformationselect

**Purpose** A transformation selection component. Requires Media Transformation Services. Enables the user to select the available media profiles.

## Configuration File

webcomponent\config\library\transformation\transformationselect\_component.xml

## Scope

All.

## Caller

This component is called by the actions [transformationtonewrendition \(type dm\\_sysobject\)](#), page 442, [transformationtonewrendition \(type dmr\\_content\)](#), page 443, [transformationtonewrelation \(type dm\\_sysobject\)](#), page 440 and [transformationtonewrelation \(type dmr\\_content\)](#), page 441, and displayed in the components [transformationcontainerrendition](#), page 1058 and [transformationcontainernewobject](#), page 1057.

# ucfinvoker

**Purpose** Enables invoking a component using the unified client facilities feature.

## Configuration File

wdk\config\ucfinvoker\_component.xml

## Scope

All.

## Parameters

---

targetComponent	(Optional) ID of the component to invoke.
targetComponentPage	(Optional) Name of the component page to render as named in the component definition

---

## Elements

---

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
---------------	--

---

# unassignqueuedtask

**Purpose** Removes a work item from a particular work queue processor's Inbox.

## Configuration File

webcomponent\config\library\queuemanagement\unassignqueuedtask\_component.xml

## Scope

All.

## Caller

This component is called by the action [unassign\\_queued\\_task](#) (type `dm_task`), page 444 and displayed in the component [unassignqueuedtaskcontainer](#), page 1063.

## Parameters

---

objectId	(Required) ID of the task to unassign.
queueName	(Required) Name of the work queue containing the task.

---

# unassignqueuedtaskcontainer

**Purpose** Container component for removing work items from a particular work queue processor's Inbox.

## Configuration File

webcomponent\config\library\queuemanagement\unassignqueuedtaskcontainer\_component.xml

## Parent Definition

multiargdialogcontainer:webcomponent/config/library/multiargdialogcontainer/multiargdialogcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [unassign\\_queued\\_task](#) (type `dm_task`), page 444 and displays the component [unassignqueuedtask](#), page 1062.

## Parameters

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

# unfilteredcontainersysobjectlocator

**Purpose** Extends [sysobjectlocator](#), page 1029 and allows the user to navigate the repository from the root cabinet in order to locate a dm\_sysobject. Two types of views are supported: flatlist, in which all selectable objects in the repository are shown, and hierarchical, in which the user can drill down from a list of root container (cabinets).

## Configuration File

webcomponent\config\library\locator\unfilteredcontainersysobjectlocator\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.



contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

Inherited from [sysobjectlocator](#), page 1029.

# ungovern (type dm\_folder)

**Purpose** Confirms that objects are being removed from a Room. Allows users to ungovern just a folder or the folder and all children. Children that are governed by a different room than the folder remain governed. Requires Collaborative Edition.

## Configuration File

webcomponent\config\library\rooms\ungovern\_component.xml

## Scope

type	dm_folder
------	-----------

## Parameters

objectId	(Required) ID of the object being removed
----------	---

## unsuspendqueuedtask

**Purpose** Returns a suspended task in a work queue to active state.

### Configuration File

webcomponent\config\library\queuemanagement\unsuspendqueuedtask\_  
component.xml

### Scope

All.

### Parameters

---

objectId	(Required) ID of the suspended task.
----------	--------------------------------------

---

## updateprocessorauthentication

**Purpose** Changes the skill level of a work queue processor or the work queue document profile that the processor is authorized to process.

### Configuration File

webcomponent\config\library\queuemanagement\updateprocessorauthentication\_  
component.xml

### Parent Definition

editprocessorauthentication:webcomponent/config/library/queuemanagement/  
editprocessorauthentication\_component.xml

**Scope**

All.

**Parameters**

---

docprofile_name	(Optional) Name of a work queue document profile
skill	(Optional) Skill level of the user for this work queue. Valid values are the skill level strings from the data dictionary.
username	(Required) Name of the user.
wq_name	(Required) Name of the work queue.

---

## useracobjectlocator

**Purpose**        Browses for ACLs that are owned by the user.

**Configuration File**

webcomponent\config\admin\user\useracobjectlocator\_component.xml

**Parent Definition**

acobjectlocator:webcomponent/config/library/locator/acobjectlocator\_component.xml

**Scope**

All.

**Caller**

This component is called by the action [aclbrowse](#) (privilege sysadmin, type dm\_user), [page 208](#).

## Parameters

These parameters are inherited from `acobjectlocator:webcomponent/config/library/locator/acobjectlocator_component.xml`.

---

<code>flatlist</code>	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<code>multiselect</code>	(Optional) Boolean: Set to true to support multiple object selection.
<code>objecttype</code>	(Optional) Type of the selected object

---

# useracobjectlocatorcontainer

**Purpose** Browsing for ACLs that are owned by the user. The container extends the container `acobjectlocator`, [page 488](#).

## Configuration File

`webcomponent\config\admin\user\useracobjectlocator_component.xml`

## Parent Definition

`acobjectlocatorcontainer:webcomponent/config/library/locator/acobjectlocator_component.xml`

## Scope

All.

## Parameters

These parameters are inherited from `locatorcontainer:webcomponent/config/library/locator/locatorcontainer_component.xml`.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) Name of the Documentum repository from which to retrieve the objects for display
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## Elements

---

objecttype	Type of the selected object
flatlist	Boolean: true to list all objects in a flat list, false to display a drop down list.
multiselect	Boolean: true to support multiple object selection
selectedobjectids	Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

# userattributes

**Purpose** Creates new user profiles and edits existing user profiles. Administrators can set all the attributes of the user object, such as name, email, privileges, and default group.

## Configuration File

webcomponent\config\admin\user\userattributes\_component.xml

## Scope

All.

## Caller

This component is called by the properties, attributes, and newuser actions for objects of type dm\_member\_user and dm\_user.

## Parameters

---

objectId	(Required) Object ID of the user object whose attributes to set.
objectname	(Required) Object name of the user object whose attributes to set.

---

# userchangehomedblist

**Purpose** Displays user change home repository job requests and their status.

## Configuration File

webcomponent\config\admin\renamelog\userchangehomedblist\_component.xml

## Scope

All.

**Caller**

This component is called by the changehomedblog action in the menubar UI.

## userchangestate

**Purpose**

Changes the state of the user. When the administrator changes the state of the user from active to inactive, this component offers the choice of unlocking all objects locked by the user.

**Configuration File**

webcomponent\config\admin\user\userchangestate\_component.xml

**Scope**

All.

**Parameters**

---

objectId	(Required) Object ID of the user object whose state to change.
objectname	(Required) Object name of the user object whose state to change.

---

## userdelete

**Purpose**

Deletes an existing user profile from the repository. The component warns the administrator of the repercussions of performing this task and offers alternative solutions.

**Configuration File**

webcomponent\config\admin\user\userdelete\_component.xml

**Scope**

All.

**Parameters**

---

objectId	(Required) Object ID of the user object to delete.
objectname	(Required) Object name of the user object to delete.

---

## userimport

**Purpose**

Imports users through a LDIFF (list difference) utility. The administrator can select the ldiff format file containing information about all the users. The component also offers the choice of overriding any of the user attributes. The component must run within the userimportcontainer.

**Configuration File**

webcomponent\config\admin\user\userimport\_component.xml

**Scope**

All.



# userimportcontainer

**Purpose** Extends [contentxfercontainer](#), page 653 and imports users through a LDIF (list difference) utility. The administrator can select the ldif format file containing information about all the users.

## Configuration File

webcomponent\config\admin\user\userimportcontainer\_component.xml

## Parent Definition

contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## userlist

**Purpose** Searches for a user or pages through a list of all the users in the repository.

### Configuration File

webcomponent\config\admin\user\userlist\_component.xml

### Scope

All.

### Caller

This component is called by the action [searchuser](#) (type `dm_user`), [page 415](#) and is also included as a node in the component [usermanagement](#), [page 1074](#).

## usermanagement

**Purpose** Extends [nodemanagement](#), [page 882](#) and defines the subnodes (components) of the usermanagement node: `userlist`, `group`, and `role`

### Configuration File

webcomponent\config\admin\usermanagement\usermanagement\_component.xml

### Parent Definition

nodemanagement:webcomponent/config/admin/nodemanagement/nodemanagement\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from `nodemanagement:webcomponent/config/admin/nodemanagement/nodemanagement_component.xml`.

<code>nodeIds</code>	(Optional) Target nodes in the tree
----------------------	-------------------------------------

## Elements

<code>&lt;nodes&gt;</code>	Contains <code>&lt;node&gt;</code> elements that specify the nodes to manage
<code>&lt;node&gt;</code>	Defines a node and its handler. Contains <code>&lt;icon&gt;</code> , <code>&lt;label&gt;</code> , <code>&lt;handlerclass&gt;</code> , and <code>&lt;description&gt;</code> elements. Set the <code>componentid</code> attribute to specify the component that will be launched from a tree node in the admin branch of the tree.
<code>&lt;icon&gt;</code>	Specifies the name of an icon to be used for the node
<code>&lt;label&gt;</code>	Specifies a label to be used for the node
<code>&lt;handlerclass&gt;</code>	Specifies the class that will handle navigation within the node
<code>&lt;description&gt;</code>	String that describes the node

# useronlylocator

## Purpose

Locates users in a repository. The component allows users to navigate from user groups to find a user. This component extends [userorgrouplocator](#), page 1079.

## Configuration File

webcomponent\config\library\locator\useronlylocator\_component.xml

## Parent Definition

userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml.

---

arrangeselect	(Optional) Boolean: Set to true to list users in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements

<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.

<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## useronlylocatorcontainer

**Purpose** Extends [locatorcontainer](#), page 839 and contains user locators: [useronlylocator](#), page 1075 and [recentuseronlylocator](#), page 927.

### Configuration File

webcomponent\config\library\locator\useronlylocator\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

# userorgrouplocator

**Purpose** Extends [objectlocator](#), page 886 and locates users and groups in a repository.

## Configuration File

webcomponent\config\library\locator\userorgrouplocator\_component.xml

## Parent Definition

objectlocator:webcomponent/config/library/locator/objectlocator\_component.xml

## Scope

All.

## Parameters

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of users and groups in a sequential order.
flatlist	(Optional) Boolean: Set to true to list users and groups in a flat list, set to false to display users and groups in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple user/group selection.
objecttype	(Optional) Type of the selected object

---

## Elements

For information on configurable columns, see [doclist, page 684](#).

---

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.



<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)

<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## userorgrouplocatorcontainer

**Purpose** Extends [locatorcontainer](#), page 839 and contains the locator components [userorgrouplocator](#), page 1079 and [recentuserorgrouplocator](#), page 928.

### Configuration File

webcomponent\config\library\locator\userorgrouplocator\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

### Scope

All.

## Parameters

arrangeselect	(Optional) Boolean: Set to true to allow selection of objects in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
selectedobjectids	(Optional) IDs of the objects selected when the component is displayed

# userorgrouplelocator

**Purpose** Extends [userorgrouplelocator](#), page 1079 and locates users and group members in a repository.

## Configuration File

webcomponent\config\admin\group\userorgrouplelocator\_component.xml

## Parent Definition

userorgrouplelocator:webcomponent/config/library/locator/userorgrouplelocator\_component.xml

## Scope

All.

## Caller

The component is called by the action [adduserorgrouple](#) (type [grouplist](#)), page 222 in the menubar component.

## Parameters

These parameters are inherited from userorgrouplelocator:webcomponent/config/library/locator/userorgrouplelocator\_component.xml.

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of objects in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.

<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)

<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute <code>dqlformatted='false'</code> to quote and escape a value. Use the attribute <code>casesensitive='true'</code> to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## userorgrouplelocatorcontainer

**Purpose** Extends [userorgrouplelocatorcontainer](#), page 1082 and contains the locator components [userorgrouplelocator](#), page 1083 and [recentuserorgrouplelocator](#), page 928.

### Configuration File

`webcomponent\config\admin\group\userorgrouplelocator_component.xml`

### Parent Definition

`locatorcontainer:webcomponent/config/library/locator/locatorcontainer_component.xml`

### Scope

All.

## Parameters

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

# userproperties

**Purpose** Extends [adminpropertycontainer, page 515](#) and overrides some of the behavior of the container. Contains [userattributes, page 1069](#).

## Configuration File

webcomponent\config\admin\user\userproperties\_component.xml

## Parent Definition

adminpropertycontainer:webcomponent/config/admin/container/  
adminpropertycontainer\_component.xml

## Scope

All.

## Caller

This container is launched by the properties, attributes, and newuser actions for objects of type dm\_member\_user and dm\_user.

## Parameters

---

objectId	(Required) Object ID of the user.
objectname	(Required) Object name of the user.

---

## Elements

---

<objecttype>.<object>	Type of object that is instantiated by the container
-----------------------	--

---

# userrename

**Purpose** Reassigns an existing user. The administrator is offered choices such as reassigning all the objects owned by the current user and generating a report.

## Configuration File

webcomponent\config\admin\user\userrename\_component.xml

## Scope

All.

## Caller

This component is called by the action [reassignuser](#) (privilege `superuser`, type `dm_user`), [page 376](#).



## Parameters

---

objectId	(Required) Object ID of the user to rename.
objectname	(Required) Current object name of the user.

---

# userwhereused

**Purpose** Finds where a selected user has activity, for example, all sysobjects owned by the user, groups in which the user exists, and workflows for which the user is a designated performer.

## Configuration File

webcomponent\config\admin\user\userwhereused\_component.xml

## Scope

All.

## Caller

This component is called by the action [locations \(type dm\\_user\)](#), [page 328](#) for dm\_user objects.

## Parameters

---

objectname	(Required) Name of the user
------------	-----------------------------

---

## vdm\_preferences

**Purpose** Enables the user to set preferences for Virtual Document Manager (VDM).

### Configuration File

webcomponent\config\environment\preferences\vdm\vdm\_preferences\_component.xml

### Scope

All.

## vdmclickactionprompt

**Purpose** Displays a dialog when the user clicks a link to a virtual document that allows the user to open the virtual document in an editing application or open the virtual document manager to edit the structure.

### Configuration File

webcomponent\config\navigation\vdm\vdmclickactionprompt\_component.xml

### Scope

All.

### Caller

This component is called by the action [vdmclickactionprompt](#) (type `dm_sysobject`), page [453](#) and displayed in the component [vdmclickactionpromptcontainer](#), page [1091](#).

## Parameters

---

contentSize	(Optional) Size of the object in bytes
contentType	(Optional) File format to use when opening the object (passed to the operating system)
isReference	(Optional) Boolean flag that indicates whether the object is a reference object
launchViewer	(Optional) Boolean flag that indicates whether to launch a viewing application.
nodeId	(Optional) ID of the node in a virtual document
objectId	(Required) ID of the object to be viewed.
pathInfo	(Optional) Path to be displayed in the breadcrumb
selectedNodeIds	(Optional) Comma-separated list of nodeIds that should be initially selected when the vdmclist component is first displayed
structureComponent	(Optional) Name of the component that displays virtual document structure
useAssembly	(Optional) Set to true for virtual document assembly objects

---

## Elements

The configurable elements are the same as those in [doclist](#), page 684.

# vdmclickactionpromptcontainer

**Purpose**      Contains one or more vdmclickactionprompt components

## Configuration File

webcomponent\config\navigation\vdm\vdmclickactionpromptcontainer\_component.xml

## Parent Definition

dialogcontainer:wdk/config/dialogcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [vdmclickactionprompt](#) (type `dm_sysobject`), page 453 and displays component [vdmclickactionprompt](#), page 1090 .

## Parameters

These parameters are inherited from `container:wdk/config/container_component.xml`.

---

component	(Required) Contained component.
-----------	---------------------------------

---

# vdmcopyoption

## Purpose

Launches the `vdmcopyoption` component in the `dialogcontainer` to present the user with options for the type of copy operation on a virtual document.

## Configuration File

webcomponent\config\library\vdm\copyoption\copyoption\_component.xml

---

**Scope**

All.

**Caller**

This component is called by the action [vdmcopyoption](#) (type `dm_sysobject`), page 454 and is displayed in the component [combocontainer](#), page 644.

**Parameters**

---

objectName	(Required) Name of the object that will be copied.
------------	--

---

## vdmlist

**Purpose**

Displays the structure of a virtual document. The user can navigate to the different parts of the virtual document by clicking on the breadcrumb or the child node. This component has an extended version in Webtop called `vdmlist`.

**Configuration File**

`webcomponent\config\navigation\vdm\vdmlist_component.xml`

**Scope**

All.

**Caller**

This component is called by the actions [viewvdm](#) (type `dm_sysobject`), page 475 and [viewvirtualdoc](#) (type `dm_sysobject`), page 476.

## Parameters

---

nodeId	(Optional) ID of the node in a virtual document.
objectId	(Required) ID of the virtual document.
pathInfo	(Optional) Path to be displayed in the breadcrumb.
selectedNodeIds	(Optional) Comma-separated list of nodeIds that should be initially selected when the vdmlist component is first displayed.

---

## Elements

---

<default_binding_filter_values>	List of version labels to include in the default child version dropdown. These values are in addition to "CURRENT" and existing version labels on the virtual document root node.
<version_label>	Additional label to display, for example, Approved
<dragdrop>	Supports the drag and drop feature.
<sourceactions>	Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.
<targetactions>	Contains zero or more <targetaction> classes that support drag and drop actions on targets in the component.
<targetaction>	Contains a fully qualified class name of the class that implements IDragTargetAction, for example, com.documentum.web.formext.control.dragdrop.CopyToFolderTargetAction.
<dataproviders>	Contains one or more <dataprovider> elements.
<dataprovider>	Contains a <format> element and a <provider> element.

---

<dataprotider>.<format>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files. <Ask Lani about (Need List)>
<dataprotider>.<provider>	Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For information on configurable columns, see [doclist](#), page 684.

## Usage

To make the late binding value display by default for each node, make the `r_object_id` column visible in the component definition. Users who customize the column display can add this as the "Fix to Version" column.

# vdmstreamline

**Purpose** Provides a streamline view of virtual documents. Both the vdmstreamline components provide a UI that displays the children and root in a virtual document, allows navigation through the virtual document's hierarchy, and displays menus to launch actions on selected items.

## Configuration File

webcomponent\config\navigation\ vdm\vdmstreamline\_component.xml

## Scope

All.

## Caller

This component is called by the action [streamlineviewvirtualdoc](#) (type `dm_sysobject`), [page 430](#).

## Parameters

<code>home_component</code>	(Optional) In Portal application, specifies portal component that calls this component. Used to construct a breadcrumb back to the portal.
<code>home_component_args</code>	(Optional) In Portal application, specifies arguments that are passed to the portal component that calls this component when the user clicks the breadcrumb to go back.
<code>nodeId</code>	(Optional) ID of the node in a virtual document.
<code>objectId</code>	(Required) ID of the virtual document.
<code>pathInfo</code>	(Optional) Path to be displayed in the breadcrumb.

## Elements

For information on configurable columns, see [doclist](#), [page 684](#).

<code>&lt;default_binding_filter_values&gt;</code>	Contains a list of version labels to be included in the default child version drop-down list. These values are added in addition to CURRENT and any other version labels, which may exist on the virtual document root node.
<code>&lt;default_binding_filter_values&gt;.&lt;version_label&gt;</code>	Add any version labels for versions that should be displayed. For example, to display approved versions only, add the value "Approved".

### Drag and Drop Elements

<code>&lt;dragdrop&gt;</code>	Supports the drag and drop feature.
-------------------------------	-------------------------------------



---

<sourceactions>	Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.
<targetactions>	Contains zero or more <targetaction> classes that support drag and drop actions on targets in the component.
<targetaction>	Contains a fully qualified class name of the class that implements IDragTargetAction, for example, com.documentum.web.formext.control.dragdrop.CopyToFolderTargetAction.
<dataproviders>	Contains one or more <dataprovider> elements.
<dataprovider>	Contains a <format> element and a <provider> element.
<dataprovider>.<format>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<dataprovider>.<provider>	Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

## Usage

To make the late binding value display by default for each node, make the `r_object_id` column visible in the component definition. Users who customize the column display can add this as the "Fix to Version" column.

# versionlabels

**Purpose** Allows editing of version labels

## Configuration File

wdk\config\versionlabels\_component.xml

## Parent Definition

docbaserepeatingattribute:wdk/config/docbaserepeatingattribute\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from docbaserepeatingattribute:wdk/config/docbaserepeatingattribute\_component.xml.

---

control	(Required) ID of the docbaseattributevalue control.
---------	---

---

# versions (type dm\_sysobject)

**Purpose** Extends [objectgrid](#), page 884, displays and allows users to edit version information on an object.

## Configuration File

webcomponent\config\library\versions\versions\_component.xml

## Parent Definition

objectgrid:/webcomponent/config/navigation/objectgrid/objectgrid\_component.xml

## Scope

type	dm_sysobject
------	--------------

## Caller

This component is called by the action [versions \(type dm\\_sysobject\), page 456](#) and displayed in the component [navigationcontainer, page 864](#).

## Parameters

objectId	(Required) ID of the selected object
----------	--------------------------------------

## Elements

The default columns to be displayed are configured in the <columns> element. For information on this configuration, see [doclist, page 684](#). Columns for non-Documentum attributes are configured in the <nondocbasecolumns> element. Attributes named in the <nondocbasecolumns> element are not added to the datagrid query. For example, the path attribute is resolved by a [primaryfolderpathlink, page 147](#) control in the JSP page, and the folder path is displayed in the UI.

<dragdrop>	Supports the drag and drop feature.
<sourceactions>	Contains zero or more <sourceaction> classes that support drag and drop actions on sources in the component.
<targetactions>	Contains zero or more <targetaction> classes that support drag and drop actions on targets in the component.
<targetaction>	Contains a fully qualified class name of the class that implements IDragTargetAction, for example, com.documentum.web.formext.control.dragdrop.CopyToFolderTargetAction.
<dataproviders>	Contains one or more <dataprovider> elements.
<dataprovider>	Contains a <format> element and a <provider> element.

<dataProvider>.<format>	Fully qualified class name for the class that provides data for the format. Built-in formats include object ID, tree node, and external files.
<dataProvider>.<provider>	Fully qualified class name for the class that implements IDragDropDataProvider and provides data for the associated format.

---

For more information on the drag and drop feature, see "Supporting Drag and Drop" in *Web Development Kit and Client Applications Development Guide*.

## versions (type wdk5\_testtype\_1)

**Purpose** Extends [objectgrid](#), [page 884](#), displays, and edits version information on an object.

### Configuration File

wdk\config\fxtest\_component.xml

### Parent Definition

versions:/webcomponent/config/library/versions/versions\_component.xml

### Scope

---

type	wdk5_testtype_1
------	-----------------

---

### Parameters

These parameters are inherited from versions:/webcomponent/config/library/versions/versions\_component.xml.

---

objectId	(Required) ID of the selected object
----------	--------------------------------------

---

## versions (type wdk5\_testtype\_2)

**Purpose** Extends [objectgrid, page 884](#) and displays and edits version information on an object.

### Configuration File

wdk\config\fxtest\_component.xml

### Parent Definition

versions:/webcomponent/config/library/versions/versions\_component.xml

### Scope

type	wdk5_testtype_2
------	-----------------

### Parameters

These parameters are inherited from versions:/webcomponent/config/library/versions/versions\_component.xml.

objectId	(Required) ID of the selected object
----------	--------------------------------------

## versions (type wdk5\_testtype\_3)

**Purpose** Extends [objectgrid, page 884](#) and displays and edits version information on an object.

### Configuration File

wdk\config\fxtest\_component.xml

### Parent Definition

versions:/webcomponent/config/library/versions/versions\_component.xml

### Scope

---

type	wdk5_testtype_3
------	-----------------

---

### Parameters

These parameters are inherited from versions:/webcomponent/config/library/versions/versions\_component.xml.

---

objectId	(Required) ID of the selected object
----------	--------------------------------------

---

## view\_record\_relationships (entitlement recordsmanager)

**Purpose** View relationships between records. Requires a Records Manager license.

### Configuration File

webcomponent\config\library\records\relationships\record\_relationships.xml

### Parent Definition

objectgrid:/webcomponent/config/navigation/objectgrid/objectgrid\_component.xml

### Scope

---

entitlement	recordsmanager
-------------	----------------

---

**Caller**

Called by the action and runs in the [navigationcontainer, page 864](#) container.

**Parameters**


---

objectId	(Required) Object ID of the source record.
----------	--

---

**view (type dm\_sysobject, version 5.2.5)****Purpose**

This is the WDK 5.2.5 view component for dm\_sysobjects. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to download repository content to the browser local file system and open a viewing application on the client.

**Configuration File**

webcomponent\config\library\view\view\_component.xml

**Scope**


---

type	dm_sysobject
version	5.2.5

---

**Caller**

Cannot be called directly. Must be extended.

**Parameters**


---

contentType	(Optional) Type of object to view
objectId	(Required) ID of the object to be viewed

---

## view (type dm\_sysobject)

**Purpose** This is a WDK 5.3 view component. It is used to download repository content to the browser local file system and open a viewing application on the client. The view component must run in the viewcontainer component. This component is defined for dm\_sysobject and dmr\_content scopes. The component is undefined for dm\_folder using the notdefined attribute, so that folders cannot be viewed with a viewing application. The viewcontainer extends [contentxfercontainer](#), page 653.

### Configuration File

webcomponent\config\library\contenttransfer\view\view\_component.xml

### Scope

---

type	dm_sysobject
------	--------------

---

### Caller

This component is called by the actions [downloadcontent](#) (type dm\_sysobject), page 281, [viewcontent](#) (type dm\_sysobject), page 472, and [view](#) (type dm\_sysobject), page 465, and is displayed in the components [downloadcontentcontainer](#) (type dm\_sysobject), page 691 and [viewcontainer](#) (type dm\_sysobject), page 1109.

### Parameters

---

contentType	(Optional) Type of object to view
isIgnoreDescendents	(Optional) Boolean value that indicates whether or not to download descendants in a virtual document.
objectId	(Required) ID of the object to be viewed
pageModifier	(Optional) A valid value for the page_modifier attribute of the object. You should not pass in your own value for PDF renditions.

---



## Elements

---

<ucfrequired>	If this element is present, then the component requires UCF to be installed.
---------------	--

---

## view (type dmr\_content, version 5.2.5)

**Purpose** This is the WDK 5.2.5 view component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to download repository content to the browser local file system and open a viewing application on the client.

## Configuration File

webcomponent\config\library\view\view\_component.xml

## Scope

---

type	dmr_content
version	5.2.5

---

## Caller

Cannot be called directly. Must be extended.

## Parameters

---

contentType	(Optional) Type of object to view
objectId	(Required) ID of the object to be viewed

---

## view (type dmr\_content)

**Purpose** This is a WDK 5.3 view component. It is used to download repository content to the browser local file system and open a viewing application on the client. The view component must run in the viewcontainer component. This component is defined for dm\_sysobject and dmr\_content scopes. The component is undefined for dm\_folder using the notdefined attribute, so that folders cannot be viewed with a viewing application. The viewcontainer extends [contentxfercontainer](#), page 653.

### Configuration File

webcomponent\config\library\contenttransfer\view\view\_component.xml

### Scope

---

type	dmr_content
------	-------------

---

### Caller

This component is called by the action [view \(type dmr\\_content\)](#), page 468, and displayed in the component [viewcontainer \(type dmr\\_content\)](#), page 1112.

### Parameters

---

contentType	(Optional) Type of object to view
objectId	(Required) ID of the object to be viewed
pageModifier	(Optional) A valid value for the page_modifier attribute of the dmr_content object. You should not pass in your own value for PDF renditions.

---

# viewassemblies

**Purpose** Displays a list of all assemblies that contain a specified object

## Configuration File

webcomponent\config\library\vdm\viewassemblies\viewassemblies\_component.xml

## Parent Definition

objectgrid:/webcomponent/config/navigation/objectgrid/objectgrid\_component.xml

## Scope

All.

## Caller

This component is called by the action [viewassemblies](#) (type `dm_sysobject`), page 470.

## Parameters

---

assembledFromId	(Optional) ID of the virtual document from which the assembly was assembled
folderId	(Optional) Specifies the ID of the primary location for the folder in which the assembly is located
folderPath	(Optional) Specifies Specifies the path to the folder in which to start browsing. This path must be a complete path from the Docbase root, for example, /Documentation%20Library/subfolder

isVirtualDoc	(Optional) Boolean flag that specifies whether the object is a virtual document (required for pre-4.1 repository virtual documents)
linkCount	(Optional) Number of links (child nodes) in a document. If greater than 0, the document is a virtual document.
objectId	(Required) ID of the assembly to view

---

## viewcontainer (type dm\_sysobject, version 5.2.5)

**Purpose** This is the WDK 5.2.5 viewcontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to display components that enable viewing objects in an associated application on local machines.

### Configuration File

webcomponent\config\library\view\viewcontainer\_component.xml

### Parent Definition

contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml

### Scope

---

type	dm_sysobject
version	5.2.5

---

## Parameters

component	(Required) Supplied by the action execution class). The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent class.) Lst of arguments and values to pass to each contained component.
launchViewer	(Optional) Boolean: Set to true to launch a viewing application.

## Elements

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default:5.2.5
<ucfrequired>	If this element is present, then the component requires UCF to be installed.

# viewcontainer (type dm\_sysobject)

**Purpose** This is the WDK 5.3 viewcontainer component. It is the container used to display components that enable viewing objects in an associated application on a local machine.

## Configuration File

webcomponent\config\library\contenttransfer\view\viewcontainer\_component.xml

## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

---

type	dm_sysobject
------	--------------

---

## Caller

This component is called by the action [view \(type dm\\_sysobject\), page 465](#) and displays the component [view \(type dm\\_sysobject\), page 1104](#).

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
<control-propagation>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.

---

<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;ucfrequired&gt;</code>	If this element is present, then the component requires UCF to be installed.

---

## viewcontainer (type dmr\_content, version 5.2.5)

**Purpose** This is the WDK 5.2.5 viewcontainer component. It cannot be addressed directly by URL or ID, because it is not the current version of this component. It is provided for backward compatibility. It is the WDK 5.2.5 component used to display components that enable viewing objects in an associated application on local machines.

### Configuration File

webcomponent\config\library\view\viewcontainer\_component.xml

### Parent Definition

contentxfercontainer:webcomponent/config/library/contentxfer/contentxfercontainer\_component.xml

### Scope

---

type	dmr_content
version	5.2.5

---

## Caller

Cannot be called directly. Must be extended.

## Parameters

---

component	(Required) Supplied by the action execution class). The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent class.) List of arguments and values to pass to each contained component.
launchViewer	(Optional) Boolean: Set to true to launch a viewing application.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: 5.2.5
---------------------------	---

---

# viewcontainer (type dmr\_content)

**Purpose** This is a WDK 5.3 viewcontainer component. It is a container used to display components that enable viewing objects from repositories.

## Configuration File

webcomponent\config\library\contenttransfer\view\viewcontainer\_component\_dmr.xml



## Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

## Scope

---

type	dmr_content
------	-------------

---

## Caller

This component is called by the action [view \(type dm\\_sysobject\), page 465](#) and displays the component [view \(type dm\\_sysobject\), page 1104](#).

## Parameters

These parameters are inherited from combocontainer:wdk/config/combocontainer\_component.xml.

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## Elements

---

<bindingcomponentversion>	Specifies the version of components that are supported by the container component. Default: latest
<ucfrequired>	If this element is present, then the component requires UCF to be installed.

---

## viewexternalresult

**Purpose** This is an internal component that is invoked by the view action to view the results of a search on external sources. Failover is not enabled for this component.

### Configuration File

webcomponent\config\library\search\searchex\viewexternalresult\_component.xml

### Scope

All.

### Caller

The component is called by the view action for objects of type dm\_externalresult.

### Parameters

---

entryId	(Required) Internal key to the query results.
queryId	(Required) Internally used to identify the current search context

---

## viewxforms

**Purpose** Extends editxforms component to display forms. Contains the [xforms, page 1159](#) component. If the user does not have the form\_user role, the form will be displayed as XML.

## Configuration File

webcomponent\config\xforms\viewxforms\_component.xml

## Parent Definition

editxforms:webcomponent/config/xforms/editxforms\_component.xml

## Scope

All.

## Caller

This component is called by the actions [view](#) (type `dm_sysobject`), [page 465](#) and [downloadcontent](#) (type `dm_sysobject`), [page 281](#) when the selected object is a form.

## Parameters

objectId	(Required) ID of the selected form
templateId	(Optional) ID of the associated form template

## Elements

<readOnly>	Set to true to display forms as readonly
------------	--

# virtuallinkconnect

## Purpose

Supports virtual links (URL to a single document) by providing authentication and a list of root paths that are used to resolve the path in the URL. A virtual link has the syntax `http://host[/docbase:path]/document`.

## Configuration File

wdk\config\virtuallinkconnect\_component.xml

## Scope

All.

## Parameters

---

redirectUrl	(Required) URL to be displayed when the object cannot be located in the repository.
virtualLinkPath	(Required) URL to the feature that provides anonymous access, which allows the use of predefined login credentials (per repository) instead of requiring a user to log in using their credentials.

---

## Elements

---

<rootpaths>	Contains one or more <rootpath> elements, which can optionally be enclosed within a <filter> element
<filter>	Optionally wraps a <rootpath> element to make the root path available based on the value of the qualifier. For example, you could make a root path available only to a certain role or repository: <filter role='administrator'>
<rootpath>	The value is an absolute repository path
<defaultaccounts>	Contains anonymous access login information in one or more <defaultaccount> elements
<filter docbase='docbasename'>	Optional filter to specify the account for a single repository. Set the repository attribute on this element to the repository name.
<defaultaccount>	Sets up anonymous access for virtual links.
<username>	Name of a user in the target repository

---

<password>	Password for the anonymous user. Can be encrypted with the Trusted Authenticator Tool. Refer to <i>Web Development Kit and Client Applications Development Guide</i> for instructions.
<domain>	Optional network domain name for login to the repository

---

## Usage

This component is part of the virtual link service. It has no UI. The component is called by the browser after the virtual link servlet determines that the URL may contain a virtual link. The component reads the configuration information from the component definition file and provides authentication details to the virtual link handler based on the user's current session. If the user does not have a session, the virtual link service presents a login page or is automatically logged in using anonymous access.

See *Web Development Kit and Client Applications Development Guide* for more details about the virtual link service.

# visiblerepository\_preferences

## Purpose

Enables the user to select repositories that will be visible in the login dialog. The repositories that are displayed for selection are those that are accessible to the connection brokers in the user's preferred connection broker list. The user can select or enter new connection brokers by clicking the **Configure connection broker** link.

## Configuration File

webcomponent\config\environment\preferences\visiblerepository\visiblerepository\_preferences\_component.xml

## Scope

All.

# webwfm

**Purpose** Launches the Web workflow manager applet. This component is launched by the following actions: new, view, editfile, and delete on a single object of type dm\_process (template) objects, and the view action on a single dm\_workflow (template) object. The container extends [combocontainer](#), page 644.

## Configuration File

webcomponent\config\library\workflow\webwfm\webwfm\_component.xml

## Scope

All.

## Caller

This component is called by the actions [delete \(type dm\\_process\)](#), page 268, [editfile \(type dm\\_process\)](#), page 290, and [view \(type dm\\_process\)](#), page 460 when invoked on a dm\_process object (workflow template). It is called by the action [newprocess \(type dm\\_folder\)](#), page 346 to create a new workflow template. It is displayed in the component [webwfmcontainer](#), page 1120, which allows editing of a workflow template.

## Parameters

---

objectId	(Required) ID of a process, a workflow, or a folder or cabinet in which to create the workflow.
----------	---

---

## Elements

---

<helppath>	Subdirectory of the help folder that contains online help for the Web workflow manager. For example, a path of webwfmhelp could be resolved to C:\Program Files\Apache Group\Tomcat 4.1\webapps\webtop\help\en\webwfmhelp and a similar path for each installed language pack (replacing "en")
<maxidletime>	Specifies how often (minutes) the Web workflow manager sends event to the server to keep the session alive.
<map>	Contains elements that configure the report map view
<height>	Default height in pixels for report map view for applets that cannot resize with the browser, for example, Netscape

---

## Usage

The Web workflow manager applet requires the Sun virtual machine in the client browser. The user is prompted to install the required VM when the workflow manager is called.

The Web workflow manager is called for a single object. If a workflow or template is edited, it must be checked in through the Web workflow manager UI. Checkin for workflows and templates is disabled in the Webtop UI.

# webwfm\_checkjvm

**Purpose** This component is a container for the web workflow manager component that also checks the Java Virtual Machine to ensure that it is compatible.

## Configuration File

webcomponent\config\library\workflow\webwfm\webwfm\_component.xml

### Parent Definition

webwfm:webcomponent/config/library/workflow/webwfm/webwfm\_component.xml

### Scope

All.

### Parameters

These parameters are inherited from webwfm:webcomponent/config/library/workflow/webwfm/webwfm\_component.xml.

---

objectId	(Required) ID of a process, a workflow, or a folder or cabinet in which to create the workflow.
----------	---

---

## webwfmcontainer

**Purpose** Container component for the Workflow Manager.

### Configuration File

webcomponent\config\library\workflow\webwfm\webwfmcontainer\_component.xml

### Parent Definition

combocontainer:wdk/config/combocontainer\_component.xml

### Scope

All.



## Caller

This component is called by the action [view](#) (type `dm_workflow`), [page 467](#) and displays the component [webwfm](#), [page 1118](#).

## Parameters

These parameters are inherited from `combocontainer:wdk/config/combocontainer_component.xml`.

<code>component</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) The name of the component to contain.
<code>componentArgs</code>	(Required) Supplied by the <code>LaunchComponent</code> action execution class.) List of arguments and values to pass to each contained component.

## Elements

<code>&lt;control-propagation&gt;</code>	(Optional) Contains the property elements. Enables propagating the enabled or visible control property to all instances of a contained component. For example, if you add this element and the property elements to your custom checkin component definition, then the enabled and visible state of all controls are propagated to the JSP shown for each object being checked in.
<code>&lt;control-propagation&gt;.&lt;property name="enabled"/&gt;</code>	(Optional) Propagates the enabled property on all controls on a JSP from one instance of the contained component to the next.
<code>&lt;control-propagation&gt;.&lt;property name="visible"/&gt;</code>	(Optional) Propagates the visible property on all controls on a JSP from one instance of the contained component to the next.

# webwfmeditcontainer

**Purpose** This component calls the component [webwfm](#), [page 1118](#) to launch the Web workflow manager applet in editing mode. This container extends [combocontainer](#), [page 644](#).

## Configuration File

```
webcomponent\config\library\workflow\webwfm\webwfmeditcontainer_
component.xml
```

## Parent Definition

```
webwfmcontainer:webcomponent/config/library/workflow/webwfm/webwfmcontainer_
component.xml
```

## Scope

All.

## Caller

This component is called by the action [editfile](#) (type [dm\\_process](#)), [page 290](#) for a [dm\\_process](#) (template) object, and displays the component [webwfm](#), [page 1118](#).

## Parameters

These parameters are inherited from [combocontainer:wdk/config/combocontainer\\_](#)  
[component.xml](#).

---

component	(Required) Supplied by the LaunchComponent action execution class.) The name of the component to contain.
componentArgs	(Required) Supplied by the LaunchComponent action execution class.) List of arguments and values to pass to each contained component.

---

## wfgrouponlylocator

**Purpose** Extends [userorgrouplocator](#), page 1079 and selects groups in workflow operations. The container component extends [locatorcontainer](#), page 839.

### Configuration File

webcomponent\config\library\workflow\wfgrouponlylocator\_component.xml

### Parent Definition

userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml

### Scope

All.

### Parameters

These parameters are inherited from userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml.

---

arrangeselect	(Optional) Not used.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Type of the selected object

---

## Elements

---

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...

<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)

<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute <code>dqlformatted='false'</code> to quote and escape a value. Use the attribute <code>casesensitive='true'</code> to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## wfgrouponlylocatorcontainer

**Purpose** This component is the container for the component [wfgrouponlylocator](#), page 1123, which is used to select groups in workflow operations.

### Configuration File

`webcomponent\config\library\workflow\wfgrouponlylocator_component.xml`

### Parent Definition

`locatorcontainer:webcomponent/config/library/locator/locatorcontainer_component.xml`

### Scope

All.

## Parameters

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple selection.
repository	(Optional) ID of the repository in which to search

# wfrecentuseronlylocator

**Purpose** This component is used to search for users who were recently selected or changed. It is used in conjunction with workflow operations.

## Configuration File

webcomponent\config\library\workflow\wfuseronlylocatorcontainer\_component.xml

## Parent Definition

recentuseronlylocator:webcomponent/config/library/locator/useronlylocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from recentuserorgrouplocator:webcomponent/config/library/locator/recentuserorgrouplocator\_component.xml.

---

multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type of the items to display in the list.

---

## wfsysobjectlocatorcontainer

**Purpose** Contains [sysobjectlocator](#), page 1029 and [recentsysobjectlocator](#), page 926.

### Configuration File

webcomponent\config\library\workflow\wfsysobjectlocatorcontainer\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

### Scope

All.

### Caller

This component is called by the action [addwfattachment](#), page 223.

### Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.



---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to search
multiselect	(Optional) Boolean. Set to true to allow multiple selections.
objecttype	(Optional) Type of object selected
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

## Elements

---

<multidocbasesupport>	Set to true to enable locator for multiple repositories
-----------------------	---

---

# wftemplatefolderlocator

**Purpose** Locates folders containing dm\_process objects (workflow templates) in a repository.

## Configuration File

webcomponent\config\library\locator\wftemplatelocators\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Caller

This component is called by the component [wftemplatelocatorcontainer](#), page 1132.

## Parameters

These parameters are inherited from `sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator_component.xml`.

---

<code>contentType</code>	(Optional) Content type for filetype filter
<code>flatlist</code>	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
<code>folderId</code>	(Optional) ID of the folder in which to search
<code>multiselect</code>	(Optional) Boolean: Set to true to support multiple object selection.
<code>objecttype</code>	(Optional) Type of the selected object

---

## Elements

For information on configurable columns, see [doclist](#), page 684.

---

<code>&lt;objecttype&gt;</code>	Base type to be located
<code>&lt;containerselectable&gt;</code>	Set to true to specify that the container type is selectable when the container is a subtype of the <code>objecttype</code>
<code>&lt;flatlist&gt;</code>	Set to true to display a flat list of all selectable objects
<code>&lt;views&gt;</code>	Contains <code>&lt;view&gt;</code> elements
<code>&lt;view&gt;</code>	View element configures root (cabinet view), container, and flatlist views. The <code>applyto</code> attribute must specify one or more of the views in a comma-separated list, for example, <code>applyto='root,container,flatlist'</code> .

<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter

<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## wftemplatelocatorcontainer

**Purpose** Contains workflow template locators: [allwftemplatelocator](#), page 540, [wftemplatesubscriptionlocator](#), page 1133, [mywftemplatelocator](#), page 860, [wftemplatefolderlocator](#), page 1129, and [recentwftemplatelocator](#), page 931.

### Configuration File

webcomponent\config\library\locator\wftemplatelocators\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

### Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of the folder in which to begin the search.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

## wftemplatesubscriptionlocator

**Purpose** Locates and displays workflow templates to which the user has subscribed.

### Configuration File

webcomponent\config\library\locator\wftemplatelocators\_component.xml

### Parent Definition

subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml

### Scope

All.

## Caller

This component is called by the component [wftemplatelocatorcontainer](#), page 1132.

## Parameters

These parameters are inherited from subscriptionlocator:webcomponent/config/library/locator/subscriptionlocator\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

---

## Elements

For information on configurable columns, see [doclist](#), page 684.

---

<objecttype>	Base type to be located
<containerselectable>	Set to true to specify that the container type is selectable when the container is a subtype of the objecttype
<flatlist>	Set to true to display a flat list of all selectable objects
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.

<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)

<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

## wfuseronlylocator

**Purpose** Contains workflow user locators [useronlylocator](#), page 1075 and [recentuseronlylocator](#), page 927.

### Configuration File

webcomponent\config\library\workflow\wfuseronlylocatorcontainer\_component.xml

### Parent Definition

useronlylocator:webcomponent/config/library/locator/useronlylocator\_component.xml

### Scope

All.



## Parameters

These parameters are inherited from userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml.

arrangeselect	(Optional) Boolean: Set to true to allow selection of users in a sequential order.
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## Elements

<columns>	Contains <column> elements to display or hide attributes
<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'>. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.
<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements

<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. If a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.

<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<columns>	Contains <column> elements to display or hide attributes
<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.

---

## wfuseronlylocatorcontainer

**Purpose** This component allows the selection of users for workflow operations. It displays the component [wfuseronlylocator](#), page 1136.

### Configuration File

webcomponent\config\library\workflow\wfuseronlylocatorcontainer\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

### Scope

All.

### Parameters

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) ID of the selected object
repository	(Optional) ID of the repository in which to search

---

## wfuserorgroupfromgrouplocator

**Purpose** Locates users who are members of a specific group. It is used in workflow operations.

## Configuration File

webcomponent\config\library\workflow\wfuserorgroupfromgrouplocator\_  
component.xml

## Parent Definition

userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_  
component.xml

## Scope

All.

## Parameters

groupName	(Optional) Name of group from which to locate users and groups
multiselect	(Optional) Set to true to support multiple selection.
objecttype	(Optional) Type of object to locate

## Elements

For information on configurable columns, see [doclist, page 684](#).

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.

<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter

<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)

---

# wfuserorgroupfromgrouplocatorcontainer

**Purpose** Contains the userorgrouplocator component used in the sendtodistributionlist component.

## Configuration File

webcomponent\config\library\workflow\wfuserorgroupfromgrouplocator\_component.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Parameters

---

arrangeselect	(Optional) True to allow selection of workflow performers in a sequential order.
groupName	(Optional) Name of the group of which the listed users/groups are to be members
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object
repository	(Optional) Repository in which to search

---

# wfuserorgrouplocator

**Purpose** This component is used to locate active users or groups in a repository. The component allows a user navigate from usergroups to find a user or group.

## Configuration File

```
webcomponent\config\library\workflow\wfuserorgrouplocatorcontainer_component.xml
```

## Parent Definition

```
userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator_component.xml
```

## Scope

All.

## Parameters

These parameters are inherited from userorgrouplocator:webcomponent/config/library/locator/userorgrouplocator\_component.xml.



---

arrangeselect	(Optional) Boolean: Set to true to allow selection of users in a sequential order.
flatlist	(Optional) Boolean: Set to true to list objects in a flat list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) ID of the selected object

---

## Elements

For information on configurable columns, see [doclist, page 684](#).

---

<objecttype>	Base type to be located
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...

<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.
<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes
<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)

---

<value>	Use the attribute <code>dqlformatted='false'</code> to quote and escape a value. Use the attribute <code>casesensitive='true'</code> to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<columns>	Contains <column> elements to display or hide attributes
<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'>. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.

---

## wfuserorgrouplocatorcontainer

**Purpose** Container for simple user-or-group locator used in sendtodistributionlist.

### Configuration File

webcomponent\config\library\workflow\wfuserorgrouplocatorcontainer\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

**Scope**

All.

**Parameters**

---

arrangeselect	(Optional) Boolean: Set to true to allow selection of users in a sequential order.
flatlist	(Optional) Boolean: Set to true to list objects in a flat list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) The type of the selected object
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) IDs of objects that are selected when the component is displayed

---

## wizardcontainer

**Purpose** Displays a component within a wizard framework, where each layout page (JSP) is presented as a different page in the wizard. Provides navigation buttons.

**Configuration File**

wdk\config\wizardcontainer\_component.xml

**Parent Definition**

dialogcontainer:wdk/config/dialogcontainer\_component.xml

**Scope**

All.

## Parameters

These parameters are inherited from container:wdk/config/container\_component.xml.

component	(Required) The contained component. Set the requiresVisit attribute to true to require the component to be visited before an OK button is displayed.
-----------	--

## workflowavailability

**Purpose** Launched by the action [workflowavailability, page 477](#) in [dialogcontainer, page 669](#), this component sets the user's availability to receive workflows.

### Configuration File

webcomponent\config\library\workflow\availability\workflowavailability\_component.xml

### Scope

All.

### Caller

This component is called by the action [workflowavailability, page 477](#) and displayed in the component [dialogcontainer, page 669](#).

## workflowstatusdrilldown

**Purpose** Lists workflow objects and their statuses, using a streamline layout.

## Configuration File

webcomponent\config\library\workflow\status\workflowstatusdrilldown\_  
component.xml

## Scope

All.

## Caller

This component is called by the action [workflowstatusdrilldown](#), page 478.

# workflowstatuslist

## Purpose

Lists workflow objects and their statuses, using a list layout.

## Configuration File

webcomponent\config\library\workflow\status\workflowstatuslist\_component.xml

## Scope

All.

## Caller

This component is called by the action [workflowstatuslist](#), page 478.

## Parameters

mode	(Optional) (Required) Specifies the workflows to be displayed. Valid values: all (shows all workflows)   supervisor (shows workflows supervised by the current user)   creator (shows workflows created by the current user)   supervisorcreator (shows workflows supervised or created by the current user)
------	--

# workqueuecategorylocatorcontainer

**Purpose** Container component for locating work queue categories.

## Configuration File

webcomponent\config\library\queuemanagement\workqueuecategorylocators\_component.xml

## Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

## Scope

All.

## Caller

This component is called by the action [myworkqueuecategories](#), page 339.

## Parameters

---

arrangeselect	(Optional) Not used.
multiselect	(Optional) Boolean: Set to true to support multiple selection.
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

# workqueuecategoryproperties

**Purpose** Sets the properties for a work queue category.

## Configuration File

webcomponent\config\library\queuemanagement\workqueuecategoryproperties\_component.xml

## Parent Definition

adminpropertycontainer:webcomponent/config/admin/container/adminpropertycontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [attributes](#) (type `dmc_workqueue_category`), [page 236](#), [newWorkQueueCategory](#), [page 351](#), and [properties](#) (type `dmc_workqueue_category`), [page 370](#), and displays the component [categoryattributes](#), [page 605](#).



---

## Parameters

---

objectId	(Required) ID of the work queue category object.
objectname	(Required) Name of the work queue category.

---

# workqueuedocprofileproperties

**Purpose** Sets the properties of a work queue document profile.

## Configuration File

webcomponent\config\library\queuemanagement\workqueuedocprofileproperties\_component.xml

## Parent Definition

adminpropertycontainer:webcomponent/config/admin/container/adminpropertycontainer\_component.xml

## Scope

All.

## Caller

This component is called by the actions [attributes](#) (type `dmc_workqueue_doc_profile`), page 237, [newWorkQueueDocProfile](#) (type `dmc_workqueue_doc_profile`), page 352, and [properties](#) (type `dmc_workqueue_doc_profile`), page 371, and displays the component [docprofileattributes](#), page 687.

## Parameters

---

objectId	(Required) ID of the work queue document profile object.
objectname	(Required) Name of the work queue document profile.

---

# workqueuelocator

**Purpose** Lists the available work queues.

## Configuration File

webcomponent\config\library\queuemanagement\workqueuelocator\_component.xml

## Parent Definition

persistentobjectlocator:webcomponent/config/library/locator/persistentobjectlocator\_component.xml

## Scope

All.

## Caller

This component is called by the component [workqueuecategorylocatorcontainer](#), page 1151.

## Parameters

These parameters are inherited from persistentobjectlocator:webcomponent/config/library/locator/persistentobjectlocator\_component.xml.

---

flatlist	(Optional) Not used.
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type of the objects to include in the list.

---

## Elements

This component has configurable columns. (See [doclist](#), page 684 for a description.)

---

<objecttype>	Sets the user preference for the object type to be displayed. The default is dm_group.
<iconwell_icon>	Path to the file containing the icons for display
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<queryfilter>.<containertypes>	Comma-separated list of navigable object types, such as dm_cabinet, dm_folder.

<queryfilter>.<includetypes>	Optional element (cannot be more than one instance of this element) that specifies a comma-separated list of Documentum types to be included in the view. The container must be listed along with the subtypes within the container, for example, dm_folder, dm_document
<queryfilter>.<excludetypes>	Optional element (cannot be more than one instance of this element) that specifies the type of object to be excluded from the view. if a container type is not one of the subtypes listed in this tag, the containers are not listed. If a container type is listed, the objects of the container type will be hidden, for example, dm_folder, dm_document hides all folders or documents.

---

## workqueuelocatorcontainer

**Purpose** Container component for locating available work queues.

### Configuration File

webcomponent\config\library\queuemanagement\workqueuelocator\_component.xml

### Parent Definition

locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml

### Scope

All.

## Parameters

These parameters are inherited from locatorcontainer:webcomponent/config/library/locator/locatorcontainer\_component.xml.

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a navigation tree.
folderId	(Optional) ID of the folder in which to begin the search.
multiselect	(Optional) Boolean: Set to true to support multiple selection.
objecttype	(Optional) Object type of the objects to include in the list.
repository	(Optional) ID of the repository in which to search
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

---

# workqueuepolicyproperties

**Purpose** Sets the properties for a work queue policy.

## Configuration File

webcomponent\config\library\queuemanagement\workqueuepolicyproperties\_component.xml

## Parent Definition

adminpropertycontainer:webcomponent/config/admin/container/adminpropertycontainer\_component.xml

**Scope**

All.

**Caller**

This component is called by the actions [attributes \(type dmc\\_workqueue\\_policy\)](#), page 237, [newWorkQueuePolicy \(type dmc\\_workqueue\\_policy\)](#), page 353, and [properties \(type dmc\\_workqueue\\_policy\)](#), page 372, and displays the component [policyattributes](#), page 893.

**Parameters**

---

objectId	(Required) ID of the work queue policy object.
objectname	(Required) Name of the work queue policy.

---

## workqueueproperties

**Purpose** Sets the properties for a work queue.

**Configuration File**

webcomponent\config\library\queuemanagement\workqueueproperties\_component.xml

**Parent Definition**

adminpropertycontainer:webcomponent/config/admin/container/adminpropertycontainer\_component.xml

**Scope**

All.

## Caller

This component is called by the actions [attributes](#) (type `dmc_workqueue`), page 238, [newWorkQueue](#) (type `dmc_workqueue`), page 350, and [properties](#) (type `dmc_workqueue`), page 373, and displays the component [queueattributes](#), page 918.

## Parameters

---

<code>objectId</code>	(Required) ID of the work queue object.
<code>objectname</code>	(Required) Name of the work queue.

---

# xforms

## Purpose

Displays a form for viewing or editing a selected form object. The user must have the role `form_user`. The component is launched within the `editxforms` container or the `viewxforms` container by the `edit` or `view` actions, respectively. The component passes parameter values to the `xforms` control, which renders the form.

## Configuration File

`webcomponent\config\xforms\xforms_component.xml`

## Scope

All.

## Parameters

---

<code>objectId</code>	(Required) Object ID of the form.
<code>readOnly</code>	(Optional) Specifies whether the form should be shown in edit or view mode. In edit mode, WDK buttons are not hidden (form is modal). In view mode, form buttons are disabled, and WDK buttons are displayed (Close and Help).

---

# xformsattachmentimportlocatorcontainer

**Purpose** This component is a locator container that contains a locator tab and an import tab. The user is able to use the import tab to initiate an import action. After the import is completed, the corresponding imported object will be included in the selected list.

## Configuration File

```
webcomponent\config\xforms\locator\xformsattachmentimportlocatorcontainer_
component.xml
```

## Parent Definition

```
xformsattachmentlocatorcontainer:webcomponent/config/xforms/locator/
xformsattachmentlocatorcontainer_component.xml
```

## Scope

All.

## Parameters

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
importlocation	(Required) ID of the repository folder to which the document will be imported
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Required) Type of the selected object
restrictformats	(Optional) Specify the formats the locator will display
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected



---

startinlocation	(Optional) ID of the folder where the locator should begin the search
whereclause	(Optional) Specifies the where clause that will be used to locate the selected objects

---

## xformsattachmentlocator

**Purpose** This component is a locator specialized for locating document attachments based on their type and required version label(s).

### Configuration File

webcomponent\config\xforms\locator\xformsattachmentlocator\_component.xml

### Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

### Scope

All.

### Parameters

---

objecttype	(Optional) Type of the selected object
------------	--

---

### Elements

---

<iconwell_icon>	Path to the file containing the icons for display
<objecttype>	Base type to be located

<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<privatecabinetvisible>	Set to true to make private cabinets visible
<allversionsvisible>	Set to true to display all versions

## xformsattachmentlocator (type rm\_classification\_guide)

**Purpose** This component is a locator specialized for locating document attachments based on the rm\_classification\_guide type.

### Configuration File

webcomponent\config\library\records\locator\classification\_guide\_locator\_component.xml

### Parent Definition

xformsattachmentlocator:webcomponent/config/xforms/locator/xformsattachmentlocator\_component.xml

### Scope

---

type	rm_classification_guide
------	-------------------------

---

### Parameters

---

objecttype	(Optional) Type of the selected object.
------------	---

---

## Elements

---

<iconwell_icon>	Path to the file containing the icons for display.
<objecttype>	Base type to be located.
<containerselectable>	Boolean: Set to true to specify that the container type is selectable when the container is a subtype of the objecttype.
<privatecabinetvisible>	Set to true to make private cabinets visible.
<allversionsvisible>	Set to true to display all versions.
<flatlist>	Boolean: Set to true to display a flat list of all selectable objects.
<views>	Contains <view> elements
<view>	View element configures root (cabinet view), container, and flatlist views. The applyto attribute must specify one or more of the views in a comma-separated list, for example, applyto='root,container,flatlist'.
<queryfiltersets>	Contains <queryfilterset> elements that present a drop-down list that will be visible if there is more than one filter (<queryfilterset>) defined.
<queryfilterset>	Contains a set of queries contained in <queryfilter> elements that filter the selection list, for example, folders only. Defines one dropdown item. Each view can contain up to three filter sets.
<queryfilter>	Each filter contains a DQL query: select <includetypes> from <containertypes> not <excludetypes> where <attribute> <predicate> <value> <and> <attribute> <predicate> <value> ...
<queryfilter>.<displayname>	Specifies the name to be displayed for the queryfilter. Can contain a <nlsid> element or text string.
<attributefilters>	Cannot be more than one instance of this element. Contains <attributefilter> sets that filter objects based on their attributes

<attributefilter>	Contains <and>, <attribute>, <predicate>, and <value> to compose an attribute filter
<and>	Boolean: true to combine attribute filters, false to perform OR filter (single tag only)
<attribute>	(Required) String attribute name, for example, a_content_type (single tag only)
<predicate>	Contains a logical operation such as equals. Valid values: sw (starts with), ew (ends with), co (contains), nc (not contains), eq (equal), ne (not equal), gt (greater than), ge (greater than or equal), lt (less than), le (less than or equal)
<value>	Use the attribute dqlformatted='false' to quote and escape a value. Use the attribute casesensitive='true' to require a case-sensitive comparison (must be true for integer attributes on Content Server/DB2 environment)
<columns>	Contains <column> elements to display or hide attributes
<column>	True to display a column of attribute values. The attribute to be displayed is specified as the value of the "attribute" attribute. For example, <column attribute='owner_name'>. Valid values of this element: false   true   singleselect   multiselect. The first value hides the column, the last three make the column visible. Set to singleselect to allow only one selection from the located components, multiselect to allow multiple selection.

---

# xformsattachmentlocatorcontainer

**Purpose** This component displays the component [xformsattachmentlocator](#), page 1161, which is used to locate attachments in the repository based on type and required version labels.

## Configuration File

```
webcomponent\config\xforms\locator\xformsattachmentlocatorcontainer_component.xml
```

## Parent Definition

```
locatorcontainer:webcomponent/config/library/locator/locatorcontainer_component.xml
```

## Scope

All.

## Parameters

---

flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Required) Type of the selected object
restrictformats	(Optional) Specify the formats the locator will display
selectedobjectids	(Optional) Comma-delimited or semicolon-delimited list of object IDs that are initially selected

startinlocation	(Optional) ID of folder in which to start the search
whereclause	(Optional) Specifies the where clause that will be used to locate the selected objects

---

## Elements

---

<multidocbasesupport>	Set to true to enable locator for multiple repositories
-----------------------	---

---

# xformsimportlocator

**Purpose** A dummy locator to be shown on an attachment locator container.

## Configuration File

webcomponent\config\xforms\locator\xformsimportlocator\_component.xml

## Parent Definition

sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml

## Scope

All.

## Parameters

These parameters are inherited from sysobjectlocator:webcomponent/config/library/locator/sysobjectlocator\_component.xml.

contentType	(Optional) Content type for filetype filter
flatlist	(Optional) Boolean: Set to true to list all objects in a flat list, set to false to display objects in a drop-down list.
folderId	(Optional) ID of folder in which to begin the search
multiselect	(Optional) Boolean: Set to true to support multiple object selection.
objecttype	(Optional) Type of the selected object

## xformsproperties

**Purpose** This component displays the properties panel for an xform object.

### Configuration File

webcomponent\config\xforms\xformsproperties\_component.xml

### Parent Definition

xforms:webcomponent/config/xforms/xforms\_component.xml

### Scope

All.

### Parameters

These parameters are inherited from xforms:webcomponent/config/xforms/xforms\_component.xml.

---

objectId	(Required) ID of the xform for which the properties are displayed
readOnly	(Optional) Boolean: Set to true to display the properties in read-only mode.

---



## A

- absolutefilepathvalidator control, 20
- acclassvalueformatter control, 20
- action (control attribute), 14
- actionbutton control, 21
- actionbuttonlist control, 22
- actionimage control, 22
- actionlink control, 23
- actionlinklist control, 24
- actionmenuitem control, 26
- actionmultiselect control, 26
- actionmultiselectcheckall control, 27
- actionmultiselectcheckbox control, 28
- actions
  - common attributes, 14
  - control attributes, 14
- adminrequiredfieldvalidator control, 28
- applet attributes
  - align, 16
  - archive, 16
  - code, 16
  - codebase, 16
  - height, 16
  - id, 16
- appletattributes
  - bufferize, 17
  - checkoutlocationunix, 18
  - checkoutlocationwindows, 18
  - contentlocationunix, 17
  - contentlocationwindows, 17
  - contentticket, 18
  - contentxferlisteners, 18
  - debug, 17
  - enablesubdircreation, 18
  - enableuniquedircreation, 18
  - encoderesult, 17
  - errorhandler, 16
  - errorhandlerarg, 16
  - executionmgrclassname, 17
  - expand, 18

- idstoretrieve, 18
- linkinstructions, 18
- locale, 17
- progressmonitorclassname, 18
- registrylocationunix, 17
- serverversion, 17
- serviceurl, 17
- successhandler, 17
- successhandlerarg, 17
- argument control, 29
- argument control (Documentum-enabled), 30
- attributeselector control, 30

## B

- bannerbox control, 32
- body control, 33
- bookmarklink control, 34
- booleanformatter control, 35
- breadcrumb control, 35
- <breadcrumbiconwellmode>, 115
- browerrequirements control, 37
- button control, 37

## C

- celllist control, 39
- celltemplate control, 40
- checkbox control, 41
- checkjvmapplet control, 42
- clientenvpanel control, 43
- colorpicker control, 46
- columnlist control, 44
- columnpanel control, 47
- combocontainerpanel control, 47
- commentimage control, 48
- commentrow control, 48
- comparevalidator control, 50
- componentinclude control, 51
- componenturl control, 52

- containerinclude control, 52
- controls
  - event attributes, 13
  - set focus, 12
  - WDK, 11
- cssclass attribute, 12

## D

- datadropdownlist control, 53
- datagrid control, 54
- datagridrow control, 55
- datalistbox control, 57
- dataoptionlist control, 58
- datapagesize control, 58
- datapaging control, 59
- datasortimage control, 60
- datasortlink control, 61
- dateinput control, 63
- datetime control, 65
- datetimevalidator control, 68
- datevalidator control, 70
- datevalueformatter control, 69
- disabledclass, 14
- disabledstyle, 14
- div control, 70
- docbaseattribute control, 71
- docbaseattributedelegate control, 73
- docbaseattributelabel control, 75
- docbaseattributelist control, 76
- docbaseattributeseparator control, 79
- docbaseattributevalidator control, 79
- docbaseattributevalue control, 80
- docbaseattributevalueassistance control, 81
- docbaseattributevalueassistancevalue control, 82
- docbasefoldertree control, 85
- docbaseicon control, 84
- docbaselockicon control, 88
- docbaseobject control, 88
- docbaseobjectvalidator control, 91
- docbasepriorityicon control, 92
- docbaseselector control, 93
- docformatvalueformatter control, 93
- docsizevalueformatter control, 94
- DONTSHOWAGAIN, 905
- dragdrop control, 94
- dragdropregion control, 95
- dropdownlist control, 97

- duplicatenamevalidator control, 98
- dynamic attribute, on action control, 14

## E

- enabled, 12
- erasevalueimage control, 99
- events
  - control, 13
- extpermvalueformatter control, 99

## F

- filebrowse control, 100
- fileselectorapplet control, 101
- filters
  - dynamic (with LaunchComponent), 9
- fireclientevent control, 102
- firepresubmitclientevent control, 103
- focus attribute, 12
- folderexclusionformatter control, 104
- folderselectionvalidator control, 104
- form control, 105
- formatmessageparams control, 106
- forminclude control, 107
- formurl control, 107
- fragment control, 108
- frame control, 110
- frameset control, 109

## G

- governedicon control, 110

## H

- head control, 111
- helpimage control, 112
- hidden control, 112
- html control, 113
- htmlsafetextformatter control, 113

## I

- iconwell control, 114
- id attribute
  - controls, 12
- ILaunchComponentEvaluator, 9
- image control, 116
- inlinewebform control, 117
- inputmaskvalidator control, 118

**L**

label attribute  
     controls, 12  
 label control, 119  
 LaunchComponent  
     dynamic filters, 9  
 link control, 120  
 listbox control, 121

**M**

macclientdetect control, 122  
 menu control, 122  
 menugroup control, 123  
 menugroupactionmap control, 124  
 menugroupconfig control, 125  
 menuitem control, 125  
 menuseparator control, 126  
 multidocbasetree control, 126  
 multilinestringlengthformatter  
     control, 129  
 multiselector control, 130  
 multivaluesinputmaskvalidator  
     control, 132  
 multivaluesrequirefieldvalidator  
     control, 133

**N**

name attribute  
     controls, 12  
 nlsid attribute  
     controls, 12  
 nodatarow control, 134  
 numberformatter control, 134

**O**

<olecompound>, 248  
 onchange event, 13  
 onclick event, 13  
 oncomplete, 14  
 onselect event, 13  
 open control, 135  
 option control, 136

**P**

pane control, 137  
 panel control, 137  
 paneset control, 139

password control, 140  
 permissionvalueformatter control, 141  
 policystatenameformatter control, 142  
 postserverevent control, 142  
 preference control, 144  
 preferencedisplaygroups control, 145  
 preferencscope control, 145  
 primaryfolderpathlink control, 147  
 privilegevalidator control, 147  
 progressbar control, 148  
 progressbarupdater control, 149

**Q**

quotevalidator control, 149

**R**

radio control, 150  
 rangevalidator control, 151  
 rankvalueformatter control, 152  
 regexpvalidator control, 153  
 repositorysearch control, 153  
 requiredfieldvalidator control, 154  
 richtextdisplay control, 157  
 richtextdocbaseattribute control, 155  
 richtextdocbaseattributelabel control, 156  
 richtextdocbaseattributevalue control, 157  
 richtexteditor control, 159  
 richtextpanel control, 162  
 roombanner control, 162  
 row control, 163  
 run time  
     attribute values resolved, 19  
 runatclient attribute, 13

**S**

searchattribute control, 163  
 searchattributegroup control, 165  
 searchdateattribute control, 166  
 searchfulltext control, 167  
 searchlocation control, 168  
 searchobjecttypedropdownlist  
     control, 168  
 searchscopecheckbox control, 169  
 searchsizeattribute control, 170  
 serviceprogressfeedback control, 171  
 showifinvalid, 14  
 showtopicdocbaseattribute control, 172

- showtopicdocbaseattributelabel control, 173
- showtopicdocbaseattributevalue control, 174
- span control, 174
- stringlengthformatter control, 175
- style attribute, 12
- symbolicversionlabelformatter control, 176
- symbolicversionlabelvalidator control, 176

## T

- tab control, 177
- tabbar control, 177
- table control, 180
- td control, 180
- termshighlightingformatter control, 180
- text control, 181
- textarea control, 182
- th control, 183
- thumbnail control, 183
- title control, 184
- tooltip attribute, 12
- tooltipdatafield attribute, 12
- tooltipnlsid attribute, 12
- topic control, 184
- topicpanel control, 185
- topicstatus control, 186
- tr control, 187
- transformparam control, 187

- tree control, 188

## U

- userstatevalueformatter control, 191
- utf8stringlengthvalidator control, 191

## V

- validationsummary control, 192
- varequiredfieldvalidator control, 192
- vdmbindingruleformatter control, 193
- vdmtreegrid control, 193
- vdmtreegridrow control, 194
- visible attribute, 12

## W

- webform control, 195
- workflowcompletetypeformatter control, 197
- workflowdisabledformatter control, 196
- workflowdurationformatter control, 197
- workfloweditorapplet control, 197
- workflowstateicon control, 198
- workqueueflagvalueformatter control, 199
- workqueuetaskstatusvalueformatter control, 199

## X

- xforms control, 200