opentext™

OpenText™ Documentum™ Content Management

Client Configuration Guide

Configure and administer client-side user interfaces, including Smart View, Classic View, and Mobile.

EDCCL250400-AGD-EN-01

OpenText™ Documentum™ Content Management
Client Configuration Guide
EDCCL250400-AGD-EN-01
Rev.: 2025-Nov-20

This documentation has been created for OpenText™ Documentum™ Content Management CE 25.4.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Preface

OpenText™ Documentum™ Content Management consists of the following main components:

- **Configuration:** – The web-based application, known as client configuration, for administrators to use to configure settings such as automated content-handling processes and background settings for OpenText Documentum CM user interfaces.
- **Smart View:** – The modern, responsive, web-based user interface that provides the ability to interact with content in repositories. See “Understanding the Smart View client” on page 237.
- **Classic View:** – The legacy user interface that remains configurable along with Smart View using client configuration.
- **Mobile:** – A mobile user interface accessible through mobile browsers or for Android and iOS as a downloadable app. For more information about the administration of mobile, see “Configuring mobile” on page 243.

When this guide refers to OpenText Documentum CM, it refers to the product as a whole, not the individual components. This guide notes when features are available for one user interface or another.

Intended audience

The information in this guide is for administrators who access and configure OpenText Documentum CM using client configuration.

Chapter 1

Understanding the Configuration Matrix and Contexts

1.1 Introduction to Client Configuration

The client configuration is the administration component of OpenText Documentum CM. You can use it to:

- Customize the user interfaces for end users.
- Configure automatic processes.
- Configure actions available in user interfaces.
- Configure interaction with external applications such as OpenText Documentum CM Administrator, rendition servers, and Retention Policy Services.

Checklists that can help you create your first OpenText Documentum CM applications and set up the Smart View client can be found in “Configuring Your First Application“ on page 449.

1.2 Understanding the Configuration Matrix

The configuration matrix is the user interface that you use to control the application of configuration elements to contexts as shown in the following screenshot:

To use the configuration matrix:

- Create a configuration component. Configuration components are the types of configurations, such as autonaming and security. Specific instances of configurations are saved as configuration elements, shown as collapsible rows in the vertical Y-axis of the configuration matrix. For example, you can create a security element for quality assurance draft writers with WRITE access. You can access the element by expanding the Security configuration component.

- Create a context, shown as the horizontal X-axis of the configuration matrix. Contexts comprise of logical groupings of content qualified by conditions and composed of properties such as document types, DQL conditions, and user groups. Contexts determine when a configuration element applies to content within the repository. For example, you can create a context that applies for all dm_document content types created for quality assurance and set to the Draft status.
- Apply the configuration element to the context. For example, applying the configuration element to the context created in the first two examples leads to a configuration such that the Draft dm_document quality assurance content provides quality assurance draft writers WRITE access.

The OpenText Documentum CM engine determines which config, that is autolinking, check-in, inheritance, and other such component configurations to use, by scanning all contexts defined in the matrix from left to right. The first context which is matched will result in the engine applying all component configurations associated to the same context from top to bottom.

Related Topics

"Organizing the Configuration Matrix" on page 16

"Configuring Contexts" on page 19

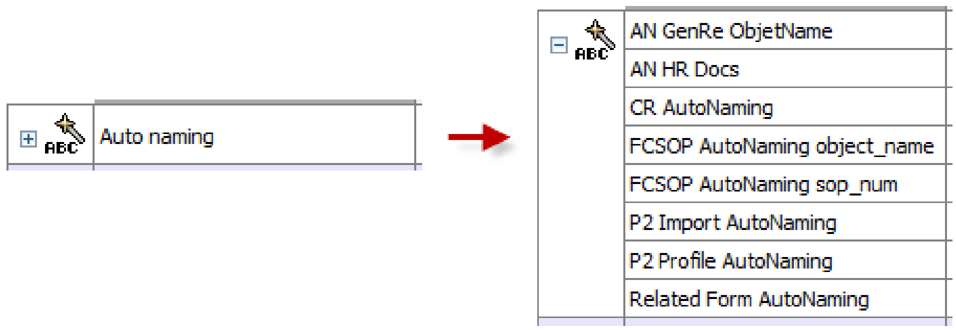
"Configuring Application Filters and Permissions" on page 20

1.3 Organizing the Configuration Matrix

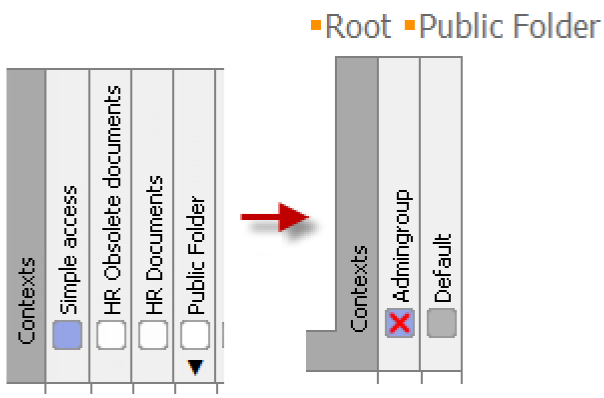
You can organize the configuration matrix in three ways:

- Organize configuration elements and contexts by assigning them to applications. You can designate a user group for an application and assign a set of permissions separate from the Security configuration component. Configuring Application Filters and Permissions contains further instructions on configuring applications.

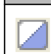
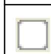
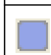
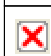
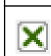
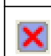
  The list box on the configuration matrix menu bar allows you to filter the configuration matrix by application.
- Expand and collapse configuration components to show or hide the configuration elements. In the following example, clicking **Expand** next to Auto naming opens the list of Auto naming configuration elements.



- Organize contexts in a tree by designating parent contexts, then expand and collapse contexts to show or hide the child contexts. In the following example, clicking the arrow next to Public Folder opens the Public Folder configuration matrix.



The following table describes the icons next to contexts:

Icon	Description
	A set with a type and user group definition
	A set with one or several type(s)
	A set with one or several user group(s)
	A set with a type not marked (in use)
	A set with a type not marked (in use) in the selected application
	A set with a user group not marked (in use)

Icon	Description
	A set with a user group not marked (in use) in the selected application
	A set with a type and user group not marked (in use)

Related Topics

"Understanding the Configuration Matrix" on page 15

1.4 Understanding Matrix Context Precedence

It is important to understand how the enabled configuration is processed in the matrix contexts (columns) from left to right.

The order of the context in the matrix is crucial because it establishes an order of precedence. When conflicts between configurations exist (for example, two or more configurations apply to a selected document or to the same user), the one associated with the left-most context in the configuration matrix takes precedence. It does this by performing DQL queries to match elements with the properties of the contexts from left to right. This precedence rule spans applications.

Best practice is to place the most specific contexts to the left and more general contexts to the right.

The "Default" context is the column at the farthest right of the matrix, and it acts as a catch all. For example, in a configuration where there are both group or type contexts, certain configurations may be bound to only those contexts so enabling them in the Default context would allow users to see additional configurations. When a user logs in, the matrix is analyzed and processed for only types involved in `dm_user/group` and not content related types. So if a Workspace or Widget config is bound to a type rather than a group it may not show the expected configuration enabled in the matrix.

Note: The best practice for objectless configurations like WidgetMapping and Folder Structure Import Mapping is to omit Type and Condition in the context.

Mapping a property page to a default context can cause the default property pages to be overridden by the matrix configuration. You can create your own property pages for users and groups. For example, you can create a user type and map a property page to that type that shows the customized property page for users.

1.5 Understanding the Matrix Evaluation Mode

In **Tools > Options** Runtime Mode section, you have the option to set a select a **Matrix Evaluation** mode:

- **Standard Evaluation**: The default mode, as described in *Understanding Matrix Context Precedence* applies to all contexts, regardless of their application designation in client configuration.
- **Application Evaluation**: Restricts the Contexts and Configurations to only those marked for a given application before determining Context Precedence and evaluation. This mode must be selected if you want to allow users to select an application at login.

  Note: If you are enabling **Application Evaluation** mode for the first time after an upgrade, you must restart the application server before settings take effect.

Note: In any given configuration where it is possible to select another configuration, the list of configurations presented shows all possible configurations, not just the configurations for a given application. For example, when you design a creation profile, you can choose to assign to it a dictionary, property page, default value template, inheritance, or lifecycle from a choice list that shows any of the configurations already created. But if the selected configuration is not assigned to the same application as the creation profile, the application evaluation will not behave as expected.

1.6 Configuring Contexts

Any change to the configuration matrix in client configuration requires the contexts cached in the App Server to be refreshed. This can be done from **Tools > Refresh** cache, for which the client URL needs to be added in the Clients URL section of **Tools > Options**.

1. Navigate to **Go to** > **Context** from the menu bar.
2. Click **New** to create a context.

   If you want to create a child context that inherits the properties of an existing context, select a context and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.

Field	Description
Applications	Add or remove the applications to which this context applies. For example, adding the `QA` application causes the context to only apply to matching quality assurance cases.
Parents	Add or remove parent contexts for this context to correspond to a context tree.
Type	Add or remove content types.
Condition	Type the DQL-query filter.
Group	Add or remove user groups. The field conjoins groups using the `OR` statement. **Note:** Property Pages and the following configs are only supported for use with Type or Document Type Context and should not be linked to Group context: • auto-naming • auto-link • security config • lifecycle

4. Click **Save**.

Related Topics

“Understanding the Configuration Matrix” on page 15

“Configuring Application Filters and Permissions” on page 20

1.7 Configuring Application Filters and Permissions

1. Navigate to **Tools** > **Application** from the menu bar.
2. Click **New** to create an application.

   If you want to create a child application that inherits the properties of an existing application, select an application configuration and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name.

Field	Description
Description	Type a description.
Label <language>	Add localized labels for the application. These will appear in the Application drop-down on the login page, if enabled.
Application Parameter	Type a URL parameter for the application so you can create intelligent URLs for user to access objects from a browser. For example, an **Application Parameter** of `DocLibrary` would appear in the intelligent URL syntax as shown: `http(s)://[URL]:[port]/D2?repo=corp1&application=DocLibrary` **Note:** The Application Parameter field accepts only valid URL characters.
Include at Login	Allow the application to be selected by the user on the login dialog (requires further setup in **D2 > Options** Runtime section).
User/Group names	Use the list controls to add and remove users and groups.
Permissions	Use the list controls to add and remove permissions. If more than one application is applicable to content, end users can only perform actions shared across all applications. For example, Application A grants the permission **Can drag and drop context**, while Application B grants the permissions **Can drag and drop context** and **Can create, modify, and delete content**. The end user can only drag and drop content because both applications grant the permission.

4. Click **Save**.

Related Topics

"Understanding the Configuration Matrix" on page 15

"Configuring Contexts" on page 19

1.8 Understanding Parent and Child Configurations

You can create a configuration that inherits the settings of and creates a parent-child relationship with an existing configuration. This action copies the configuration module itself, but does not create a child application that has all the modules assigned to that application and context. When you import a configuration update, you can set OpenText Documentum CM to check for changes made to the parent configuration. It then generates a PDF document listing the configuration components that were changed and its child configurations. OpenText Documentum CM does not automatically apply the changes to child configurations because you may not want to change one or more of the child configurations.

If you delete a parent configuration, the child configurations continue to show the deleted configuration as a parent.

Importing a D2 Configuration contains instructions for importing a configuration.

Chapter 2

Defining Repository Terminology

Note: Due to browser display constraints, a 750 row limit is in place for each element described in this section: Dictionary, Taxonomy, and Registered Table.

2.1 Defining Terminology Using Dictionaries

A OpenText Documentum CM Dictionary is a collection of key values and one or more sets of descriptive labels, used to provide value assistance to the end user. Dictionaries help end users by defining choice sets for property values, generating autonaming patterns, and providing input assistance during advanced searches. The choice sets ensure that uniform terminology is used across the context. As a result, it is possible to:

- Increase the likelihood of a property being filled out by an end user.
- Reduce labeling errors such as extraneous spaces, inconsistent plurality, and so on.

You can define values for a dictionary, then apply a language, an alias, or both to each value. Use languages to specify a different value based on the language used by the end user. For example, you can have the value One in English but assign the value Ichi in Japanese. When an end user using an English interface opens a property page with a list box of numbers, they see one of the options is One. An end user using a Japanese interface opens the same property page with the same list box and sees the option Ichi. The stored value is treated as an equivalent due to the dictionary.

You can use an alias to create an alternate equivalent of the value. For example, you can have the value Draft, and assign as aliases the values Rough Draft, First Draft, and In Progress. In configuration components such as property page fields, you can designate the dictionary and alias used. You can use the different terms to assist end users. For example, in some cases when an end user accesses a list box, they see the option Rough Draft. You can use aliases so that in other cases, the end user sees the option In Progress. Despite the difference, both terms are treated as equivalent values and receive equivalent treatment from automatic processes.

The following screenshot shows an example dictionary:

	Key	French	English
☑	change_request	Demande de modification	Change request
☑	directive1	Directive	Form
☑	instruction1	Instruction	Instruction
☑	procedure1	Procédure	Procedure
☑			

You can use two types of dictionaries:

- Static values such as the category of a document, which is a fixed property.
- Dynamic values such as property of an author that is accessed using a DQL query for existing users.

As dictionaries encourage uniform terminology, OpenText Documentum CM can then enforce automatic content linking rules with more confidence. Dictionaries as a whole improve content management from both a manual and automatic perspective. Dictionaries are used in the following configuration components:

- Property pages for combo lists, labels, and so on.
- Creation profile for type and attribute selection.
- Autonaming for alias or languages in naming content.
- Autolinking for using Properties labels for content placement.
- Search mapping for input assistance during advanced searches.
- Column mapping for use of labels and languages.

Related Topics

"Defining Terminology Using Taxonomies" on page 25

"Configuring a Dictionary" on page 26

"Storing a Registered Table" on page 31

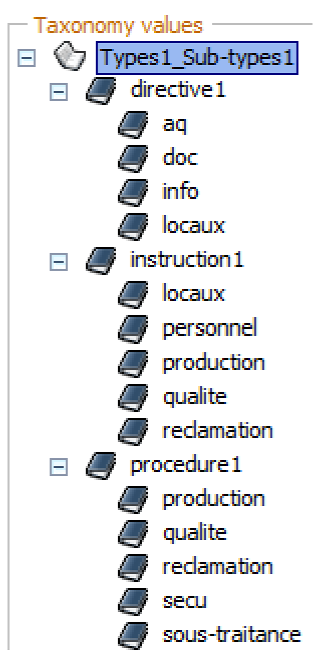
"Importing a Large Dictionary" on page 29

2.2 Defining Terminology Using Taxonomies

A OpenText Documentum CM Taxonomy is a set of values in a tree hierarchy, driven by Dictionaries. Use a Taxonomy if you want to filter values or design a tree of values. You can select dictionaries used in a taxonomy, then select which values to use and at what level of a hierarchy. OpenText Documentum CM uses taxonomies in the property page to allow the display of values available for series of attributes. Like dictionaries, taxonomies help end users by defining choice sets for property values, generating autonaming patterns, and providing input assistance during advanced searches. The choices sets ensure that uniform terminology is used across the context. As a result, it is possible to:

- Increase the likelihood of a property being filled out by an end user.
- Reduce labeling error cases such as extraneous spaces, inconsistent plurality, and so on.

For example, you can define a level of Types, then define subtypes for each type, as shown in the following screenshot:



Related Topics

"Defining Terminology Using Dictionaries" on page 23

"Configuring a Taxonomy" on page 30

"Storing a Registered Table" on page 31

2.3 Configuring a Dictionary

1. Navigate to **Data** > **Dictionary** from the menu bar.

2. Select one of the options for creating a dictionary.

   If you want to create a child dictionary that inherits the properties of an existing dictionary, select a dictionary and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

   Click **Restore dictionary** to open a specific version of an existing dictionary.

   Click **Import** to import a table as a dictionary.

3. Fill out the form as described in the following table:

   Note: Avoid the use of special characters (\/*&^%$#@!\) when creating Dictionary elements such as keys as they can cause unexpected issues with auto-linked paths.

   Make sure that the Dictionary ID and NAME tokens begin with a letter ([A-Z,a-z]) followed by any number of letters, digits ([0-9]), hyphens (-), underscores (_), colons (:), and periods (.).

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this dictionary applies. For example, adding the `QA` application would cause the dictionary to only apply to matching quality assurance cases.
Versioned dictionary	Select to use automatic version control for each save operation.
Groups > Administration group	Select a user group to limit the administration and updating of the taxonomy to only the members of the group through the Taxonomy Admin Widget.
Groups > Search Dictionary available for the group	Select a user group to have the ability to search the dictionary.
Labels > Label <*language*>	Type a label to appear in the configuration matrix and during content creation in the client user interfaces.
Manual sort	Select to enable manual ordering of dictionary values instead of alphabetizing them.

Field	Description
Create a registered table	Select to create an external table based on the values contained within the dictionary. OpenText Documentum CM creates the table when you save the changes. **Note:** The dictionaries share a single database table in the repository. The **Create a registered table** option allows the administrator to define the dictionary in a dedicated registered table.

4. Select **DQL dictionary** to enable DQL dictionary creation.

   DQL dictionary is displayed as an option only when **Dictionary caching** is selected. If **Dictionary caching** is not selected for DQL Dictionary then administrators will not be able to select the dictionary under **Interface > Display Configuration** in client configuration.

   a. Select **Dictionary cache** to enable dictionary updates only when you restart the web application server or when you refresh the cache. Selecting the option increases performance. To use this option, add the Client URLs in the **Client Options** tab.

   b. Type the **DQL query**. The following table describes commands you can use:

Command	Description
`as name`	Set the preceding attribute as a key attribute.
`as is_enabled`	Enable or disable boolean properties.
`as mon_alias`	Create a new matching list for the key. The command is similar to the alias column.
`as label_en/fr`	Translate values
`'$value()'`	Insert into the WHERE statement to refine results

Command	Description
`'$value(filter)'` `'$value(filter_ex)'`	The users’ input will generate DQL that includes the attribute of the field they are typing in. The attribute and user input are tied together. **Note:** It is possible to use `$value(filter)` with `union` queries if the DQL where `$value(filter)` is used employs the same attribute as the user is typing into in each instance that `$value(filter)` appears in the DQL. If this is not the case, you can only use`$value(filter)` once in any given DQL query. Incorrect usage, for example, where the first half of the `union` DQL, `$value(filter)` is using `dm_user` while the second half of the DQL `$value(filter)` is based on `dm_group`.: `SELECT user_name as name , user_name as label  from dm_user where $value(filter) union select group_name as name, group_name as label from dm_group where $value(filter)` `$value(filter_ex)` is also available to perform filter operations for multiple attributes together. For more information, see “Optimizing DQL filtering on multiple attributes” on page 440.

5. Add dictionary values.

   If **Manual sort** was selected, use the up and down arrows to reorder values.

   You cannot delete dictionary values used in a creation profile and property page. Unselect **Key** to remove it from the dictionary.

6. Add or remove **Languages** values.

7. Add or remove **Alias** values.

8. To save the dictionary:

   - Click **Save** to use automatic versioning.
   - Click **Save dictionary** to manually version the dictionary.
   - Click **Export** to export the dictionary in a file format you want.

9. Click **Update repository** to load the changes in the repository without restarting the server.

If the dictionary is modified, restart the Application Server for the changes to be reflected.

Related Topics

"Defining Terminology Using Dictionaries" on page 23

"Storing a Registered Table" on page 31

"Importing a Large Dictionary" on page 29

2.4 Importing a Large Dictionary

You can use the Dictionaries widget to import or export dictionary data in the form of a CSV, Excel, or XML file.

To import large dictionaries, use the `D2JobImportDictionary` job. This job helps avoid performance related problems when you import large dictionaries through a client user interface or client configuration.

1. Create an index for the Oracle table `d2_dictionary_value` using the following command:

   ```
   EXECUTE make_index WITH type_name='d2_dictionary_value',
   attribute='dictionary_name', attribute='object_name'
   ```

2. Import the dictionary using the following command:

   ```
   EXECUTE do_method WITH method = 'D2ImportDictionaryMethod', arguments
   ='-overwrite true -folder <path to the dictionary> -sorted false
    -transaction false'
   ```

Related Topics

"Defining Terminology Using Dictionaries" on page 23

"Configuring a Dictionary" on page 26

"Storing a Registered Table" on page 31

2.5 Configuring a Taxonomy

1. Navigate to **Data** > **Taxonomy** from the menu bar.
2. Select one of the options for creating a taxonomy.

   If you want to create a child taxonomy that inherits the properties of an existing taxonomy, select a taxonomy and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

   Click **Restore taxonomy** to open a specific version of an existing taxonomy.

   Click **Import** to import a table as a taxonomy.
3. Fill out the form as described in the following table:

   Note: XML reserved characters cannot be used in the taxonomy. For example, an error message would appear if a dictionary value included a forward slash character.

Field	Description
Name	Type a name that will be listed in the configuration matrix.
Administration group	Select a user group to limit the administration and updating of the taxonomy to only the members of the group through the Taxonomy Admin Widget.
Description	Type a description.
Versioned taxonomy	Select to enable automatic version control of the taxonomy.

4. Add or remove dictionaries used in the taxonomy. You can only remove the lowest level in the list.
5. Use the list controls to add or remove taxonomy values. Check **Default** to set selected values as the defaults in the 2nd to 7th levels of the taxonomy. These defaults can be used to populate the initial values of Combo fields on a property page.
6. To save the taxonomy:
   - Click **Save** to use automatic versioning.
   - Click **Save taxonomy** to manually version the taxonomy.
   - Click **Export** to export the taxonomy in a file format you want.

If the taxonomy is modified, restart the Application Server for the changes to be reflected.

Related Topics

"Defining Terminology Using Taxonomies" on page 25

"Storing a Registered Table" on page 31

2.6 Storing a Registered Table

Registered tables are relational database management system tables that are not part of the repository but are known to the OpenText Documentum CM Server . The *OpenText Documentum CM Server DQL Reference* contains further information about creating and querying registered tables. You can use registered tables as read-only dictionaries. You can import:

- Exported dictionaries.
- Imported .csv or .xsl files. This option can be used only on an existing registered table to edit its values.

Use registered tables as a reference.

1. Navigate to **Data** > **Registered table** from the menu bar.
2. Select one of the options for showing or editing a registered table. **New** does not create a registered table, you must create registered tables in OpenText Documentum CM Administrator or through a dictionary.

   If you want to create a child registered table that inherits the properties of an existing registered table, select a registered table and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Select a registered table in the repository.
Description	Type a description.
Applications	Add or remove the applications to which this registered table applies. For example, adding the `QA` application would cause the registered table to only apply to matching quality assurance cases.
Administration group	Select a user group to have administration rights.

4. Add or remove table values.

   Add a row by typing values into the empty row at the bottom of the table.
5. To save the registered table:

   - Click **Save** to save the registered table.

- Click **Export** to export the registered table in a file format you want.

Related Topics

"Defining Terminology Using Dictionaries" on page 23

"Configuring a Dictionary" on page 26

"Importing a Large Dictionary" on page 29

"Defining Terminology Using Taxonomies" on page 25

"Configuring a Taxonomy" on page 30

Chapter 3

Configuring Content Creation Components

3.1 Determining Which Creation Processes to Use

You can configure context-dependent automated processes, which are performed when an end user creates content.

Use the following table to determine which processes you want performed:

Process	Description	Example
Context "Configuring Contexts" on page 19 contains further instructions.	You can define which configurations will be applicable for a given action based on the document properties and/or group membership.	Apply the configurations in this Contexts column if the doctype is procedure and the user is a member of the Authors group.
Auto naming Configuring Autonaming contains further instructions.	You can configure an automatic process to change property values.	An end user creates a quality assurance Draft document. An auto naming process can be run to rename the file to include quality assurance (QA) and Draft, such as QA_project_draft_7-30-2014. docx.

Process	Description	Example
Property pages Configuring a Property Page contains further instructions.	You can create and configure property pages to determine the properties that end users can view and edit. If you do not specify a properties page in a creation profile, the content creation wizard skips the properties step. If you remove attributes from an object type using other applications such as Composer or OpenText Documentum CM Administrator, OpenText Documentum CM shows the property pages with blank spaces. You must manually update the property pages to remove the deleted properties.	An end user creates a quality assurance Draft document. You can configure a property page to show a comment field and a checkbox for whether it is a private or public draft. This allows end users to view and add comments to the document and set its privacy status. End users cannot view or make changes to all other properties such as the document title, author, and status.
Inheritance Configuring Inheritance contains further instructions.	You can configure an automatic process to copy properties from existing content to new content.	An end user creates a Project 122 documentation that inherits properties from Project 120 documentation. Keywords such as Project 12X are copied from the 120 documentation to the 122 documentation, alleviating the need to manually copy properties.

Process	Description	Example
Security Configuring Security contains further instructions.	You can configure security models, which define dynamic access control lists (ACLs) that apply to contexts. You can specify a condition based on three identifiers to apply a level of permissions. The three types of identifiers are specific users, user groups, and content properties. If multiple conditions match a specific situation, the most extensive set of rights is applied. If no security model is selected, OpenText Documentum CM applies the default ACL configured depending on the repository configuration.	An end user can belong to the quality assurance Draft Writers group, which gives them WRITE permission within the quality assurance documentation folder.

Process	Description	Example
Auto linking Configuring Autolink contains further instructions.	You can configure an automatic process to place content in specified locations within the repository. You can configure an auto link template to have multiple placements, in which case the content is placed in the first, then linked to the other placements. Changing the property values of content and affecting its context can cause a change in the placement of the current version of the content. If no auto link rule is configured, OpenText Documentum CM adds content to the selected folder. If no folder is selected, OpenText Documentum CM adds content to the cabinet level. Folders with a high volume (roughly 500) of content can have performance issues and errors. Configure auto linking of content to minimize the amount of content within each folder. For example, a project may contain 500 documents, leading to performance degradation. To work around the issue, you can configure an auto link to create and fill subfolders for project phases to reduce the number of content directly found in the project folder.	An end user creates a quality assurance Draft document. The document is placed in the quality assurance Draft folder with all other quality assurance drafts. If the end user sets the document to Private, in which it is hidden from other users, auto link can move the document to a personal, locked folder in the quality assurance Draft folder. When the end user decides to set the document back to Public, auto link can move the document back to the quality assurance Draft folder.

Process	Description	Example
Template list Configuring Template Lists contains further instructions.	You can configure a list of content templates suggested to an end user during the end of the content creation process. Assign lists of templates to contexts, which allows you to provide templates that are only relevant to specific circumstances. End users can be part of the administration group for template lists. This allows end users to configure templates directly without access to client configuration. If there is only one template that an end user can select and the end user does not have template creation privileges, the content creation wizard selects the template and skips the template step.	You can bundle a set of templates for all quality assurance documentation, such as drafts, presentations, and reports. When an end user creates content using the quality assurance application, they can select from a list of quality assurance-specific templates.
VD Template Configuring Virtual Document Templates	You can configure a virtual document template for inheritance of properties and components. End users can use virtual document templates to insert components of a selected virtual document into a virtual document or create a copy of the template virtual document.	An end user creates a presentation content and selects a virtual document template to inherit. The presentation gains the virtual document children applied to the template.
Folder Structure Import "Configuring Folder Structure Import" on page 129 contains further instructions.	You can configure the folder types, autolinking, folder security, set what creation profile will be used for the documents when you allow users to import entire folders, and the folders content, including the folder structure. In addition, you can choose to have the newly imported folder structure automatically converted to a virtual document.	An end user selects the option to import a folder structure so they can recreate a folder and it's content including sub-directories on their system in the repository.

Process	Description	Example
Folder Structure Conversion Configuring Folder Structure Conversion	You can configure the behavior about what happens when a user chooses to convert an existing directory in the repository into a virtual document.	An end user can create a virtual document from an existing folder structure.
Linked document Configuring Document Linking contains further instructions for configuring linked documents for new and imported content. Configuring Linked Documents for Importing Emails contains further instructions for importing attachments as linked documents.	You can configure the content creation process to create additional documents during content creation and link them to the base content. This enables end users to create batches of documents that are controlled by a central document.	An end user can create a business plan, then link notes, presentation slides, and so on to the business plan.
Checkin Configuring Checkin contains further instructions.	You can configure aspects of checkin such as versioning and logging.	An end user checks in a quality assurance Draft Checkin configurations result in no versioning. If an end user checks in a quality assurance In-Review document, checkin configurations can result in minor versioning.
Checkout Configuring Checkout contains further instructions.	You can configure aspects of checkout such as warning messages and checkout range.	An end user checks out an approved document. Checkout configuration displays a warning message reminding the end user that the document has already been approved. If the end user checks out a draft document, checkout configuration can differ, such as showing no warning message.

Process	Description	Example
Smart View Preset Profiles "Configuring Smart View Preset Profiles" on page 106 contains further instructions.	You can configure presets in the Smart View client that provide process shortcuts to users who create or import the same kind of content on a regular basis. With these presets, users can skip several standard decision points and save time. Presets are shown in the creation wizard **Select type** screen and can be added as individual preset tiles on the Landing Page or actions in the **+** menu.	An end user needs to import invoices on a regular basis. All of these invoices have the same document type and associated sub-types, so an invoice preset with this information is created and added as a tile to the Smart View Landing Page, or as a menu item so it can be selected by the user as a shortcut to the import process.
Lifecycle "Configuring a Lifecycle" on page 136 contains further instructions.	You can configure aspects of a lifecycle such as entry conditions to lifecycle states, actions to run when going entering a state, and what should happen when moving to the next state.	An end user takes a new document and applies a lifecycle to it making it a draft document. The lifecycle configuration checks to make sure the user belongs to a specific group that is allowed to perform the action and that the document is not checked out and it is marked as *Current*. Then it sets a new a_status of *Draft* on the document and applies security to it.
Lifecycle batch "Configuring a Lifecycle Batch" on page 145 contains further instructions.	You can configure a lifecycle, target state, and transition type to apply to a group of documents instead of doing so individually.	An end user has uploaded a large number of documents that are **Approved** procedures. The user now needs to make sure they are indicated as approved in the repository and set the applicable attributes associated with Approved SOP documents.

Process	Description	Example
Workflow “Configuring a Workflow” on page 149 contains further instructions.	You can configure a workflow and the various rules on what is allowed to be sent to a workflow, task parameters, and other options associated with each step of the workflow.	An end user wants to have other authors review his document and then approve it for publication. The end user does so by selecting the document and sending it into the workflow, where the workflow validates that it meets various entry criteria and allows the end user to send the document to authors for review and approval.
Menu “Configuring Client Menus” on page 293 contains further instructions.	You can configure the menu options that appear in the client user interfaces for such things as right-click actions, new, import, or content menu items.	You have a set of users where you want to minimize the number of menu choices available to them. Using the menu configuration you can make sure that only the necessary menu options appear when they right-click on documents or use the new, import, or content menus.
User Settings “Configuring Default User Settings” on page 88 contains further instructions.	You can configure the default user settings for checkout and temp directories, date formats, login settings, user interface settings, and notifications.	Your company wants to make sure that all the users by default use their local home directory for the checkout path and the users temp directory for the temp path. In addition, they want menu position to default to the right hand side and list 50 items per page.
Filter “Configuring Display Filters” on page 326 contains further instructions.	Using a DQL query, you can configure a filter that can be applied to a doclist widget to display only specified content in the doclist.	You want end users to be able to quickly see procedure documents modified in the last 30 days, so you create a filter that they can select from the filter menu to quickly filter their view of the content in the doclist folder.

Process	Description	Example
Search "Configuring Global Search Settings" on page 169 contains further instructions.	You can configure what types of content can be searched, what properties of those types can be used to search, if a dictionary or a DQL query should be used, and if the properties should be used for the Facets widgets and how.	You want to create a search configuration so that users can search for all your specific document types, key properties of those types and make sure the properties can be used as facets so the user can further refine their results.
Query form "Configuring a Query Form Search" on page 174 contains further instructions.	You can configure a DQL query or Advanced Search query with facets that users can choose to run from a list of saved searches in a Search Widget or can be used to create a dedicated query form widget that can be added to a workspace.	Your end users have a need to query for specific set of content and refine the results using pre-selected facets, so you create an advanced search query form using specific types and properties, set key properties to be used as facets, and then make this query available via a query form widget in a prominent location of the users workspace.
Mailing list "Configuring a Mailing List" on page 189 contains further instructions.	You can configure the recipients, email subject, message, and attachments that should be included to a mail message.	Your end users need to send messages to users notifying or reminding them of a document that needs to be reviewed, so the end user selects the document and then chooses the mailing list that has been configured. An email will automatically be sent to the pre-defined recipients with applicable message and attachment.
Send mail "Configuring Options for Sending Emails through OpenText Documentum CM" on page 192 contains further instructions.	You can configure email subject, message, and attachment behavior to allow end users to send email message concerning a selected object.	Your end users want the ability to send messages to both internal and external users of the system for a selected object. Using the send mail configuration you have pre-defined the message and subject of the email to streamline this task.

Process	Description	Example
Subscription “Configuring a Subscription” on page 194 contains further instructions.	You can define specific events and associated emails that end users can subscribe to, so they can be alerted when the event occurs.	Your end users want to know anytime someone checks in a new version of a particular document. By creating a subscription, the end user can subscribe to this event for this document and be notified by email when a new version has been checked in.
Distribution “Configuring a Distribution” on page 196 contains further instructions.	You can configure a simple workflow like process where you can define attachments, property pages, recipients, an email message, and any electronic signature requirements when accepting or rejecting the requested distribution actions.	Your end users want a quick way of sending a document for review to some people but want to make sure that they accept or reject the new changes in the document. Further, any approval must be electronically signed and the approver must indicate the reason for approval.
Uniqueness check “Configuring Uniqueness Check” on page 85 contains further instructions.	You can configure a qualification via a DQL statement that can be used in workflows to make sure that objects meet specific qualifications, and if not, what message should be displayed to the end user.	You have configured a specific uniqueness check that makes sure a document being sent into a workflow is not checked out. If it is, then a specific message is displayed to the user who attempts to send that document into the workflow.
Mass update “Configuring Mass Update” on page 86 contains further instructions.	You can a configure warning message, options, a property page to be used, and who can use the configuration to update multiple selection of documents at one time.	You have configured a specific group of users to allow them to update the properties of a group of selected documents rather than the user having to update the documents individually.

Process	Description	Example
Audit “Configuring Audit” on page 217 contains further instructions.	You can configure the list of audited events, which events get displayed in the Audit widget, whether to only list OpenText Documentum CM events, and whether you want to use extended columns and messages to describe events in place of standard event columns and messages.	You have configured several events to be audited and a subset of events to be displayed in the Audit Widget. In addition you want a simplified more user friendly view of the columns and events, so you have enabled *Extended display*. Finally, you set the properties to be audited. Now, when your users who have view audit privileges looks at the audit trail of a document, they see the date, user, document version, and event description.
Rendition server “Configuring a Rendition Server Connection” on page 199 contains further instructions.	You can configure the rendition server to generate renditions for documents and specify the rendition server information related to it's queue, event, message, and format information.	You want to have PDFs renditions created for the documents, so you configure one rendition server configuration to handle that and you have a different rendition server configuration set to generate thumbnail images of the documents.
BOCS cache “Enabling a Branch Office Caching Service (BOCS) Element” on page 201 contains further instructions.	You can configure an existing repository network location to be used by end users when logging in as their BOCS network location.	You have defined the network locations in the repository already, and you want to configure their use by your users so that they are made available to the users based on the configuration context.
Plugin	You can choose from a list of existing plug-ins so that you can create configurations to be used in the matrix.	You want to make sure that one or more of the plug-ins you use are only available for some of your users, based on the context in the matrix. So, you create a plug-in configuration for the desired plug-in and then associate it with the matrix to the desired configuration.

Related Topics

“Configuring Autonaming” on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

3.2 Configuring Autonaming

1. Navigate to **Go to** > **Auto naming** from the menu bar.
2. Click **New** to create an autonaming rule.

   If you want to create a child autonaming rule that inherits the properties of an existing autonaming rule, select an autonaming rule and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this autonaming rule applies. For example, adding the `QA` application would cause the autonaming rule to only apply to matching quality assurance cases.

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Generate the document name for each property modification</td><td>When selected, for every save (<code>dm_save</code>) operation performed on the object, for example:

When a user clicks OK or Save on a properties page with or without any changes.
When using Set Properties and Apply Parameters in a lifecycle that calls the autonaming method.

Then the auto name configuration will be re-applied since these actions result in a <code>dm_save</code> event trigger. This name regeneration action happens regardless if the Auto Naming template includes, or does not include document property, date, or counter placeholders.

For example, if an autonaming rule uses the <code>a_status</code> property of the content and uses the counter value option, the following would be expected:

A user edits the properties of an object, makes a change to <code>a_status</code> field and clicks OK/Save:

The name will be updated and reflect the new <code>a_status</code> value and the counter value will increment.
A user edits the properties of an object makes a change but <code>a_status</code> field does not change and clicks OK/Save:

The name will be updated and still have the unchanged <code>a_status</code> value and the counter value will increment.
A user edits the properties of an object makes no changes but still clicks OK/Save instead of Cancel:

The name will be updated and still have the unchanged <code>a_status</code> value and the counter will increment.
A lifecycle sets the <code>a_status</code> property during a state change and applies these changes causing a <code>dm_save</code> event:

The name will be updated and reflect he new <code>a_status</code> value and the counter will increment.
A lifecycle sets the property of an attribute other than <code>a_status</code> during a
</td></tr>
</table>

Field	Description
	state change and applies these changes causing a `dm_save` event: The name will be updated and reflect the unchanged `a_status` value and the counter will increment.
Regenerate on new version	Select to have the document name updated whenever a new version of the content is created.
Property name	Select the property changed.

4. To add a property to the **Auto naming template**:

   a. Select a property from the **Properties** list box in the **Naming** menu bar.

   b. Select a dictionary if the property uses a dictionary. Select a language or alias.

   Note: Autonaming template does not accept dictionary names that contain brackets.

   c. Type the index position of the value if the property is a repeating property and you want to select a specific value.

   For example, if you want to select the first value of the list, type `0`.

   d. Type the date format if the property is a date property, select a DQL action, and enter the value. Use the following table to understand what values to type for each DQL action:

DQL Action	Value
DATEADD	`year|month|week|day` followed by an integer value
DATEFLOOR	`year|month|week|day`

   Note: The valid year format for autonaming is '`yyyy`'.

5. Select the position in the autonaming template, then type text to insert static text.

6. To configure the counter:

   a. Type the initial counter value in **Counter value**.

   b. Select **Counter depends on template properties** to count subtype changes. When not selected, changes to content subtype do not increase the counter.

   c. Fill the form as follows:

Field	Description
Send email notification to	Type the email address you want to receive notifications.
when the number of remaining values is	Type a counter value. When the number of remaining values reaches this number, a notification is sent to the entered email address alerting the recipient.
Interval start value	Type the starting value.
Interval end value	Type the end value.

d. Click **+** to add another interval of counter values. When one interval ends, OpenText Documentum CM continues the counter from the next sequential interval start value.

7. Click **+** to add a regular expression:

   a. Fill out the form as follows:

Field	Description
Name	Type the name of the regular expression.
Pattern	Type the statement to be replaced.
Replace	Type the replacement statement.

For example, having `underscore`, `_`, `-` respectively replaces all _ with `-`.

   b. Use the up and down arrows to reorder the regular expressions. Regular expressions are run from top to bottom.

8. Select **Do not apply for: Cross repository copy/paste** to disable the auto naming for cross repository copy and paste operations.

9. Click **Save**.

Note: If a user imports multiple files and chooses to use the same creation profile and properties for all files, the Property page wizard step will pre-populate the placeholder label <*[filename]*> instead of showing an individual file name. The property page represents all files being imported.

Related Topics

“Determining Which Creation Processes to Use” on page 33

“Configuring a Property Page” on page 48

“Configuring Inheritance” on page 73

“Configuring Security” on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

3.3 Configuring a Property Page

1. Navigate to **Go to** > **Property page** from the menu bar.
2. Click **New** to create a property page.

   If you want to create a child property page that inherits the properties of an existing property page, select a property page and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the Properties as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this property page applies. For example, adding the `QA` application causes the property page to only apply to matching quality assurance cases.
Document type	Select the document type.

4. Configure Electronic signature options as described in the following table:

Field	Description
Electronic signature required (in edition mode)	Select to require an electronic signature to save properties.
Intention required	Select to require the reason with the electronic signature.

Field	Description
Intentions dictionary	Select a dictionary to enable users to select a reason from a list of options. Alternately, you can also type a custom `$value` into the field. For example: GR Approval $value(site_id) $value(some_custom_property) Reject This input would get the first workflow document (not supporting document) in the task to determine the $value then look for that dictionary config.
Confirmation message <*language*>	Add a custom confirmation message that appears in the client when a user is required to enter their OpenText Directory Services (OTDS) login and password to validate an electronic signature.

5. Configure Dialog options as described in the following table:

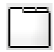
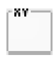
Field	Description
Height	Enter the minimum height of the property page in pixels. If the property page widget in does not scroll down far enough to show all property page content, increase the height of the property page.
Width	Enter the width in pixels.
Resizable	Select if you want the property page to be resizable.
Button position	Select the position of the buttons on the property page.
No initial invalid states	Select if you want to skip the validation check on load so that fields are not shown as invalid before a user has a chance to enter any data.
Auto Edit in Smart View	Select if you want the property page to be editable by default when the user opens the page. **Note:** Consider leaving this option unselected if your property page is complex, which might cause performance issues when the page is first opened. Users have the ability to make the property page editable through an **Edit** button on the user interface.

Field	Description
Smart View Version	Select an alternate version of the property page that will appear in the Smart View client. In some cases, you might need to create an alternate version of a property page if certain page elements do not translate well to the Smart View interface. For more information about which Classic property page elements are supported in Smart View, see “Smart View: Property Page Config Support” on page 541.
Mobile View Version	Select an alternate version of the property page that will appear in the mobile browser or mobile app. In some cases, you might need to create an alternate version of a property page if certain page elements do not translate well to the dimensions of the mobile browser.
Focused Control	Select an editable control on the property page that receives automatic focus when the user opens the page in the clients.

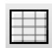
6. Configure Structure of the property page.

   a. Add organizational objects to the property page. Use the following table to determine which organizational objects to add:

Icon	Name	Description
	Copy	Copies a node, children of a node, or entire root node from a property structure.
	Paste	Pastes the copied node from one property page to another. **Note:** You cannot copy/paste duplicate properties in the same property page.
	Delete	Deletes the selected node.

Icon	Name	Description
	Tab unit	Creates an area to which you can add tabs.
	Tab	Creates a tab in a tab unit. Type a label for the field.
	Field set	Creates a fieldset to cluster property objects together. Type a label for the field and select whether to use a border.

Icon	Name	Description
	Grid	Creates a table using properties as columns. DQL using dependencies on values of other columns and taxonomies configured inside the Grid do not function. Type a height for each grid in pixels. All controls added to the grid can be toggled to have **No empty cell**. When the checkbox is selected, the field must be populated before the properties can be saved. Check **Row Based Evaluation** if you would like the enabled and required conditions for the properties in your grid to be evaluated for each row independently, instead of the column-based default that evaluates for the entire grid. This allows the user to fill in each grid row individually.**Row Based Evaluation** disables the **Visibility** condition and **No Empty Cell** option for each property in the grid. **No Empty Row** can be set, which disregards the condition required for each attribute. Select **Auto-fit (Smart View)** to equally distribute column widths across the table (Smart View only). If you deselect this option, define the width of the columns by entering a pixel value in the **Width** setting. **Note:** Repeating attributes in a grid are stored independently of one another when a

Icon	Name	Description
		property page is saved.
	Column set	Organizes property objects by a configured number of columns. Type the number of columns in the set and add objects. The property page automatically organizes objects in columns.
	Comment	Adds a static comment with font styles. Type a label for the field. You can use HTML in the label by selecting the option, or you can use the $value and $alias keywords. Select font color, size, and style.
	Empty cell	Adds an empty cell. Use empty cells in grids and column sets to organize property placement.
	Separator	Adds a horizontal bar. Use separators to add a visual marker for different sections of the property page.

b. Configure **Display time** and **Default time value** options to display the time setting options in the date picker calendar panel.

   You can set default time as **Current time** or **Specific time**.

   Note: If you are configuring for mobile, do not enable the **Display time** check box as this will cause display issues with the date picker control on the mobile property page.

c. Configure property conditions. Configuring Property Conditions contains further instructions.

7. Add property objects.

You can preview the property page by clicking the **Display preview in create mode**, **Display preview in edit mode**, and **Display preview in import mode** buttons. The buttons show a dialog box with the selected property page form, allowing you to check your work. Permissions configured for each property are active for the preview. If you do not belong to the correct groups, certain properties may not appear when previewing properties.

a. Use the following table to determine which property objects to add:

Notes

- Several of the following objects allow property attributes to be set to **Load Asynchronous**, but still allow you to disable the attribute so it is not available in the user interface. The values in a combo field, for example, are only loaded when the attribute is clicked on, so if the property is asynchronous, then the property should always be visible and enabled in the user interface so the user can click on it to run a query. Data loss could result from a disabled asynchronous attribute setting.
- Use of the **Text** and **Checkbox** fields mapped to an attribute of `date type` is supported for view or disabled only use cases in Smart View client.
- For mobile, in the case of Create content properties, tab group, grid, and multi-select field sets will not be supported. All columns will be flattened and displayed as a single column layout (without any accordions).

Icon	Name	Description
	Label	Adds as static text the property name, a colon, and the value of the property. Select a property. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer. For example, if you select the property `date_finish` and type the label `Finishing Date`, the property page shows **Finishing Date : 8/9/2017** If you type the label `Finish date` the property page instead shows **Finish date : 8/9/2017** You can also define this label to display an aspect property. An aspect can be one or more pieces of information associated with the file, such as sensitivity labels from Microsoft Purview Information Protection. To display an aspect property, set the following options: • Enable **Do not link to type property**. • Enable **Link to aspect property**. • Select the appropriate aspect from **Aspect name**. • Select the aspect property you want to appear on the properties page from **Aspect property**.

Icon	Name	Description
		• Type a label if you do not want to use the aspect property name.
	Text	Adds a text field for end users to type values into and save. Select the property for storing the value and type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer.
	Password	Adds a text field for end users to type a password into for saving. The password displays during auditing and when producing info dumps, so make sure end users account for the risk of password exposure. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer.
	Memo	Adds a text box for end users to type a memo into and save. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer and the row height of the field. You can add a memo inside a grid object, but this is only supported in Smart View.

Icon	Name	Description
	Rich text editor	Adds a rich text editor for users to format the content or value of a property. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer.
	Boolean	Adds a checkbox that returns a boolean value. Type a label if you do not want to use the label defined for the property in OpenText Documentum CMComposer.
	Date	Adds a date field for end users to select a date from a calendar. The server date and time is used to register the selected date and can show an error when you click **Save** if the dates do not match due to timezone differences. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer.

Icon	Name	Description
	URL hyperlink	Adds a list field for end users to add, remove, reorder, and access external links. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer and select which action you want users to perform to open the URL.

Icon	Name	Description
	Combo field	Adds a list for end users to select values from a DQL query, a dictionary, or a taxonomy. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer and select **Load asynchronously** to load values on click and filter as opposed to when the dialog box is opened. Use the option when the list contains a large number of values. If the combo box includes a single value and you want to populate the field with that value by default, select **Auto select single valued list**. You do not have to select the value from the list; it is auto-populated. You can set the field value to display a blank, if required. If you do not select **Auto select single valued list**, a blank value gets saved to the object. However, when another user views the property page, the field displays the single value instead of a blank. You can set the combo field's **Trigger Action** to control if a drop down list is filtered when the user activates the combo field. Choose one of the following options: **(Empty/No Value)**: Empty uses the following defaults: QUERY is used for DQL value assist and ALL is used for other cases. **ALL**: Trigger shows all values; See ALL_CLEAR

Icon	Name	Description
		if using DQL and <'*$value(filter)*'>. **ALL_CLEAR**: Trigger shows all values, and filter in <'*$value(filter)*'> is explicitly cleared so the server returns all values. **QUERY**: Trigger shows a filtered list. When **TYPE** is selected as Taxonomy, two settings are available: **Previous Property** and **Next Property**. If Combo with **TYPE** taxonomy has **Previous Property** set, than make sure this previous property is added next or below this combo. **Note:** The editable and non-editable combo fields implement client side filtering that is compatible with server side `'$value(filter)'` use. The wildcards `*` and `?` can be used to match ANY or a SINGLE character respectively. If no wildcards are specified, values that Starts with the filter value are matched. Wildcards can be added to match more complex patterns. For example `*group*` will match any name that contains the word group. If you do not to apply standard client side filtering to list items, select **No client filtering**.

Icon	Name	Description
		Client filtering implements filtering compatible with server side filtering that is performed when DQL includes `'$value(filter)'`. This client filtering can be undesirable if DQL contains alternate filtering when not using `'$value(filter)'`. When using `'$value(filter)'`, the users' input will generate DQL that includes the attribute of the field they are typing in. The attribute and user input are tied together. **Note:** It is possible to use `$value(filter)` with `union` queries if the DQL where `$value(filter)` is used employs the same attribute as the user is typing into in each instance that `$value(filter)` appears in the DQL. If this is not the case, you can only use `$value(filter)` once in any given DQL query. Incorrect usage, for example, where the first half of the `union` DQL, `$value(filter)` is using `dm_user` while the second half of the DQL `$value(filter)` is based on `dm_group`.:

Icon	Name	Description
		SELECT user_name as name , user_name as label from dm_user where $value(filter) union select group_name as name, group_name as label from dm_group where $value(filter)

Icon	Name	Description
	Editable combo field	Adds a list for end users to select values from a DQL query, a dictionary, or type. End users can type text in the field to filter the list of values. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer and select **Load asynchronously** to load values on click and filter as opposed to when the dialog box is opened. Use the option when the list contains a large number of values. You can also type a width and height for the popup dialog box shown to the end user. If the combo box includes a single value and you want to populate the field with that value by default, select **Auto select single valued list**. You do not have to select the value from the list; it is auto-populated. You can set the field value to display a blank, if required. If you do not select **Auto select single valued list**, a blank value gets saved to the object. However, when another user views the property page, the field displays the single value instead of a blank. You can set the combo field's **Trigger Action** to control if a drop down list is filtered when the user activates the combo field. Choose one of the following options: **(Empty/No Value)**: Empty uses the following defaults: QUERY is used

Icon	Name	Description
		for DQL value assist and ALL is used for other cases. **ALL**: Trigger shows all values; See ALL_CLEAR if using DQL and <*'$value(filter)'*>. **ALL_CLEAR**: Trigger shows all values, and filter in <*'$value(filter)'*> is explicitly cleared so the server returns all values. **QUERY**: Trigger shows a filtered list. **Note:** The editable and non-editable combo fields implement client side filtering that is compatible with server side `'$value(filter)'` use. The wildcards * and ? can be used to match ANY or a SINGLE character respectively. If no wildcards are specified, values that Starts with the filter value are matched. Wildcards can be added to match more complex patterns. For example `*group*` will match any name that contains the word group. When using `'$value(filter)'`, the users' input will generate DQL that includes the attribute of the field they are typing in. The attribute and user input are tied together.

Icon	Name	Description
		Note: It is possible to use `$value(filter)` with `union` queries if the DQL where `$value(filter)` is used employs the same attribute as the user is typing into in each instance that `$value(filter)` appears in the DQL. If this is not the case, you can only use `$value(filter)` once in any given DQL query. Incorrect usage, for example, where the first half of the `union` DQL, `$value(filter)` is using `dm_user` while the second half of the DQL `$value(filter)` is based on `dm_group`.: `SELECT user_name as name , user_name as label from dm_user where $value(filter) union select group_name as name, group_name as label from dm_group where $value(filter)` If you use a DQL query, use **Validate** to check the query. See DQL query validation on page 72

Icon	Name	Description
	List field	Adds a list for end users to select several values from a DQL query or a dictionary. The selected values must have the **Repeating** property. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer, select the row height for the field, select whether to sort the list in alphabetical order, select whether to enable import and export of the list of values from and to XLS documents, and type the height and width of the selection dialog box. Select **Load asynchronously** to load values on click and filter as opposed to when the dialog box is opened. Use the option when the list contains a large number of values. Select **Load on selection popup opening** to load values when the selection dialog box is opened. If not selected, users must press `ENTER` to load the list. If the list box includes a single value and you want to populate the field with that value by default, select **Auto select single valued list**. You do not have to select the value from the list; it is auto-populated. You can set the field value to display a blank, if required. If you do not select **Auto select single valued list**, a blank value gets saved to the object. However, when

Icon	Name	Description
		another user views the property page, the field displays the single value instead of a blank. Select **Auto select multiple valued list** to pre-populate the field with the default multiple values. Select **Always update upon Edit** to update the list after each edit attempt. **Display Horizontal** is a Smart View only client setting that allows you to display the list values in a horizontal orientation instead of the default vertical orientation. If you use a DQL query, use **Validate** to check the query. See DQL query validation on page 72

Icon	Name	Description
	Editable list field	Adds a list for end users to select several values from a DQL query or a dictionary, or add text. The selected values must have the **Repeating** property. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer, select the row height for the field, select whether to sort the list in alphabetical order, select whether to enable import and export of the list of values from and to XLS documents, and type the height and width of the selection dialog box. Select **Load asynchronously** to load values on click and filter as opposed to when the dialog box is opened. Use the option when the list contains a large number of values. Select **Load on selection popup opening** to load values when the selection dialog box is opened. If not selected, users must press `ENTER` to load the list. If the list box includes a single value and you want to populate the field with that value by default, select **Auto select single valued list**. You do not have to select the value from the list; it is auto-populated. You can set the field value to display a blank, if required. If you do not select **Auto select single valued list**, a blank value gets saved to the

Icon	Name	Description
		object. However, when another user views the property page, the field displays the single value instead of a blank. Select **Auto select multiple valued list** to pre-populate the field with the default multiple values. Select **Always update upon Edit** to update the list after each edit attempt. If you use a DQL query, use **Validate** to check the query. See DQL query validation on page 72
	Radio	Adds a list for end users to select a single value with a single click. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer and select whether to list the options vertically instead of horizontally.
	Checkbox	Adds a list for end users to select multiple values. The selected values must have the **Repeating** property. Type a label if you do not want to use the label that was defined for the property using OpenText Documentum CM Composer and select whether to list the options vertically instead of horizontally.

Icon	Name	Description
	Controlled relation	Adds a controlled relation field for users to add related documents for that relationship type. Select a configured relationship type from the **Relation** type list box. **Notes** • Do not use the `D2_COPY_OF` relationship type because it is an internal relation used for inheritance. When creating or importing content, the system does not create a relationship for the `D2_COPY_OF` relationship type and does not show the relation in the property page. • Controlled relation fields are not supported for relation properties (*`dm_relation`* or its subtypes).

Icon	Name	Description
	Button	Enables the ability to configure buttons. Only two buttons are supported: **OK** and **Cancel**. Button labels and other details can be configured. The following are the configuration options: • **Action executed after the property save**: Not supported. • **Label**: Enter a label for the button if you do not want to use the property defined in OpenText Documentum CM Composer. • **Width**: The width of the button. This option only affects Classic client. • **Primary button**: Set the button as the default selection of all displayed buttons. A primary button has different coloring than other buttons and typically the button that users need to use to continue their task. If you configure both buttons as primary buttons, they both use the primary color scheme. • **Honor button labels in Dctm Smart View clients**: Select this option to override default labels in Smart View and use only the labels defined in this configuration.

b. To fill the field with a value, select a **Type** and fill out the subsequent fields. You can select a type for labels, radio buttons, checkboxes, combo fields, and list fields.

The following table describes keywords that you can use if you select **DQL**:

Keyword	Description
as name	Save the value as the **Name**.
as label	Save and use the value as the **Label**.
'$value(*<property name>*)'	Use the value of the property *<property name>*.

If you load asynchronously, a DQL query for loading values must include as name, as label, and '$value(filter)'

For example, select user_name as name, user_name as label from dm_user where '$value(filter)' order by 1

Ensure that you use specific queries such as the above example. The system might not be able to process large sets of results returned by queries that are too general, such as select * from.

c. To save the value of the field to a property, select a **Property**. If you do not want to save the value to a property, select **Do not link to property** and type an ID for the field in **Control id**. Use the control ID to create value assistance fields. Configuring Value Assistance Fields contains further instructions on linking fields to create dynamic forms.

Note: Do not enter a Control ID that is the same as a property name. If you do, the value of the field is added to that property.

d. Configuring Property Conditions contains instructions on configuring conditions in which a property applies.

e. When applicable, use **<** and **>** to configure reinitialization of properties when content is modified.

8. Click **Save**.

Note: While saving a property page, incomplete property conditions are not validated if they are based on the fields that do not exist in the property page.

DQL query validation

If you use a DQL query in a combo field, editable combo field, list field, or editable list field, you can click **Validate** to check your query. The validation can check simple queries containing a single attribute, table, and expression.

These are the following validation results:

- Valid: Query border turns green. Your query will work.

- Warning: Query border turns yellow and a **Warning** message appears. The query is too complex to be validated. You must test your query in OpenText Documentum CM.
- Invalid: Query border turns red and a **Invalid Syntax** message appears. Your query will not work.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

"Configuring Property Conditions" on page 96

"Configuring Value Assistance Fields" on page 95

3.4 Configuring Inheritance

1. Navigate to **Go to** > **Inheritance** from the menu bar.
2. Click **New** to create an inheritance rule.

   Select an inheritance rule and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.

Field	Description
Applications	Add or remove the applications to which this inheritance rule applies. For example, adding the `QA` application would cause the inheritance rule to only apply to matching quality assurance cases.

4. Use the list controls to add and remove inherited properties.

5. Select the relation type used during inheritance. The following table describes the differences:

Relation type	Description
Default	Use the `COPY_OF` relation.
None	No relation used.
Other relation	Select a defined relation type from the list box.
Link new content as child	Enable this option to import an object as a child of the selected object.

6. If you want to prevent inheritance of Virtual Document structure:

   a. Type the inheritance prevention condition for each Virtual Document component in **Testing condition on each component**.

   b. Type a warning message in **Warning message**. The message is shown when inheritance is performed.

7. Click **Save**.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

3.5 Configuring Auto Link

Notes

- Auto linking configurations do not apply to Smart View workspaces or their contents.
- Do not include a workspace, the `dm_bws` type, or its subtypes in an auto link path. Smart View workspaces cannot be created with an auto link. This is an incorrect configuration and can cause errors.

1. Navigate to **Go to** > **Auto link** from the menu bar.
2. Click **New** to create an auto linking template.

   Select an auto linking template and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this auto linking template applies. For example, adding the `QA` application would cause the auto linking template to only apply to matching quality assurance cases.
Label for undefined property	Type a name for the folder to be created if a property is not filled. For example, if you define the path :/*<Property 1>*/*<Property 2>*/*<Property 3>* If *<Property 1>* = `Apple` and *<Property 2>* = `Berry`, but *<Property 3>* is not set with a value and you typed `Orange` in the **Label for undefined property** field, the auto link template places the content in `:/Apple/Berry/Orange`
No empty folder on the extremity	Select if you do not want to create a folder for a missing property value. For example, if you take the example from **Label for undefined property**, instead of placing the content in `:/Apple/Berry/Orange` the content is placed in `:/Apple/Berry`

Field	Description
Link all versions to current version folders	Select if you want all versions of the content to be moved to the new location in the repository. If you do not select this checkbox, other versions are kept in their original locations.
Ignore Autolink of previous versions during edit	This option is available if you select **Link all versions to current version folders**. When selected, editing a previous version (major, minor, or branch versions) of the document using lifecycle with autolink action does not move the edited document to any other folder. The document remains in the same autolinked folder of the current version of the document.

4. Configure repeating properties:

Fields	Description
Link only the first value of the repeating properties	Select to link the first values of each repeating property together.
Link repeating properties between themselves	Select to link repeating attributes together.

The following table shows examples of the two checkboxes:

Property 1	Property 2	Property 3	No checked boxes	Link only the first value of the repeating properties	Link repeating properties between themselves

`A,B`	`C,D`	`E` (single)	`/A/C/E`
			`/A/D/E`
			`/B/C/E`
			`/B/D/E`
				`/A/C/E`	`/A/C/E`
					`/B/D/E`

Property 1	Property 2	Property 3	No checked boxes	Link only the first value of the repeating properties	Link repeating properties between themselves

`A,B`	`C,D`	`E` (repeating)	`/A/C/E`
			`/A/C/Undefined`
			`/A/D/E`
			`/A/D/Undefined`
			`/B/C/E`
			`/B/C/Undefined`
			`/B/D/E`
			`/B/D/Undefined`
				`/A/C/E`	`/A/C/E`
					`/B/D/Undefined`

5. Click **Add path** to add a path.
6. Click **/+** to add a folder to a path.
7. Click the text field above the folder category to select and configure a folder property:
   a. Select a property from the **Properties** list box.
   b. Type a **Repeating Index** if the selected property is a repeating property.
   c. If applicable to the selected property, select a dictionary from the **Dictionaries** list box, then select an alias or language.
   d. Click the hand button to add the property.
   e. You can type static text into the folder property.
   f. To reorder properties, drag and drop properties.
8. Click the space between the **/** symbols and select a default value to associate with the content at different folder levels:
9. Click the space below the default values and select an ACL template for the security of the folder:
10. To set the cabinet to your personal folder, select **Home cabinet**.
11. Click **+** to add a regular expression.
    a. Fill out the form as follows:

Field	Description
Name	Type the name of the regular expression.
Pattern	Type the statement to be replaced.
Replace	Type the replacement statement.

For example, having `underscore, _, -` respectively replaces all _ with `-`.

b. Use the up and down arrows to reorder the regular expressions. Regular expressions are run from top to bottom.

12. Click **Save**.

Notes

- Auto Linking is performed only by the installation owner account. You cannot restrict this feature to a specific set of users or groups.
- If auto link is required, you must enable it in the `D2FS.properties` properties file. To do so, open the `D2FS.properties` file and change the *`copyPasteWithAutolink`* property to `true`. It is set to `false` by default.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

3.6 Configuring Template Lists

1. Navigate to **Go to** > **Template** from the menu bar.
2. Click **New** to create a template list.

   If you want to create a child template list that inherits the properties of an existing template list, select a template list and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this template list applies. For example, adding the `QA` application would cause the template list to only apply to matching quality assurance cases.
Enable document without content	Select to allow documents with empty content bodies.

4. You can create a static, defined list of templates or a dynamic list of templates based on a DQL filter. To create a static list, select **List**.

   a. Click **Browse** to show the **Templates** dialog box and use **<** and **>** to configure the list of template documents.

   For each template, you can add a template display condition. Select the template, then type a DQL statement in **Selected template display condition**.

   Note: The file extension .vd is not valid for template creation. If you want to create a virtual document template, use the file extension .vdoc.

   b. Click **Import** to show the **Import** dialog box add a template to the repository, then fill out the form as described in the following table:

Field	Description
Name	You can change the name after importing the file. By default the name of the imported file name is used.

Field	Description
File	Click **Browse**, then locate and select the document to use as a template. Do not select a blank document as a template.
Format	Select the file format.

Click **OK** to close the **Import** dialog box and import the template to the repository.

c. When you have imported all templates you wish to add to the repository, click **OK** to close the **Templates** dialog box.

5. To create a dynamic list of templates, select **Query**, then type a DQL filter into **Qualification** to create a template list based on a DQL query of the templates in the repository.

   For example, `mes_templates where a_status = 'Approved'` creates a list of templates that have been set to Approved. The produced list uses the `object_name` property as the template name.

6. Click **Save**.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

"Setting Up Thumbnails for Template List Previews" on page 81

3.6.1 Setting Up Thumbnails for Template List Previews

When renditions that are generated using the ADTS are available for the template, a preview of the template file appears. You can configure this by ensuring that template files have the correct `.jpg` rendition associated with them. To do so, you can:

- Configure the ADTS to automatically create thumbnail renditions for the environment. In this case, you should see previews instead of file format icons.
- Request a thumbnail rendition from the rendition server.

When a thumbnail server is configured, the system uses the `.jpeg` thumbnail files as follows:

- `jpeg_lres`: Used in the Thumbnails widget when an end user sets the page modifier for the widget to `large_jpeg_th`. The pictures displayed in the Thumbnails view are renditions with `full_format jpeg_lres` and the page modifier `large_jpeg_th`.
- `jpeg_lres`: Used in the Preview widget.

Use the following query to retrieve the URL used to display the thumbnail:

- `select r_object_id, mfile_url('jpeg_lres',0,'large_jpeg_lres') from dm_sysobject where r_object_id = '`

To preview, use query rendition with `jpeg_lres` format.

Related Topics

"Configuring Template Lists" on page 79

3.7 Configuring Check-in

1. Navigate to **Go to** > **Check-in** from the menu bar.
2. Click **New** to create a check-in setting.

   Select a check-in setting and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.

Field	Description
Applications	Add or remove the applications to which this check-in setting applies. For example, adding the `QA` application would cause the check-in setting to only apply to matching quality assurance cases.
Silent check-in	This is an optional setting that will increment the version by the selected value without presenting the default check-in dialog. If the operation is combined with other options below, the check-in dialog might still appear, but the document will always be incremented by the selected value. Select **Same version** to have the check-in performed in the background for same versions. Select **Minor version** to have the check-in performed in the background for minor versions. Select **Major version** to have the check-in performed in the background for major versions.
Display property page with check-in	Select to display a property page dialog during check-in. This option is also available if you have configured **Silent check-in** options.
Choose property page	(Optional) Select the desired property configuration (if any). If left blank, the property page is selected based on the matrix configuration. This option will be disabled if **Silent check-in** option is selected. Preview icon button allows to view the selected property page configuration in a dialog window. Preview icon is enabled only when a property page is selected in the **Choose property page** field.

Field	Description
Allow check-in as	Select **Same version** to allow check-in of same versions. This option is applicable only for the current version of the document. Select **Minor version** to allow check-in of minor versions. Select **Major version** to allow check-in of major versions. Select **Branch version** to allow check-in of branch versions. This option is applicable only for the non current versions of the document. It enables user to check-in the non current version of the document as a branched version. By default, minor and branch versions are selected. These options will be disabled and unchecked if **Silent check-in** option is selected. **Note:** You must select at least one of the three options for same, minor, or major versions.
Always set as current version	Select to remove the **Set as current version** option in the client's check-in dialog so when check-in occurs, the file is always set as the current version.
Always unlock new version	Select to remove the **Retain lock in repository** option in the client's check-in dialog so when check-in occurs, the file is always unlocked.
Allow check-in from file	Select to allow check-in from the file, such as when importing content as a new version.
Display log entry	Select to show the **Log entry** property for the user to add comments during check-in. The comments are saved in the `log_entry` property and can be shown in the audit.
Require log entry	Select to make log entry required. This option will be disabled if **Display log entry** option is unchecked.

Field	Description
Reset log entry before each check-in	Select to clear the **Log entry** field after each check-in. This option will be disabled if **Display log entry** option is unchecked.
Request new rendition	Select to request new renditions of the content.
Keep symbolic version label	You can configure processes such as lifecycles to apply additional labels on the content version. Select this option to keep the label on the current version. For example, if you apply the `DRAFT` label, the current content appears as: • `1.0,CURRENT,DRAFT` If this option is selected and an end user creates a new minor version, the version labels transfer: • `1.1,CURRENT,DRAFT` • `1.0` If this option is not selected and an end user creates a new minor version, the additional labels on the version in which they were introduced remain: • `1.1,CURRENT` • `1.0,DRAFT`
Sync shared files in native format	When enabled, the system shares content externally in the native format. Export configuration rules requiring a PDF rendition will not apply.

4. Click **Save**.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

3.8 Configuring Uniqueness Check

You can configure OpenText Documentum CM to check for content uniqueness using DQL qualification.

1. Navigate to **Go to** > **Uniqueness check** from the menu bar.

2. Click **New** to create a uniqueness check.

   If you want to create a child uniqueness check that inherits the properties of an existing uniqueness check, select a uniqueness check and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this uniqueness check applies. For example, adding the `QA` application would cause the uniqueness check to only apply to matching quality assurance cases.

4. Click **Add a check** to add a DQL qualification:

   a. You can use `$value` and `#repeatingvalue` in **Qualification** and **Message**.

   b. Fill out the row as described in the following table:

Field	Description
Qualification	Type a DQL directive. This inserts i_chronicle_id to the predicate if the DQL qualification contains a WHERE clause This is done to avoid testing on the same object but in different location.

Field	Description
Message <*language*>	Type a warning message for when the content is not unique.

5. Click **Save**.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Mass Update" on page 86

3.9 Configuring Mass Update

You can allow users to update multiple content with a single process using the mass update component. Mapping a mass update configuration with a context allows you to define the documents that must be updated. If you do not map a mass update configuration to any context in the client, although the configurations appear in the list, none of the documents get updated.

Note: Mass update does not support property page configurations where `$value` is configured in fields. This includes e-signature intentions where `$value` is used.

1. Navigate to **Go to** > **Mass update** from the menu bar.
2. Click **New** to create a mass update.

   If you want to create a child mass update that inherits the properties of an existing mass update, select a mass update and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this mass update applies. For example, adding the `QA` application would cause the mass update to only apply to matching quality assurance cases.
Do not display warning message	Select to prevent a warning message for confirming the mass update.
Do not display mass update options dialog	Select to prevent showing the dialog box for end users to fill out the mass update property form.
Smart View options (for Files and Folders)	Choose what gets updated automatically for mass update target items **Files** and **Folder** in the Smart View client when you selected the **Do not display mass update options dialog** checkbox.
Update all version of files	This check box is available only if you choose **Update selected files** from the **Files** list. Select to configure the mass update to change all versions of the selected files. **Note:** This might be a very long running process if the user selects a large number of files.
Update virtual document and all descendants	Select to update a virtual document and all of its components. This option is disabled when you are not configuring a virtual document property page.
Update folder and all sub-folders and objects	Select to update sub-folders and content within sub-folders. This option is disabled when you are not configuring a folder property page.
Property page	Select the property page the user fills out.
Display only for users in this group	Select the user group that can use this mass update. You can leave the field empty to allow all end users to access this mass update configuration.
Label *<language>*	Type a label.

Notes

- Grids are not supported in mass update for the Smart View client.
- When performing a mass update, an error message appears listing any items with autolink paths that exceed 450 characters. These files are excluded from the mass update and not auto-linked.

4. Click **Save**.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

3.10 Configuring Default User Settings

Default user settings are the values for individual user settings that are applied when users log in for the first time. In addition to being set as a default in these user settings, a subset of these settings are configurable by the user in User Settings.

Note: If you manually deleted the `x3_preferences` and `d2c_preferences` objects to clear the cache, you must run a **Save** in **D2-Config** > **User Settings** in order to recreate the needed objects applicable to `x3` and `d2c` preferences.

1. Navigate to **Widget view** > **User Settings** from the menu bar.
2. Click **New** to set default user settings.
3. Specify the properties as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which these user preferences apply.

4. Fill out the **General** tab as described in the following table. All the general property settings are optional.

Field	Description
Temporary path	Specify the directory in which to save viewed files. Viewed files are not checked out. The system deletes files the next time you log in to the system.
Checkout path	Specify the directory in which to save checked out files.
Import Content	Select the checkboxes to set the default import files setting options. **Same creation profile for all files** and **Same properties for all files** are available. **Note:** An administrator can set the default import file setting for the users. The default import setting is available to users from their initial login. However, a user can change this import setting in the Import dialog based on the requirement.
Date format	Select the date format used for content properties. If no value is specified, the default date format is `d MMM yyyy`.
Date and time format	Select the date and time format used for widget columns. If no value is specified, the default date and time format is `d MMM yyyy HH:mm:ss`.
Date input format	The date format used for the date picker or date input controls. The default date input format is determined by locale. For example, if the locale is `EN`, the default date input format is `dd-MMM-yyyy`, otherwise the default format is `MM/dd/yyyy`.

Field	Description
Default language	Select the default language for the user interface. This overrides the web browser language setting. Your system may be configured to hide this field.
Default Workspace	Select the default workspace(s) for the users. **Note:** Make sure you map these user settings in the hyper matrix (which has default workspace configuration). **Do not allow user to change default workspaces**: Select this checkbox to lock the default Classic View workspace selection for the users. Such workspaces are displayed with the lock icon for users. You can uncheck the checkbox to allow users to change the default works pace settings. Even with default workspaces locked, users can add additional non-default workspaces.

Field	Description
Login settings	**Restore Session**: Select this option to restore the previous Classic View workspaces and make this option available to the users when they log in. This option is selected by default. **Reset application/default workspaces**: Select this option to set the default Classic View workspace settings for the users. If selected, the same will be reflected in User Settings. **Do not allow user to change default workspaces**: Select this option to lock the default Classic View workspace selection for the users. You can clear the checkbox if you want to allow users to change the default workspace settings. **Do not allow user to change restore session option**: Select this option to make the "Restore session" or "Reset application/default workspaces" options non-editable for users. This checkbox is cleared by default. **Note:** You can also choose to restrict users' concurrent login sessions. See `allow_concurrent_login` in section 26.1 "D2FS.properties settings reference" in *OpenText Documentum Content Management - Client Installation Guide (EDCCL-IGD)* for more information.

5. Fill out the **User Interface** tab as described in the following table:

Field	Description
User client experience	Select the content transfer framework defaults and fallbacks for client users. If you leave this field blank, users will be able to select their own defaults from User Options in the client interfaces. • **Thin**: uses the native functionality of the client browser. • **WSCTF**: sets Web Socket Content Transfer Framework as the default mode. • **WSCTF, Thin**: sets Web Socket Content Transfer Framework as the preferred mode, falling back to Thin • **Thin, WSCTF**: sets Thin as the preferred mode, falling back to Web Socket Content Transfer Framework. **Note:** For new users, User settings for User Client Experience configuration will be honored automatically. For existing users, data needs to be cleared in the `d2c_preferences` and `x3_preferences` table to recognize the new configuration. These tables can be cleared by running the `D2JobPreferencesUpdate` job. For example, if an existing user has Thin set, and the administrator changes user settings to WSCTF, Thin, the existing user will continue to use Thin until the administrator runs the job.
Content by page	Select the number of items you want to show in each page of a list.
Group folders then files	Select to sort and group list widget folders first, followed by files.
Table row height	Select the height of a row in a list.
Menu position	Select to position the menu at the top or left of the position. Log out and log in to see the change.

Field	Description
Template Selection Default	Select the default view for content creation templates: **Gallery View** (which shows graphical representations of each template), or **List View** (which summarizes each template with a set of non-configurable columns: **Name**, **Title**, and **Subject**). The **Subject** field of the d2c_template object that represents your template field can be updated to contain information to help the user choose which template to use.
Smart View accessibility enabled	Select to enable Accessibility Compliant Mode in the Smart View client, which improves the usability of accessibility aids such as screen readers by: • Always using the native browser scrollbars. • Ensuring all toast messages persist until cleared by the user.
Smart View on task completion	Select how the Smart View Task user interface responds when a Task is completed or aborted by the user. **No task selected**: The Task interface does not automatically move to the next task in the user's list. **Next task in list**: The Task interface automatically moves to the next Task in the user's list.
During file import	

6. Fill out the **Events** tab as described in the following table:

Field	Description
Subscription events	Use the list controls to add or remove the subscriptions events that create notifications.

7. Click **Save**.

3.10.1 Default User Settings for Checkout and View Location Environment Variables

When using the OpenText Documentum CM applet, the user settings define checkout and view locations. These locations contain references to operating system environment variables from the client system. The variable names enclosed in braces ("{" and "}") are replaced with the user profile when files are viewed or edited. For example, <*{USERPROFILE}*>\Documentum\checkout.

Note: If the user is still in the same session after changing the path in the environment variables, the document will be checked out to the previously configured location. Close the browser to reflect the changes made in the environment variables.

3.10.2 Restricting Folder Location Access in the Properties Dialog

The combo-picker control on the Properties dialog is a dual trigger control that accepts the standard assist attributes and a new *restrict* attribute. Using *restrict* attribute, you can restrict the folder location access and other folder operations. You must export, edit, and re-import the property page to add this new control.

For example, if you are using the dictionary assist to populate the list with UNC paths. Use the Browse button of the combo-picker to open the folder selection dialog with the selected path. If *restrict* is set to startpath, the folder selection dialog restricts the folder selection to the start path or below the hierarchy only. Select a path in the folder selection dialog and click **OK** to update the text field with the new selected path.

Use the following settings while editing the property page to restrict the folder selection to the startpath:

```
<combo-picker id="title" assistance_dictionary="UNCPaths"
assistance_type="dictionary" asynchronous="false" control="true"
label_en="Folder Path (title)" restrict="startpath"/>
```

When *restrict* is set to startpath, the following changes are applied to the standard folder selection dialog:

1. **Up One Level** icon is disabled (except when below the root)
2. **Create New Folder** icon is disabled
3. Toolbars are hidden (home, recent, desktop, and so on)
4. Folder selection available only below the start path

When *restrict* is set to none or not specified, the full folder is accessible.

3.10.3 Configuring Value Assistance Fields

You can set fields in property pages to act as assistance for filling out other fields.

For example, you can have a **State** list field and a **Facility** list field. With this example, an end user can select a state, which populates the **Facility** field with a list of facilities specific to the selected state.

This helps an end user to find a facility within a state.

1. To use DQL assistance:

   a. Set the **Type** of the property object receiving assistance to **DQL**.

   b. Type a filtering query using '$value(*<the control ID of the assisting object>*)'

   For example, during configuration of the **State** field, you can enter selected_state as the **Control id**. During configuration of the **Facility** field, select **DQL** as the **Type**, then type as the query select facility_name from facilities where state='$value(selected_state)'

   The following table shows a summary of the fields:

Assisting property object (State)	Assisted property object (Facility)
Control id: selected_state	Type: **DQL**
	DQL Query: select facility_name from facilities where state='$value(*<selected_state>*)'

   If the user selects **Maryland** in the **State** field, the DQL query is run. The DQL query filters all facilities with the state value matching **Maryland** and populates the **Facility** field with a selectable list.

   Append enable (return_top *<maximum number of results>*) to limit the result set of the query to the indicated value. Use this parameter to avoid performance problems associated with large result sets.

2. To use Taxonomy assistance:

   a. In the property object being used as assistance, select **Taxonomy**, fill out the respective fields, then select the level of the taxonomy to be used in the assisted field in **Next Property**.

   b. In the property object receiving assistance, select **Taxonomy** and select the **Level** to match the **Next Property** field selected previously.

   This example uses the taxonomy **Facilities by state**, which has the hierarchy of **Country** > **State** > **Facility**.

   During configuration of the **State** field, select **Taxonomy**, select the taxonomy **Facilities by state**, select the level **State**, and select in **Next Property** the subtype

Facilities. During configuration of the **Facility** field, select the taxonomy **Facilities by state** and select the level **Facilities**.

The following table shows a summary of the fields:

Assisting property object (State)	Assisted property object (Facility)
Type: **Taxonomy**	Type: **Taxonomy**
Taxonomy: **Facilities by state**	Taxonomy: **Facilities by state**
Level: **State**	Level: **Facilities**
Next Property: **Facilities**	

If the user selects **Maryland** in the **State** field, the **Facility** field is populated by the subtypes of **Maryland** in the taxonomy.

Related Topics

"Configuring a Property Page" on page 48

"Configuring Property Conditions" on page 96

3.11 Configuring Property Conditions

You can configure the display, activation, and requirements for every property object added to a property page, as described in the following table:

Tab name	What the conditions are for
Enabled condition	Conditions configured in this tab determine whether the property object can be edited.
Visibility condition	Conditions configured in this tab determine whether the property object can be viewed.
Required condition	Conditions configured in this tab determine whether the property object is a required property. Not all objects contain a required condition tab.
Validation constraint	The regular expression configured in this tab forces a validation to be performed on a text field, password field, date fields, or a combo field/editable combo field in a grid. This tab does not appear for other property objects. Validation is performed: • When validating the dialog during creation. • While leaving each field during editing.

There are two editors for property conditions:

- Simple view editor: This editor provides a user interface for adding conditions.

- Advanced view editor: This editor provides a text field for inputting complex property conditions that are not supported in the Simple view editor.

1. On the **<tab>** tab:

   a. If you are using the simple view editor, use the list controls to add and remove conditions. If you do not set conditions, the system assumes that conditions have been met. Select a condition attribute and operator and type the condition value. For example, you can have the condition attribute `object_name`, the operator **Contains**, and the value `QA`. The property then requires the content to have `QA` in the file name. The simple conditions allow you to evaluate if 'all of' or just 'one of' the conditions are met to fulfill the rule.

      When using 'all of' it is the equivalent of having 'And' **&&** between all conditions.

      When using 'one of' it is the equivalent of having 'Or' || between all conditions.

   b. If you are using the Advanced view editor you can type in a more complex condition to be evaluated. This allows any number of conditional statements using parenthesis **( )**, And **&&**, Or || conditions.

      You must provide the attribute field names that exist on the property page. If an attribute field name is not present, then the condition will be evaluated in unexpected ways. To determine how the javascript engine evaluates the condition at run time, logging can be set to `FINE` or `FINEST`.

Supported Operators for Comparing and Evaluating Different Conditions	Example
Parenthesis to control order evaluation**( )**	`(((<cond1>) && (<cond2>)) || <cond3>) || (<cond4> && <cond5>)`
And **&&**	
OR||	

Supported Operators for Evaluating Conditions	Examples
contains	`getValue(<attribute>).contains(<'value>')`
notContains	`getValue(<attribute>).notContains('<value>')`
equals	`getValue(<attribute>).equals('<value>')`
notEquals	`getValue(<attribute>).notEquals('<value>')`
isGreaterOrEqualsThan	`getValue(<attribute>).isGreaterOrEqualsThan('<value>')`

Supported Operators for Evaluating Conditions	Examples
isGreaterThan	getValue(<*attribute*>).isGreaterThan('<*value*>')
isLessOrEqualsThan	getValue(<*attribute*>).isLessOrEqualsThan('<*value*>')
isLessThan	getValue(<*attribute*>).isLessThan('<*value*>')
is	getValue(<*attribute*>).is('<*value*>')
isNot	getValue(<*attribute*>).isNot('<*value*>')
beginWith	getValue(<*attribute*>).beginWith('<*value*>')
endWith	(getValue(<*attribute*>).endWith('<*value*>')

- getValue and Operators are case sensitive.
- '<*value*>' is enclosed in single quotes and '<*value*>' can be a constant string or refer to another * attribute that exists on the property page. For example: getValue(<*attrName*>).operator('getValue(<*attrName2*>)')
- Date attributes will be resolved using the format as configured for the user.
- White spaces are ignored everywhere except between getValue().operator()

c. Fill out the form as described in the following table:

Field	Description
Enabled only for group	Select the content subtype for which the property is enabled.
Reinitialize when this control is disabled	Select to reset the attribute value when the content is deactivated.
Creation mode	Select to use this property element when content is being created.
Edit mode	Select to use this property element when the properties page of content is being edited.
Import mode	Select to use this property element when content is being imported.

2. On the **Visibility condition** tab:

   a. Use the list controls to add and remove conditions. If you do not set conditions, the system considers that conditions have been met.

If you are using the simple view editor, select a condition attribute and operator and type the condition value. For example, you can have the condition attribute `object_name`, the operator `contains`, and the value `QA`. The property then requires the content to have QA in the file name.

If you are using the advanced view editor, type your condition.

b. Fill out the form as described in the following table:

Field	Description
Visible only for group	Select the content subtype for which the property is shown.
Reinitialize when this control is disabled	Select to reset the attribute value when the content is deactivated.
Reserved space	Select to keep the space allotted for the property element even if the element is set to invisible.
Creation mode	Select to use this property element when content is created.
Edit mode	Select to use this property element when the properties page of content is edited.
Import mode	Select to use this property element when content is imported.

3. On the **Required condition** tab:

a. Use the list controls to add and remove conditions. If you do not set conditions, the system considers that conditions have been met.

If you are using the simple view editor, select a condition attribute and operator and type the condition value. For example, you can have the condition attribute `object_name`, the operator `contains`, and the value `QA`. The property then requires the content to have QA in the file name.

If you are using the advanced view editor, type your condition.

b. Fill out the form as described in the following table:

Field	Description
Required only for group	Select the group for which filling out this property element is required.
Creation mode	Select to use this property element when content is being created.
Edit mode	Select to use this property element when the properties page of content is being edited.
Import mode	Select to use this property element when content is being imported.

4. On the **Validation constraint tab**, set a validation constraint:

   a. If you are adding a combo field or an editable combo field to a grid, you can select the **Unique values only** checkbox to ensure that a user is forced to pick a unique value from each control. Their selection is removed from subsequent fields so that no two fields can contain the same value. This functionality can be used in combination with the regular expression logic outlined below.

   b. Configure the regular expression:

      - Text property validation accepts Javascript regular expressions.
      - Date validation accepts two types of constraints (`equals`, `notEquals`, `Greater`, `Lesser`, `LesserOREquals`, `GreaterOREquals`) when comparing to other date fields, or `$TODAY` when comparing to the current date. The other field must be available in the dialog box.

   c. Type a **Message** to show when validation fails. This message is appended to the default message:

      `'attr' is invalid.`

      For example, if a user is validating a phone number for an attribute `pnumber` with the label `Phone Number`, and a message `Please use the format (xxx) xxx-xxxx`. The resulting display to the user will be: `Phone Number is invalid. Please use the format (xxx) xxx-xxxx`.

      Note: 'attr' is a keyword that is replaced with the attribute name. You cannot use this value in your own text.

5. Click **Save**.

Related Topics

“Configuring a Property Page” on page 48

“Configuring Value Assistance Fields” on page 95

3.12 Configuring Virtual Document Templates

1. Navigate to **Go to** > **VD Templates** from the menu bar.

2. Click **New** to create a virtual document template.

   If you want to create a child virtual document template that inherits the properties of an existing virtual document template, select a virtual document template and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Label <*language*>	Type a label.
Description	Type a description.
Applications	Add or remove the applications to which this virtual document template applies. For example, adding the `QA` application would cause the virtual document template to only apply to matching quality assurance cases.
Asynchronous build	Select to create the virtual document in asynchronous mode.

4. Click **+** to add an operation. All applicable operations for each child of the VD are applied:

   a. To create a virtual document with copied root and components, select **Copy** as the **Operation**, then fill out the properties as described in the following table:

Field	Description
Qualification DQL	Type a single DQL query to filter the source components. For example, `dm_document where a_status = 'Draft'`
New object type	Select the object type of the newly created component.
Inherit content	Select to inherit the content of the component.
Default values	Select a default values template.
Component inheritance	Select an inheritance template for the properties of the VD children.
Root inheritance	Select an inheritance template for the properties of the VD root.
Version label	Type the version label for the copied output.
Default lifecycle	Select a default lifecycle to set the starting status of the copied output.

Note: When users create a new Virtual Document using a Virtual Document template:

- The children will inherit from the template according to the Component inheritance configuration you selected in the Virtual Document template.
- The children will inherit from the template according to the Root inheritance configuration you selected in the Virtual Document template.

If there are conflicts, Root inheritance values override Component inheritance values. Any time you are creating a new document if you want inheritance to be used for the new parent or root document, then you need to use the inheritance properties as shown in the creation profile.

To link the source component to the new virtual document, select **Link** as the **Operation**, then fill out the properties as described in the following table:

Field	Description
Qualification DQL	Type a single DQL query to filter content.
Binding rule	Type the version number or version label of the content to link. The option allows you to force a specific version in the new virtual document to display. If a matching version is not found, the current version of the source component displays.

b. To exclude the content in the VD, select **Exclude** as the **Operation**, then type a single DQL query to filter content. You can only exclude copy or link operations.

5. Click **Save**.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

3.13 Configuring Linked Documents for Creating or Importing Content

1. Navigate to **Go to** > **Linked document** from the menu bar.
2. Click **New** to create a document link.

   If you want to create a child document link that inherits the properties of an existing document link, select a document link and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this document link applies. For example, adding the `QA` application would cause the document link to only apply to matching quality assurance cases.

4. Add a document link by clicking **Add linked document**.
5. For each row of the table:

   a. Select the link type. The following table describes the three available types:

Type	Description
Relation	The document link is derived from a relation. Select a relation name and type.
VD and Relation	The document link is derived from a virtual document and a relation. Select a relation name and type.
VD	The document link is derived from a virtual document.

   b. Select the creation profile for the content being created.

c. Type the message shown before creating the document.

d. Select one of the following:

- **Required** to require the end user to configure a linked document.
- **Silent creation** to create the linked document as a background process.

  Always select a default value template when using silent creation to avoid property inheritance errors.

If you select **Required** or leave both options unselected, end users see the message typed in step 5c.

6. Click **Save**.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

3.14 Configuring Checkout

1. Navigate to **Go to** > **Checkout** from the menu bar.
2. Click **New** to create a checkout setting.

   If you want to create a child checkout setting that inherits the properties of an existing checkout setting, select a checkout setting and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this checkout setting applies. For example, adding the `QA` application would cause the checkout setting to only apply to matching quality assurance cases.
Allow checkout	Select to allow content checkout.
Warning message	Select to show a warning message before checkout.
Warning message <*language*>	Type a warning message.

4. Configure checkout settings for XML applications:

Field	Description
Checkout range	Select the scope.
Checkout support documents as well	Select to checkout supporting documents included in the DocApp.

5. Click **Save**.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Security" on page 111

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

3.15 Configuring Smart View Preset Profiles

You can configure presets in the Smart View client that provide process shortcuts to users who create or import the same kind of content on a regular basis. With these presets, users can skip several standard decision points and save time. Presets are shown in the creation wizard **Select type** screen and can be added as individual preset tiles on the Landing Page or actions in the **+** menu.

A preset will be matrix evaluated by user group. In the runtime, the first matrix evaluated preset will be returned.

In the client user interface, the top eight qualified creation profiles for an operation will be shown. For a preset, the applicable creation profile will be calculated as per the following logic:

The system retrieves the creation profiles in the following priority order:

- A standard creation profile if the **Folder properties used for creation** match.
- An extended creation profile if standard creation profiles that match the configuration property are not found.
- A standard creation profile, if an extended creation profile does not exist.

The top five qualified creation profiles will be displayed for the user as Recent presets.

To configure presets:

1. Navigate to **Go to > Smart View Preset Profiles** from the menu bar.
2. Click **New** to create a Preset Profile.

   Note: If you want to create a child preset profile that inherits the properties of an existing profile, select a profile and click **Create from**.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this configuration applies. For example, adding the `QA` application would cause the configuration to only apply to matching quality assurance cases.
Label <*language*>	Type a label.
Disable custom profile	Select to disable the ability to use custom profiles in the Smart View client during the creation process.
Disable recently accessed	Select to disable the ability to use recently accessed profiles in the Smart View client during the creation process.
Profiles	Select the creation profile or extended creation profile that applies to this preset. Some profiles feature sub-artifacts that might also need to be selected.
Available for	Displays which contexts these presets will be made available. This is determined by the creation profile information you chose in the Profiles section.
Move Profile Up/Down Add Profile Delete Profile	Use these controls to add or delete multiple presets in your Preset Profile, and to arrange the order that the presets will appear in the Smart View user interface.
User can edit	Select to allow the user to edit the preset in the Smart View client.
Select Type label	Type what the preset will be known as in the Smart View interface (multiple languages are available).
Preset Unique ID	Type the parameter that is used to call the preset when you are configuring a Smart View menu or landing page tile.

If you are completing a Create-type preset profile, the **Preset Creation Template** options are available that allow you to select a template for users:

Field	Description
Enable document without content	Select to turn off template selection. When a user chooses the preset profile, a document will be created without content.

Field	Description
Selection mode	Select the method for choosing the template. **List** allows you to select from a list of available templates (or import a new template), and **Query** allows you to select a template using a DQL query.

4. Click **Save**.

3.16 Configuring Content Assistance configuration

Content Assistance allows you to designate how users find content (through simple search or repository folder browsing) when attaching content to workflows, creating relationships, or adding children or inherited components to virtual documents.

1. Navigate to **Widget view** > **Content Assist** from the menu bar.
2. Click **New** to create a content assistance configuration.

   If you want to create a child workspace that inherits the properties of an existing configuration, select a configuration from the **Content Assist** list and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this configuration applies. For example, adding the `QA` application would cause the configuration to only apply to matching quality assurance cases.
Label <*language*>	Type a label.

Field	Description
List Assistance Options	Select the type of list assistance that will be available in the dialog. At least one option must be selected: • **Browse**: (default) user is restricted to browsing through the repository folder structure. • **Search**: user can perform a simple or advanced search. • **Favorites**: displays the content of the user's favorite items. **Note:** In the Smart View client, the Favorites item will not be functional unless the default Favorites tile is configured in the Landing Page xml file.

4. Click **Save**.

Chapter 4

Configuring Security

4.1 Configuring Security

Note: Security configurations do not apply to Smart View workspaces or their contents.

1. Navigate to **Go to** > **Security** from the menu bar.
2. Click **New** to create a security model.

   Select a security model and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix. Configuring Security Inheritance contains more information on using the Name field to configure security inheritance.
Description	Type a description.
Applications	Add or remove the applications to which this security model applies. For example, adding the `QA` application would cause the security model to only apply to matching quality assurance cases.

4. Configure levels of permissions:

   Note: Enable mandatory access control lists (**MACL**) to add additional configuration options to your permissions. For more information, see "Mandatory access control list (MACL) details" on page 114. If MACL is enabled, keywords and DQL queries described below work on all tabs.

   a. Use list controls to add, remove, and reorder permission levels.

   b. For each level, select an **Identifier** from one of the three list boxes:

   - User
   - Group
   - Property

   You can merge text and property context using the following keywords:

Keyword	Description
$dqlvalue('*<dql request>*')	Perform a DQL request. You must include the quotation marks.
'$value(*<property>*) $value(*<property>*)'	Merge two properties, for example '$value(country)$value(state)'
$value(attribute_name) $value(attribute_name)_myString MyString_$value(attribute_ name)testString	Define a dynamic group by concatenating text with attribute values, for example: $value(subject)istrator If object has subject value “Admin” then it should resolve as “Administrator”.

Note: The recommended way to manage the security configuration is via properties associated with the object in question. Using a complex DQL query may not result in expected behavior when the system is not able to compute the ACL accordingly due to the DQL complexity.

c. Use list controls to add, remove, and reorder conditions. You cannot remove the default condition.

   Conditions relate a property list box to an expression. Matching conditions is prioritized from top to bottom.

d. Toggle permissions by double-clicking on the cells as described in the following table:

Right	Description
Change State	Users can change the state of content that has been applied a lifecycle.
Change Owner	Users can change the owner of content. If the user is not the content owner or a superuser, they must also have Write permission.
Change Permit	Users can change the basic permissions of content.
Delete Object	Users can only delete the object. Grant the permission if you want to allow a user to delete without being able to read the content.
Change Folder Links	Users can create or remove links to a folder. Grant the permission if you want to allow a user to move content into a folder without having permissions to read or write within the folder.

Right	Description
Change Location	Users can move content within the repository. The user must also have Write permission to move content and Browse permission to link content. Minimum permissions required for deleting a link object are `Write` and `Change Folder Links` Extended Permissions.
Execute Proc	Users can run external procedures stored in the repository as a `dm_procedure` on certain object types.

Note: The **Conflicts** column displays validation errors on your permission entries.

5. Click **Save**.

Related Topics

"Determining Which Creation Processes to Use" on page 33

"Configuring Autonaming" on page 44

"Configuring a Property Page" on page 48

"Configuring Inheritance" on page 73

"Configuring Auto Link" on page 75

"Configuring Template Lists" on page 79

"Configuring Virtual Document Templates" on page 100

"Configuring Linked Documents for Creating or Importing Content" on page 103

"Configuring Check-in" on page 81

"Configuring Checkout" on page 105

"Configuring Uniqueness Check" on page 85

"Configuring Mass Update" on page 86

"Editing a Security Model" on page 116

"Configuring Security Inheritance" on page 117

"Understanding Minimum Security Rights" on page 115

4.1.1 Mandatory access control list (MACL) details

MACL is an additional layer of configuration for your permissions. The following table describes the purpose of each tab:

Tab	Purpose
Required groups	When users request access to content, they must be part of all group and properties listed on this tab.
Required group set	When users request access to content, they must be part of at least one group or property listed on this tab.
Permissions	Lists the levels of permissions for users, groups, and categories, as well as properties and extended permissions. The **Conflict** column lists validation warnings for the configured permission rules.
Restrictions	Lists restrictions for users, groups, and properties, as well as conditions and extended restrictions. The **Conflict** column lists validation warnings for the configured permission rules. **Important** When adding a restriction, ensure that the user, group, or role that you are restricting has a higher permission level than the restriction. If not, the restriction is ignored and only the permission entry applies.

Notes

- Dynamic ACL conflicts cannot appear in the **Conflict** columns, but do appear in Smart View logs.
- After enabling or disabling MACL options, you must restart the docbase.
- When *`macl_security_disabled`* is set to `false`, performance can be impacted because queries become more complex and permissions are checked for every sysobject subtype included in the query.

For more information, see the OpenText Documentum CM Administrator User Guide.

4.1.2 Understanding Minimum Security Rights

Users require at least a certain level of security rights to perform different content-related tasks. Use the following table to understand the minimum rights for users and superusers for creating, modifying, and copying content:

	Without folder security	With folder security			
Superuser	**User**	**Superuser**	**User**		
Creation of content	**Parent folder**	None	Browse	Write	Write
Deletion	**Parent folder**	None	Browse	Write	Write
Object	Delete	Delete	Delete	Delete	
Move	**Parent folder**	None	Browse	Write	Write
Destination folder	None	Browse	Write	Write	
Object	Write	Write	Write	Write	
Duplicate	**Parent folder**	None	Browse	Read	Read
Destination folder	None	Browse	Write	Write	
Object	Read	Read	Read	Read	

By default, the admingroup group or the system administrator's group is created during installation or upgrade and contains all Superusers in the repository. When you grant Superuser privileges to a user, you may also need to add that user to the admingroup group to enable them to run jobs or administration methods. Users belonging to the admingroup group have no inherent privileges other than those they have as Superusers. However, administrative jobs, methods, and other related objects use an ACL which restricts their use to the admingroup group. If you revoke Superuser privileges from a user, the user should also be removed from the admingroup group.

Note: Do not delete the admingroup group from the repository. The admingroup group, which includes all Superusers within the repository, is used by the jobs that make up the administration tool suite.

Related Topics

"Configuring Security" on page 111

4.1.3 Editing a Security Model

Changing an existing security model can affect newly created content as well as existing content that uses the model being changed.

1. To modify a security model and affect existing content, modify the security model and do not modify the configuration matrix.

   The system runs a job in the background to modify the ACL of all previously created content using the security model.

2. To modify a security model and avoid affecting existing content, unless you perform a manual operation:

   a. Create a copy of the security model.

   b. In the configuration matrix, unselect the previous security model and select the new security model.

   The system applies the security model to new content and to existing content when it is saved, checked in, or has a property changed.

3. To modify a security model and avoid affecting existing content:

   a. Create a copy of the security model.

   b. Create a copy of the content context used for the previous security model. Add the condition `r_creation_date > <the date of creation for the new security model>` to identify the content created for the new security model.

   c. Drag the new context before the old context in the configuration matrix.

   d. Activate the new context.

   The system applies the new model to all new content, and old content continues using the old security model. You can launch D2CoreJob using OpenText Documentum CM Administrator or run a mass update to apply the new security model to linked content.

Related Topics

"Configuring Security" on page 111

"Configuring Security Inheritance" on page 117

4.2 Configuring Security Inheritance

1. Create or configure a security model.
2. Type one of the following keywords corresponding to the set of ACLs you want inherited in the **Name** field:
   - `FOLDER` for the rights of the creation folder.
   - `USER` for the rights of the user.
   - `TYPE` for the rights of the document type.
3. Click **Save**.

Related Topics

"Configuring Security" on page 111

"Editing a Security Model" on page 116

Chapter 5

Configuring Content Creation Defaults

5.1 Understanding the Content Creation Process

End users can add content to the repository using two methods:

- Create content using a rendition format and template
- Import content from the local file directory

You need an additional configuration to allow end users to import attachments to email and messages. Configuring Importing Email Attachments contains further instructions.

You can configure actions requested of users and automated processes when content is created or imported. Automated processes can help users avoid manual content management.

When content is created or configured, you can configure:

- A creation profile with at least one document type. Depending on the user group, application, and method of creation, the end user can select a document type from a creation profile. The document type applies property pages, default values, and so on to the created content.

  If there is only one document type that the end user can select, the content creation wizard automatically selects the document type and skips the selection page. You can configure creation profiles by context in this way to streamline the content creation process for end users.

  For example, create a creation profile for the quality assurance writers group available for creation and import, then create a document type for drafts. You can then set configuration elements such as the quality assurance property page, a default value template, and so on.

  Configuring Creation Profiles contains further instructions.

- A default value template to set initial property values. Default value templates set values to properties during content creation when the creation profile using the template is used. The end user can then edit the property page.

  For example, create a default value template that sets the keywords quality assurance and Draft, uses the quality assurance property page, and enters the content into the quality assurance documentation workflow. Default values can be set for properties of the new object during the creation process even if they are not present on the properties page.

  Configuring Default Value Templates contains further instructions.

- Configuration components and contexts to apply to the content. You can configure content-related configuration components such as property pages, permissions, inheritance, automatic file renaming, and so on. When the end user saves the property page during creation, the system checks for context matches and runs the applicable configuration elements on the content.

For example, create configuration elements such as a quality assurance Draft security and autonaming of the file to conform with quality assurance documents. Create a context that applies for all dm_document content for quality assurance with the a_status Draft. After enabling the corresponding cell in the configuration matrix, whenever the end user creates a quality assurance document with the Draft status, the system applies the configuration elements.

Determining Which Creation Processes to Use contains further information about creation-related configuration components.

Related Topics

"Configuring Default Values Templates" on page 125

"Configuring Creation Profiles" on page 120

"Configuring Importing Email Attachments" on page 128

"Configuring Folder Structure Import" on page 129

"Configuring XML Import" on page 132

5.2 Configuring Creation Profiles

Before configuring creation profiles, configure the following components:

- Dictionary (required)
- Taxonomy
- Context (required)
- Inheritance
- Default value template
- Autonaming
- Autolinking
- Lifecycle
- Workflow

Note: Ensure that any sub-component of a creation profile belonging to a specific application also belongs to the application. There is no constraint on sub-component selection based on application. When you design a creation profile, you can choose a dictionary/property page/default value template/ inheritance/lifecycle that is not part of the application creation profile, which might cause unexpected results.

1. To configure:

- A non-context-based creation profile, navigate to **Creation** > **Creation profile** from the menu bar.
- A context-based creation profile, navigate to **Go to** > **Extended creation profile** from the menu bar. An extended creation profile must be mapped to a context for the content type `dm_folder`.

Note: When an end user selects a folder, OpenText Documentum CM retrieves the creation profiles in the following priority order:

- A standard creation profile if the “Folder properties used for creation” match.
- An extended creation profile if standard creation profiles that match the configuration property are not found.
- A standard creation profile, if an extended creation profile does not exist.

2. Click **New** to create a creation profile.

   If you want to create a child creation profile that inherits the properties of an existing creation profile, select a creation profile and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Source content	Type a name. The name appears in the configuration profile for extended creation profiles.
Description	Type a description.
Label	Type a label.

Field	Description
Available for	Select the creation method for which the profile is eligible. This is applicable only for a regular creation profile. • Select **Create** to allow end users to control the behavior of a creation profile during content creation but not during content import. • Select **Import** to allow end users to control the behavior of a creation profile during content import but not during content creation. • Select **Linked Documents** to allow end users to control the behavior of linked documents during the content creation and import process. • Select **Single repository paste** to allow end users to control the copy/paste behavior within a single repository. When enabling **Single repository paste** in a creation profile: – Copy and paste opens the content import wizard, which follows the same steps as adding new content. – Cut and paste moves the content directly. • Select **Multi repository paste** to allow end users to control the copy/paste behavior between multiple repositories. **Note:** If the creation profile does not exist for a content during a single or multi repository copy and paste operation, and the doctype of the copied object exists in the target repository, the object is copied as is without any changes. If the doctype of the copied object does not exist, then the content import wizard appears, and you need to follow the same steps as for importing a new content.
Users group	Select the user groups for which the creation profile applies. This is applicable only for a regular creation profile.

Field	Description
Folder properties used for creation	Select one or more properties. The creation profile for folders only displays when the list properties match the **Name** of the creation profile. Leave this field blank if you do not want to restrict the creation profile by any property. This is applicable only for a regular creation profile. For example, you can name a creation profile `Project A` and type the `keywords` property for this field. The end user does not receive the option to select this creation profile unless the content being edited contains `Project A` as a keyword.
Skip edit content step	Select to skip the content editing step after importing the content.
Hide inheritance tab	Select to hide the inheritance options from the creation profile. If you hide the inheritance tab, end users cannot modify inheritance settings during creation and import.
Properties inheritance	Select to enable inheritance of content properties.
Content inheritance	Select to enable inheritance of parent content. **Note:** Content inheritance only applies when first creating a new or first version of a document.
Virtual Document structure inheritance	Select to enable inheritance of virtual document structure. **Notes** • The Virtual Document inheritance checkbox is not yet available for use in the Smart View clients add or create file functionality. • Virtual Document inheritance only applies when first creating a new or first version of a document.
Block inheritance modification	Select to prohibit the user from modifying the source content for inheritance through a file selector in the creation profile.

4. Add or remove dictionaries and properties to the creation profile.

5. Use the list controls to add and remove document types and fill out the fields as described in the following table:

   a. Fill out the default fields as described in the following table:

Field	Description
Property	Select a property to which the value selected from the dictionary is saved.
Values	Select the values from the dictionary to which the document type relates.
Type	Select the content type.
Property pages	Select the property page applied to the content.
Version	Type the starting version number of the new content. **Note:** For creation profiles that allow paste operation, you can set the version to blank. In such cases, the source object version is retained for the newly pasted object.
Inheritance	Select the inheritance rule applied to the content.
Default values template	Select the default value template applied to the content.
Lifecycle	Select a lifecycle to which the content is entered. The lifecycle sets the status of the content to the initial state, entry conditions are skipped, and, if the lifecycle is set to **Execute actions on start**, the first action is run.
Workflow	Select a workflow that the content runs after the create or import process.

   b. Click **Add column** to add a column to the document types.

6. For extended creation profiles, toggle the configuration for contexts in the configuration profile.

7. Click **Save**.

8. Navigate to **Tools > Refresh Cache** to ensure that your changes take effect on the client.

Related Topics

"Understanding the Content Creation Process" on page 119

"Configuring Default Values Templates" on page 125

"Configuring Importing Email Attachments" on page 128

"Configuring Folder Structure Import" on page 129

"Configuring XML Import" on page 132

5.2.1 Creation profiles in Smart View workspaces

In a Smart View workspace, the standard and extended creation profiles that display match the document types assigned in Admin Console to the folders in the workspace type's templates. If no document type is assigned to a folder, the default creation profiles display based on the priority order, as described in "Configuring Creation Profiles" on page 120. For more information about assigning document types to a folder in a workspace type template, see section 3.3 "Add templates to a workspace type" in *OpenText Documentum Content Management - Workspaces Administration Help (EDCADC-H-ABW)*.

Important
Do not configure a standard or extended creation profile with `dm_bws` or its subtypes. Smart View workspaces cannot be created using a creation profile. This is an incorrect configuration and can cause errors.

5.3 Configuring Default Values Templates

You can use default value templates to define the default values assigned to properties when content is created.

1. Navigate to **Creation** > **Default value templates** from the menu bar.
2. Click **New** to create a default value template.

   Select a default value template and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name.
Description	Type a description.
Applications	Add or remove the applications to which this default value template applies. For example, adding the `QA` application would cause the default value template to only apply to matching quality assurance cases.

4. Select a **Property page** if you are setting default values for a query form and you want to restrict the **Properties** selected below to only the property attributes

available in the selected property page. This allows you to pick **Do not link to Property** attributes and provide a default value.

5. Click **+** to add properties and their default values:

   a. Select a property in the **Properties** list box.

      Note: If you select a `QueryFormList` property, ensure that it is set correctly with a dictionary or DQL that will return a value you want to be set by the default value template.

   b. Type the default value for the selected property n the **Default values** field.

      The keywords you can use in the default values field are as described in the following table:

Keyword	Description
`$FILENAME`	The file name. During content export, you can store the file name into properties other than the `object_name` by using this keyword. During content import, the file name of the imported file is stored in the property corresponding to the `$FILENAME` value in the template.
`$USER`	The connected user.
`$dqlvalue("<DQL>")`	A DQL query.
`$TODAY`	The current date.
`$NOW`	The current date and hour.
`$ALIAS`	The alias.

Note: The Smart View client's creation process features an **Apply to the below files** option that allows the user to copy properties from the first file in a batch to the remaining files. In this case, the Default value template will have lower priority than the copied value.

While performing **Apply to the below files**, only the `object_name` property will be excluded. The rest of the properties will be copied from the current file to the below files including Title and Subject. So the auto-naming configuration will be honored with the provided value on the properties page.

The Classic client uses expression substitution for the Default value template in the properties page. For example : `[filename]`. In this example, the expression value (Default value template) is substituted with the actual value while creating the document.

The Smart View client uses actual value instead of expression substitution on the properties page. Example: `f1.txt`. In this example, the Default value template configuration is honored while rendering the edit properties page.

6. Click **Save**.

Related Topics

"Understanding the Content Creation Process" on page 119

"Configuring Creation Profiles" on page 120

"Configuring Importing Email Attachments" on page 128

"Configuring Folder Structure Import" on page 129

"Configuring XML Import" on page 132

5.4 Configuring Linked Documents for Importing Emails

If an end user imports an email without any attachments, the creation wizard skips the linked document steps.

1. Navigate to **Go to** > **Linked document** from the menu bar.
2. Click **New** to create a document link.

   If you want to create a child document link that inherits the properties of an existing document link, select a document link and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this document link applies. For example, adding the `QA` application would cause the document link to only apply to matching quality assurance cases.

4. Add a document link by clicking **Add linked document**.
5. For each row of the table:

   a. Select the link type. The following table describes the three available types:

Type	Description
Relation	The document link is derived from a relation. Select a relation name and type. If the linked documentation configuration is used by an end for importing an email, the end users can only select the first configured relation.
VD and Relation	The document link is derived from a virtual document and a relation. Select a relation name and type.
VD	The document link is derived from a virtual document.

b. Select the creation profile for the content being created.

c. Type the message shown before creating the document.

d. Select one of the following:

- **Required** to require the end user to configure a linked document.
- **Silent creation** to create the linked document as a background process.

  Always select a default value template when using silent creation to avoid property inheritance errors.

If you select **Required** or leave both options unselected, the message typed in step 5c displays.

6. Click **Save**.

5.5 Configuring Importing Email Attachments

You can configure content import such that when an end user imports an email with an attachment, the attachment is processed and saved to the repository as a rendition. The content types accepted are .eml or Outlook formats.

1. Navigate to **Creation** > **Mail attachments** from the menu bar.
2. Fill out the form as described in the following table:

Field	Description
Import attachments	Select to enable import of attachments. When selected, the user interface shows a checkbox asking users if they want to import the attachments.
Keep initial mail format	Select to retain the format of the attachment.

Field	Description
Make a rendition	Select to export the email as a rendition. • **Keep attachments in rendition**: select to include email attachments in the rendition. • **Rendition format**: select the file format to save the email.

3. Use the list controls for **Excluded attachment document format** to add or remove accepted file formats.

4. Click **Save**.

Note: If sensitivity labels are configured and the user uploads a .msg file with sensitivity labels, a rendition of the .msg file is not created.

Related Topics

“Understanding the Content Creation Process” on page 119

“Configuring Default Values Templates” on page 125

“Configuring Creation Profiles” on page 120

“Configuring Folder Structure Import” on page 129

“Configuring XML Import” on page 132

5.6 Configuring Folder Structure Import

You can import a folder and its subfolders and content, and configure the folder type, security, and default autolink rules for the folders.

1. Navigate to **Go to** > **Folder structure import** from the menu bar.

2. Click **New** to create a folder structure import setting.

   Select a folder structure import setting and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.

Field	Description
Applications	Add or remove the applications to which this folder structure import setting applies. For example, adding the `QA` application would cause the folder structure import setting to only apply to matching quality assurance cases.
Label	Type a label name to appear in the Folder Import dialog.

4. Set import settings as described in the following table:

Field	Description
Root folder type	Select the folder type applied to the root folder.
Root folder autolink	(Optional) Select the autolink rule you want to use for placing the folder.
Sub-folder type	Select the folder type applied to subfolders.
Sub-folder structure security	Select the security configuration you want used on the root folder and all subfolders. **Note:** The security template for all files loads from the configuration matrix after it is evaluated based on the user. If this does not provide a security template, then the default ACL applies, as defined in OpenText Documentum CM Administrator.
Creation Profile	Select the content type of the imported folder.
Convert folder structure to virtual document automatically after import	Select to automatically initiate a virtual document conversion after folder structure import.

5. Click **Save**.

Related Topics

"Understanding the Content Creation Process" on page 119

"Configuring Default Values Templates" on page 125

"Configuring Creation Profiles" on page 120

"Configuring Importing Email Attachments" on page 128

"Configuring XML Import" on page 132

5.7 Configuring Folder Structure Conversion

You can configure to convert a folder structure in a repository into a virtual document.

1. Navigate to **Go to** > **Folder Structure conversion** from the menu bar.
2. Click **New** to create a folder structure conversion setting.

   If you want to create a child folder structure conversion setting that inherits the properties of an existing folder structure conversion setting, select a folder structure conversion setting and click **Create from**.Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this folder structure import setting applies. For example, adding the `QA` application would cause the folder structure import setting to only apply to matching quality assurance cases.

4. Set **Conversion settings** as described in the following table:

Field	Description
Create profile	Select the content type of the virtual document root folder.
Child node type	Select the type of the subfolders in the new virtual document.
Child node default values	Select to automatically set the default values for the subfolders.
Root node inheritance	Select the inheritance configuration to define the properties you want to inherit from the root folder.
Source folder inheritance	Select the inheritance configuration to define the properties you want to inherit from the source folder. **Note:** Source folder inheritance takes precedence over virtual document root inheritance.

5. Click **Save**.

5.8 Configuring XML Import

Use XML import for the integration of:

- Large amounts of data, such as migrating a content management system or importing a content dump from an external application.
- Content on a routine basis, such as the daily retrieval of content produced by another application.

For example, you can configure an external application to save content to a specific folder. You can then configure the XML import component to scan the folder on a regular basis. The system automatically imports content found into the repository.

1. Create a folder on the Documentum CM Server to store content imported into the repository.
2. In OpenText Documentum CM Administrator, configure D2JobImportMassCreate similarly to the example described in the following table:

If you are using a creation profile	If you are not using a creation profile
`-folder` *<C:/d66_02/import>*	`-folder` *<C:/d66_02/import>*
`-naming` *<false>*	`-naming` *<false>*
`-createconfig` *<matrix_name>*	
`-args` *<argument_name>*	

List of D2 Jobs contains further information.

3. In OpenText Documentum CM Administrator, configure D2JobCoreDeQueue similarly to the following example:

`-scan_queue true`

`-naming true`

`-create true`

List of D2 Jobs contains further information.

4. Create an XML file to function as a metadata template for the content imported from the folder. The XML must contain:

   - A root element: `<?xml version="1.0" encoding="UTF-8"?>`
   - Attribute values: `<d2_import_xml>` followed by properties, such as `title`, `authors`, and `keywords`. The specific properties used in the XML file depend on the configuration of the content type and property page. You can designate a property as a repeating property by adding the `<value/>` tag.
   - The same file name as the content to be imported. To import only the properties of content, name the XML file *<filename of the content>*`-meta.xml`

- If you did not use a creation profile using `-createconfig`, specify the `r_object_type` to determine the context and configuration components to use. If you do not set an `r_object_type` nor a `-createconfig`, the system does not perform the import operation.

The D2JobImportMassCreate scans the folder according to the configured interval, and D2JobCoreDeQueue browses the d2c_mass_create_queue to apply job parameters and D2CoreMethod to import the content into the repository.

Related Topics

“Understanding the Content Creation Process” on page 119

“Configuring Default Values Templates” on page 125

“Configuring Creation Profiles” on page 120

“Configuring Importing Email Attachments” on page 128

“Configuring Folder Structure Import” on page 129

Chapter 6

Configuring Content Processes Components

6.1 Understanding Lifecycles

A lifecycle is a sequence of states and is used to:

- Denote the status of the content as it passes from one business protocol to another.
- Apply and change content retention and markup policies.

The following table describes an example lifecycle:

State	Conditions	Next state
Draft	None	Reviewed
Reviewed	Content is in Draft state r_lock_owner is empty	Draft Approved: requests electronic signature
Approved	Content is in Reviewed state Content is a PDF	None

In this example:

- Content can progress from Draft to Reviewed to Approved. Content can progress by repeatedly going to Draft after Reviewed.
- If you progress from Reviewed PDF to Approved, the system confirms the conditions have been met. If this progression requests an electronic signature, a prompt appears. When you electronically sign the approval, the system progresses the state to Approved.
- If you try to progress a spreadsheet instead of a PDF, the system rejects the progression request because one of the conditions was not met.

For each state, you can configure:

- Entry conditions for the state, such as conditions based on properties, permissions, and linked content.
- Actions to run when moving to the next step, such as filling a property, creating a version, applying security, and changing the document state.
- Next states and transitions between states, such as popup dialog boxes, messages, and menu labels.
- Whether the user can perform a lifecycle state change on multiple documents

There are two types of lifecycle state changes:

Individual processing. Allows the user to capture input for all files individually in sequence before submitting the lifecycle state change.

Bulk processing: Allows the user to capture input for all files before submitting the lifecycle state change.

For more details see section 4.4.2 “Multi-file lifecycle support” in *OpenText Documentum Content Management - Smart View User Guide (EDCCL-UGD)*

Related Topics

“Understanding Workflows” on page 146

“Configuring a Lifecycle” on page 136

“Configuring a Lifecycle Batch” on page 145

6.2 Configuring a Lifecycle

1. Navigate to **Go to** > **Lifecycle** from the menu bar.
2. Click **New** to create a lifecycle.

   If you want to create a child lifecycle that inherits the properties of an existing lifecycle, select a lifecycle and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Select a lifecycle.
Description	Type a description.
Applications	Add or remove the applications to which this lifecycle applies. For example, adding the `QA` application would cause the lifecycle to only apply to matching quality assurance cases.
Execute actions on start	Select to run actions related to the lifecycle when an end user starts the lifecycle.
Alternate attribute for state (default is a_status)	Select a property to control the state of the content.

4. Click **Add lifecycle state** to add a state to the lifecycle:
   a. Type a name for the lifecycle state.
   b. Select **Start state** to initialize the content with the selected state.
   c. Select **Direct state** if all states of the lifecycle can access the state.
5. Configure the entry conditions for each **Lifecycle state**:

a. Use the list controls to add, remove, and reorder entry conditions.

b. Fill out the form as described in the following table depending on the entry condition selected:

Entry condition	Description
Condition VD children	Select the lifecycle state that is required of all virtual document components, type a DQL request to filter virtual document components, and type a warning message that appears when the condition is not met. **Note:** This entry condition test is not used to verify the state of a linked document or virtual document children, but to test each condition you have configured into entry condition of target state.
Condition check by Method	Select the method to run for verifying that the condition is met, type arguments and parameters required by the method, and type a warning message that appears when the condition is not met.

Entry condition	Description
Condition on group	Select whether the group was **Saved in a property** or whether the end user types the group name using **Free input**. **Saved in a property** extracts the group name from the property of the document associated with the lifecycle operation. When you create a document, <*group_name*> is defaulted to the group name of the creator. When you perform the transition, the extracted group name will be verified against the allowed groups in the dictionary alias. Note that the property is not restricted to <*group_name*>. You can create a new property where you save your group name while creating a document but it has to be the same property name you specify in the configuration. **Free input** is more restrictive. You have to specify a fixed group name under **Property containing group or group name** which will be verified against the dictionary alias values. Type the property that contains the group or group name, select the dictionary and alias, and type a warning message that appears when the condition is not met.
Condition on linked document	Select the relation type for the linked documents, select the lifecycle and state to which the linked documents are to be set, and type a warning message that appears when the condition is not met.
Condition on property	Select the property, operation, and value for a property-based condition, and type a warning message that appears when the condition is not met.
Condition on rendition	Select the required rendition format, and type a warning message that appears when the condition is not met.
Condition on uniqueness	Select the configured uniqueness check that must be passed, and type a warning message that appears when the condition is not met.
Condition permission	Select the minimum permission level required, and type a warning message that appears when the condition is not met.

6. Click a state from **Lifecycle state** to configure the actions of each lifecycle state:

    a. Click **Add action** and select an action.

    b. Fill out the form as described in the following table depending on the action selected:

Action	Description
Action on VD children	Select the lifecycle state you want to apply to all virtual document components and select actions you want ignored.
Apply autolink	No entry required. The highest priority configuration in the configuration matrix is the default.
Apply autonaming	No entry required. The highest priority configuration in the configuration matrix is the default. **Note:** For any lifecycle that calls the autonaming action, the autonaming configuration used (by the configuration/matrix evaluation) must have the **Generate the document name for each property modification** setting enabled in the auto-naming config in order for the autonaming rule to run.
Apply method	Select the method to run against the content and type the arguments and parameters required. Check the **Fetch Current** checkbox to ensure that the new object's current version is fetched so that the next action operation can pick up the right version to work on.

Action	Description
Apply parameters	No entry generally required. The highest priority configuration in the configuration matrix is the default. **Apply Parameters** can also be run if parameters need to be applied at a specific step. For example, if you need parameters from properties set before calling a server method, or before O2 method implementation and rendition creation. **Note:** The **Apply Parameters** action does not work for controls that are not linked to properties (for example, attributes), and < *$value*> expressions do not correctly resolve for these controls.
Apply security	No entry required. The highest priority configuration in the configuration matrix is the default.
Change linked document state	Select the relation type for the linked documents, select the relation direction between the parent and child documents, and select the lifecycle and state to which the linked documents are set.
Change other versions state	Select the state you want other versions to be set to and type a DQL query to further specify the versions.
Copy repeating property	Select the repeating property you want to copy, the copy operation, the target property, and select **Remove duplicate values** if you want to remove duplicate values.
Make version	Select the type of versioning you want to perform.
Manage distribution	Select the distribution configuration and action you want to use.
Mark	Type a version label to **with value**.
Remove other versions	Select **Authorize destruction of annotated document** to enable removal of annotated content and type a DQL query to further specify the versions.
Rendition request	No entry required.
Send email	Select the mailing list configuration.

Action	Description
Send in workflow	Select the workflow configuration and type a name and note for the workflow being entered.
Set property	Select the property and type the value to assign.
Set repeating property	Select the repeating property and the set operation, type the value you want to assign, then select **Remove duplicate values** to remove duplicate values.
Snapshot	Type a title.
Unmark	Type a version label to **with value**.
Work offline	No entry required. This action will allow the file to be kept locked and checked-out.
Delete this action	No entry required.

Note: Only the following tokens are applicable for the **Set Properties**, **Set Property**, **Set Repeating**, **Apply Method** argument, and **Launch workflow** argument actions in the lifecycle:

- $USER
- $TODAY
- $NOW
- $value
- $alias
- $dqlvalue

c. Use **Action up** and **Action down** to reorder the lines in the table of actions.

7. Click a state from **Lifecycle state** to configure next states and transition condition:

a. Click **Add transition** and select a next state.

b. Fill out the form as described in the following table:

Field	Description
Type	Select the type of transition to perform during this state change.
Action to perform	Select the type of action to perform. • Checkin: check in the content.

Field	Description
Dialog box	Select the dialog box to display as a popup form for the end user to fill.
Trigger event for transition	Select the event that serves as a trigger for automatic progression of the lifecycle. • Check in: lifecycle progresses when checked in. • Import: lifecycle progresses when the file is imported.
Menu label <*language*>	Type a menu label that is visible when the end user opens the lifecycle context menu.
Electronic signature required	Select to require the reason with the electronic signature.
Intention required	Select to require an electronic signature to end the lifecycle state.
Intentions dictionary	Select a dictionary to enable users to select a reason from a list of options. Alternately, you can also type a custom `$value` into the field. For example: GR Approval $value(site_id) $value(some_custom_property) Reject This input would get the first workflow document (not supporting document) in the task to determine the $value then look for that dictionary config.
External e-signature confirmation message <*language*>	Add a custom confirmation message that appears in the client when a user is required to enter their OpenText Directory Services (OTDS) login and password to validate an electronic signature.
Confirmation message <*language*>	Type a confirmation message.

c. Configure Smart View Lifecycle Options:

Allow bulk transitions	If the value is checked, multiple files are sent to the target state at the same time and are transitioned as a single batch. Lifecycle state change in bulk is enabled on the following conditions: • The selected documents are pointing to the same or different lifecycle configuration and the lifecycle state name is the same.

Skip UI processing	Enabled when **Allow bulk transitions** is checked. If the value is checked, the user has the ability to submit the lifecycle state change request asynchronously. The processing and outcome of the lifecycle state change occurs in the background and is not reflected in the UI.
Request e-sign for each document	Enabled when **Electronic signature required** is checked. If the value is checked, the user has the ability to perform an electronic signature for each document sequentially so the user can provide different intentions for each document.
Summary mail	When **Allow bulk transitions** and **Skip UI processing** are enabled, a mailing list may be selected from the **Summary mail** drop-down list to send success or failure summary information about the asynchronous state change. This mailing list is configured using the `$d2_email_core_content` parameter (see "Configuring Summary Mail for Asynchronous Multi-file Lifecycle State Changes" on page 191. By default, the email is sent to the performing user but additional recipients can be added in the mailing list configuration. **Note:** For the feature to work, you must configure the appropriate email server parameters as shown in "Configuring the Mail Server" on page 186. The performing user must have a valid email address in their user account as must any other recipients configured in the mailing list configuration.

d. Configure the entry conditions for each transition.

8. Configure the transition conditions for each **Lifecycle state**:

   a. Select a **Next state** to set a transition condition from the selected lifecycle state to the selected next state.

   b. Use the list controls to add, remove, and reorder transition conditions.

   c. Fill out the form as described in the following table depending on the transition condition selected:

Entry condition	Description
Condition VD children	Select the lifecycle state that is required of all virtual document components, type a DQL request to filter virtual document components, and type a warning message that appears when the condition is not met.
Condition check by Method	Select the method to run for verifying that the condition is met, type arguments and parameters required by the method, and type a warning message that appears when the condition is not met.
Condition on group	Select whether the group was **Saved in a property** or whether the end user types the group name using **Free input**. Type the property that contains the group or group name, select the dictionary and alias, and type a warning message that appears when the condition is not met.
Condition on linked document	Select the relation type for the linked documents, select the lifecycle and state to which the linked documents are to be set, and type a warning message that appears when the condition is not met.
Condition on property	Select the property, operation, and value for a property-based condition, and type a warning message that appears when the condition is not met.
Condition on rendition	Select the required rendition format, and type a warning message that appears when the condition is not met.
Condition on uniqueness	Select the configured uniqueness check that must be passed, and type a warning message that appears when the condition is not met.
Condition permission	Select the minimum permission level required, and type a warning message that appears when the condition is not met.

9. Click **Save**.

Related Topics

“Understanding Lifecycles” on page 135

“Configuring a Lifecycle Batch” on page 145

6.3 Configuring a Lifecycle Batch

You can configure lifecycle batches to control lifecycles for all content assigned to a context simultaneously. For example, you can create the Draft, Review, and Approved folders, and configure a context for each folder. Using autolink, you can configure content created by an end user to appear in the Draft folder. You can configure content in the Draft folder to automatically pass through the Review and Approved folders.

Lifecycle batches use the D2JobLifecycleBatch job. List of D2 Jobs contains further information.

1. Navigate to **Go to** > **Lifecycle batch** from the menu bar.
2. Click **New** to create a lifecycle batch.

   If you want to create a child lifecycle batch that inherits the properties of an existing lifecycle batch, select a lifecycle batch and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Select a lifecycle batch.
Description	Type a description.
Applications	Add or remove the applications to which this lifecycle batch applies. For example, adding the `QA` application would cause the lifecycle batch to only apply to matching quality assurance cases.
Lifecycle	Select the lifecycle you want to use.
Target state	Select the target state.
Transition type	Select the type of transition performed.

4. Click **+** to add and configure properties modified by the lifecycle batch.

   Configure each property as described in the following table:

Field	Description
Property	Select a property from the list box.
Value	Type the value of the property.

5. Click **Save**.

Related Topics

"Understanding Lifecycles" on page 135

"Configuring a Lifecycle" on page 136

6.4 Understanding Workflows

A workflow is a formalized version of a business process which has been broken down into individual tasks. Users use workflows to:

- Apply business processes into the lifecycle of a content.
- Support commenting, tracking of workflow progress, management of various participants, and attaching of supplementary content.

Related Topics

"Understanding Lifecycles" on page 135

"Configuring a Workflow" on page 149

6.4.1 Using Workflow Manager or Process Builder

You can use Workflow Manager to create and configure workflows, then install them as templates into the repository. Client configuration recognizes the templates, allowing you to select the templates when creating a workflow for use by end users.

Note: If the Workflow templates are created using Process Builder, the workflows are processed using Process Engine so you need to install Process Engine. If the Workflow templates are created using Workflow Manager, the workflows are processed using the Workflow Engine of the Documentum CM Server so you do not need to install Process Engine.

In order to use a workflow template, the following conditions must be met:

- There must be a task.
- There must be a name for each task.
- There must be flow.
- There must be aliases for the recipients of the workflow.
- There must be triggers and transitions.
- There must be delegation if necessary.
- There must only be a single package from start to end.

Workflows work with the following performer types that are available in Documentum CM Server:

- All users in group
- Multiple sequential performers
- Previous activity's performer

- Single user from group
- Specific user
- Workflow supervisor
- Repository Owner

The following performer type is not supported:

- Work queue
- Some users from a group

Note: Workflows do not manage the **Some users from group** alias. You must configure workflows to use either the **All users from group** alias or the **Single user from group** alias. When you use the **All users from group** alias, a temporary group for the workflow instance is created and the list of recipients can be specified.

6.4.2 Using Advanced Workflow Designer or XCP Process Designer

You can use Advanced Workflow Designer (DAW) or XCP Process Designer to create and configure workflows and then install them as templates into the repository. The client configuration recognizes the templates, allowing you to select the templates when creating a workflow for use by end users.

Notes

- If the templates are migrated from Workflow Manager or Process builder to Advanced Workflow Designer, you can use the migration utility to migrate the Workflow Config and map to the migrated templates.
- If the Workflow templates are created or migrated to Advanced Workflow Designer or XCP Process Designer, the workflows are processed using Process Engine so you need to install Process Engine.

In order to use a workflow template, the following conditions must be met:

- There must be a task.
- There must be a name for each task.
- There must be flow.
- There must be aliases for the recipients of the workflow.
- There must be triggers and transitions.
- There must be delegation if necessary.
- There must only be a single package from start to end.

Workflows work with the following performer types that are available in Documentum CM Server:

- Individual users and group members at the same time.
- Individual users and the first group member to accept the task at the same time.
- Individual user or group member with fewest tasks.
- Individual users and the first group member to accept the task, in sequence.
- A user or group from process data.
- A user or group from a parameter.
- A user or group from an alias set.
- A workflow supervisor.
- A repository owner.

Note: Workflows built with Workflow Manager are not compatible with workflows built with Advanced Workflow Designer. Advanced Workflow Designer has a migration tool for legacy Workflow Manager/Process Builder workflows, but it does not migrate the client configuration information. To migrate configurations, you can use a utility called `d2workflowconfigmigration` which can be found in the `\D2-Config\utils\d2workflowconfigmigration` folder after extracting `D2-Config.war`. For instructions on using the tool, see the `D2 Workflow Config Migration Instructions.pdf` document in `D2-Config\utils`.

Once a workflow is migrated, you do not need to migrate it again.

6.4.3 Using Workflow Designer

Workflow Designer is used to create workflows for OpenText Documentum CM clients such as Smart View and Webtop. A workflow defines a business process that can be used repeatedly. You can create and run multiple instances of a workflow process at runtime. A workflow consists of one or more activities linked together by flows. An activity represents a task that a user must perform on the business process objects to pass the document to the next activity in the workflow. For example, reviewing a document, checking the document into the repository, or approving the document. A flow connects two activities and defines the sequence of activities in a business process. It also defines the packages that are exchanged between the activities. A Package is an object (Sysobject, Content, Folder, or Cabinet) that is processed during an activity and then passed to the next activity in the business process.

See *OpenText Documentum Content Management - Workflow Designer Deployment Guide (EDCPKL-IWD)* for more information about deploying Workflow Designer in the OpenText Documentum CM environment.

See *OpenText Documentum Content Management - Workflow Designer User Guide (EDCPKL-AWF)* for more information on usage of the Workflow Designer.

Notes

- Workflows that are built using Workflow Manager are not compatible with workflows that are built using Workflow Designer. Workflow Designer has a migration tool for legacy Workflow Manager built workflows, but it does not migrate the client configuration information. To migrate OpenText Documentum CM configurations, you can use a utility called `d2workflowconfigmigration` which can be found in `D2-Config\utils`. For additional instructions, see the `D2 Workflow Config Migration Instructions.pdf` document in `D2-Config\utils`.
- You must install workflows in Workflow Designer for them to appear as templates in client configuration.

6.4.3.1 Using process variables

Process variables are a Workflow Designer feature that allow you to track data or information through the tasks of a workflow.

For more information on process variables and how to implement in the workflow designer, see *OpenText Documentum Content Management Workflow Designer - Online Help (EDCPKL-H-AWF)*.

6.5 Configuring a Workflow

If you use Workflow Manager to create a workflow template, you can use OpenText Documentum CM to configure the workflow template to allow end users to perform business processes on content. When configuring the workflow template in Workflow Manager:

- Create short aliases in the list of performers.
- Do not use spaces in the list of performers.
- Select **Optional** in the **Version** field if you want to be able to send any version of a document into a workflow, or select **Current** to allow only the current version of a document.

  Note: The Workflow Overflow widget only shows the workflows launched on the selected document (single version of the document) with **Optional** version selected in the workflow template.

OpenText Documentum CM creates groups in real time to represent performers.

1. Navigate to **Go to** > **Workflow** from the menu bar.
2. Click **New** to create a workflow.

   If you want to create a child workflow that inherits the properties of an existing workflow, select a workflow and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix. For best performance, the workflow name should not exceed 64 characters.
Workflow label <*language*>	Type a label that appears when the end user access the workflow context menu.
Description	Type a description.
Applications	Add or remove the applications to which this workflow applies. For example, adding the `QA` application would cause the workflow to only apply to matching quality assurance cases.
Workflow template	Select a workflow template. The version field displays the version number of the template and **CURRENT** if it is the latest version.
View	Click to see a popup dialog with a graphical diagram of the workflow template.

4. Click **Add condition** and select entry conditions for the workflow.

   Fill out the form as described in the following table depending on the entry condition selected:

Entry condition	Description
Condition on property	Select the property, operation, and value for a property-based condition, and type a warning message that appears when the condition is not met.
Condition permission	Select the minimum permission level required, and type a warning message that appears when the condition is not met.
Condition on group	Select whether the group was **Saved in a property** or whether the end user types the group name using **Free input**. Type the property that contains the group or group name, select the dictionary and alias, and type a warning message that appears when the condition is not met.
Condition on rendition	Select the required rendition format, and type a warning message that appears when the condition is not met.

Entry condition	Description
Condition check by Method	Select the method to verify that the condition is met, type arguments and parameters required by the method, and type a warning message that appears when the condition is not met.
Condition on uniqueness	Select the configured uniqueness check that must be passed, and type a warning message that appears when the condition is not met.
Condition on file type	Select the allowed formats for the workflow and type a warning message that appears when an unsupported file type is added.
Condition on file size	Select the maximum file size allowed for the workflow document and type a warning message that appears when the file size exceeds the limit.

a. Use **Condition up** and **Condition down** to reorder conditions.

5. Configure workflow abort options as described in the following table:

Field	Description
Lifecycle	Select the lifecycle to use for an aborted workflow.
Target state	Select the final state of the content.
Apply on attachments	Select to apply the state to all associated content.
Allow deletion of workflow history	Select to allow an option for workflow supervisor or users with superuser or sysadmin privileges to delete workflow history when they abort a workflow. This setting appears in the Smart View client only.

Note: During a workflow abort, the entry conditions set for the lifecycle state are not evaluated when the state is changed.

6. If necessary, configure workflow options that are specific to Smart View and mobile as described in the following table:

Field	Description
Key package	If a process is configured with multiple packages, select which package to use as the key package. The key package is mandatory, it is the file considered by entry conditions, and it populates property values. If a user initiates a workflow without a file that meets the key package entry conditions, they cannot start the workflow until they do. If a user initiates a workflow with multiple files, the key package entry conditions determine which file is used as the key package. If more than one file meets the key package entry conditions, the first selected file is used. **Important** If the selected key package is deleted from the process in the workflow designer, this field becomes blank and you must select a new package. If you do not, the user cannot initiate this workflow.
Initiate activity	For most workflow configurations, use the default **Manual** setting. If you are configuring an inbound activity workflow, set this to **Manual/Inbound** to display workflow activities for workflows initiated by BPS in Smart View. Only one workflow configuration can be set to **Manual/Inbound** per process. For more information about inbound activity workflows, see *OpenText Documentum Content Management Advanced Workflow - User Guide (EDCPKLPR-H-UGD)*.
Disable schedule workflow options	Select to remove the **Schedule** section controls (including planning and follow-up options) from the Launch Workflow panel in the Smart View client.
Disable notes	Select to remove the **Notes** tab from the Task Details view in the Smart View client.

Field	Description
Possible workflow supervisors	By default the user who initiates the workflow is the supervisor. If you enable this setting, the workflow supervisor can be changed by qualified users. The list of possible supervisors shown in the client is based on the Dictionary or DQL query you use in the **Query** field.

7. If necessary, configure workflow filters.

   Workflow filters allow users to limit what items appear in the workflow overview pane. If you do not define filters, the user can only use default filter options. If you define more than one workflow filter condition, all conditions are evaluated with "AND" logic.

   These filters only appear in Smart View and mobile. The following table describes the filter options:

Field	Description
Workflow Filter Label	Enter the name of the filter. Enter names for other languages by expanding the filter name list. Click **+** to add another filter. Click **x** to delete a filter. **Note:** Filter labels cannot end with a special character such as * or &.
Workflow Filter Condition	Configure the filter by selecting an attribute, operator, and a value. These three elements determine which workflows appear in a filter. To the right of the condition options, click **+** to add condition. Click **x** to delete a condition.
Condition attribute	Select which attribute the condition uses to filter workflows. Here are the options: • Task Name • Task Subject • Task State • Activity Type

Field	Description
Condition operator	Select the operator for the filter. An operator is like the kind of filter. The operator and the value you enter determines which workflows appear in a filter. For example, the **contains** operator lists workflows that include the value you enter, or the **greater than** operator lists workflows with values higher than the value you enter.
Condition value	The value used by the condition to determine if a workflow should appear in a filter.

8. If necessary, configure workflow package options as described in the following table:

Field	Description
Disable supporting files	Select to remove the **Supporting Files** tab from the Task Details view in the Smart View client or disable the **Supporting Documents** field when the user launches a workflow in the Classic client.
Maximum supporting files	Enter the maximum number of supporting files that can be sent along with a workflow.
Maximum working files	Enter the maximum number of working files that can be sent along with a workflow.

9. Select a task from the **Task configuration** list. Manual tasks are shown with a white background, while tasks that are automatically processed are shown with a gray background.

   Note: The **Initiate** activity only appears for processes made with Advanced Workflow Designer or Workflow Designer. It does not appear for processes created with Workflow manager.

   a. Fill out the **Task parameters** tab as described in the following table:

Field	Description
Subject <*language*>	Type a subject to appear in the Client user task inbox. Use '$value(<*property name*>)' to refer to properties.

Field	Description
Message <*language*>	Type a message to appear in the Client Task details widget. Use `'$value(<property name>)'` to refer to the properties of the master document.
Category <*language*>	Type a category with which to associate the task when an end user sorts the Workflow task browser widget by category. You can create a subcategory using `/` to designate a category path. For example, to create the subcategory `Reviewing` under the category `Review Process`, type `Review Process/Reviewing`. You can configure autolinks to organize the user task inbox.
Priority	Select a priority level for the task as it appears for end users in their list of tasks.
Replace "Accept" with Label <*language*>	Type a custom task action label. The custom label replaces the default **Accept** label.
Replace "Reject" with Label <*language*>	Type a custom task action label. The custom label replaces the default **Reject** label.
Display tasks in workflow manager	Select to show the task in the workflow manager for end users.
Manual acquisition	Select to require users to manually acquire a task. This is necessary when the manual task is set in Workflow Manager to use a single user from a group. When not selected, tasks are automatically considered being performed by the assigned user. Requiring users to manually acquire tasks adds a level of work to the end user in performing tasks, but lets you see when an end user has seen and acknowledged the task.
Check lifecycle consistency on task end	Select to verify lifecycle entry conditions before an end user accepts or rejects a task.

Field	Description
Always display dialog box	Select to improve the speed of the validation process by hiding dialog boxes such as the comment box of a task.
Check lock state of primary documents	Select to restrict end users from accepting, rejecting, or delegating a task when a primary document is locked.
Check lock state of supporting documents	Select to restrict end users from accepting, rejecting, or delegating a task when a supporting document is locked.
Mandatory comments in case of reject	Select to require end users to provide comments when rejecting a task.
Mandatory comments in case of forward	Select to require end users to provide comments when forwarding a task.
Property page in case of reject	Select a property page to show when an end user rejects a task. The option enables end users to supply additional information for the content. This information is saved with the attachment content only if you add auto activity after manual activity, select the properties in the **Workflow property** list provided in client configuration workflow configuration, and select **Apply on attachment** to apply the changed attribute on attachments.
Property page in case of forward	Select a property page to show when an end user forwards a task. The option enables end users to supply additional information for the content. This information is saved with the attachment content only if you add auto activity after manual activity, select the properties in the **Workflow property** list provided in workflow configuration, and select **Apply on attachment** to apply the changed attribute on attachments.
Audit information add-on	Configure a manual tasks audit for electronic signatures by typing additional audit information. You can use the DQL keywords `$value` and `$dqlvalue` for this field.

Field	Description
Confirmation message *<language>*	Add a custom confirmation message that appears in the client when a user is required to enter their OpenText Directory Services (OTDS) login and password to validate an electronic signature.
Audit task on acquisition	Select to audit when users acquire a task.
Audit choice of next task	Select to audit the progression of the task.
Don't send notification for this task	Select to avoid sending an email notification to the workflow performer.

Notes

- You can use HTML in the body of the email.
- You can use `$keywords` in tasks. For more information, see “Using $keywords with workflows” on page 167.

b. Fill out the **Participants management** tab as described in the following table:

Field	Description
Task participant	Type the alias of the group performing the task.
Can add or remove a task when the activity is running	Select to allow current performers to add or remove participants while a workflow is in progress. **Notes** • This option is only available for dynamic tasks and only works for the *<All Users in Group>* performer type in the activity definition. • In the docbase `server.ini` file, ensure that *`mail_notification`* is set to `True` and *`performers`* has the email ID configured to receive dynamic tasks,
Participants available and Participants modifiable	Use the list controls to configure available and modifiable participant lists.
Required on Forward	Select if a participant must be assigned to the task if it is Forwarded.

Field	Description
Required on Reject	Select if a participant must be assigned to the task if it is Rejected.
Allow delegation notes	Select if a task note can be entered when assigning a task.
Require delegation notes	Select if a task note must be entered when assigning a task.
Type	Select the user list to propose delegation. You can use **DQL** or **Dictionary** to filter the list of users. You can use the DQL keyword `$value`. **Note:** Group delegations are not supported. Only user delegation is supported. The `D2Delegation` job only retrieves tasks which are assigned to a single user in `dmi_queue_item`. If **Single user of the group** is configured in a Workflow template, delegation cannot be completed. You can achieve comparable behavior if you configure an Activity like this example: i. Set **All users in group** as performer. ii. Complete the activity when one performer completes the task in Transition. If you use a DQL query, you can click Validate to check your query. These are the following validation results: • Valid: Query border turns green. Your query will work. • Warning: Query border turns yellow and a **Warning** message appears. OpenText Documentum CM cannot validate your query because it is too complex. You must test your query in OpenText Documentum CM. • Invalid: Query border turns red and a **Invalid Syntax** message appears. Your query will not work.

Note: In the Workflow Performers widget, selection of a query such as `select * from dm_user` will not work because, in this case the first attribute of the schema is `r_object_id`. The following supported statements can be used instead:

```
select user_name as name from dm_user
```

```
select user_name as name from dm_user enable (return_top 3)
```

```
select user_name from dm_user where r_is_group = false and user_
name like '$value(owner_name)' order by 1
```

c. Fill out the **External task** tab as described in the following table:

Field	Description
Define external task	Select to send an email to an external participant.
Attach document	Select to allow attachment of documents.
Add workflow notes in the message	Select to allow addition of workflow notes to the message.
Add workflow scheme (HTML/VML)	Select to allow addition of the workflow scheme in HTML/VML format.
Add rendition with format	Click **Browse**, then use the list controls to modify the list of formats.
Checkin options	Select **Add as annotation** to check in the change as a content annotation. Select **Silent checkin minor version** to check in the change as a minor version and as a background process. Select **Silent checkin same version** to check in the change without version control and as a background process. Select **No checkin** to not check in the change.
Log entry <*language*>	Type a log entry.
Mandatory attached file in case of reject	Select to require end users to attach a file if they reject the task.
Add reply as a workflow note	Select to add the mail message as a workflow note.

d. Fill out the **Task follow up** tab as described in the following table:

Field	Description
Task duration in days	Type an integer for the total number of days allotted for the task.

Add or delete followup procedures using **Add notification** and **Delete this notification**, and configure the process as described in the following table:

Field	Description
Remaining days before	Type an integer for the number of days remaining in the task. When the number of days remaining until the task deadline reaches the entered number, the follow up process is triggered. You can type a negative integer to send a notification when the deadline has passed.
Notification recipients	Select one or more of the following users to receive the notification: • **Supervisor** • **Task owner** • **Other** If you select **Other**, you can use the DQL keywords `$value`, `$alias`, and `$dqlvalue`.
Action	You can select an action to perform on the workflow when the follow up is sent.
Subject <*language*>	Type the subject of the email.
Message <*language*>	Type the message body of the email.

Notes

- You can use HTML in the body of the email.
- You can use `$keywords` in notifications. For more information, see “Using $keywords with workflows” on page 167.

e. Fill out the **Reject follow up** tab as described in the following table:

Field	Description
Activities available	Select an activity in the list and click > to enable a follow up activity for this task.

Field	Description
Reject Follow up Activities	Select an activity in the list and click **<** to disable the activity as a follow up. Reject follow-up allows for an alternate icon to be displayed when a reject action happens for the indicated workflow action. When the task is sent back to the configured users, the rejected task that is configured via the follow-up function will now have an clipboard icon with a red 'return' arrow in their task widget. This is used to indicate to the user that the task was previously approved, but has been rejected back to their queue.

f. Fill out the **Signature** tab as described in the following table:

Field	Description
Electronic signature	
Electronic signature required	Select to require an electronic signature to progress the task.
Required for	Select which progression case for which you want an electronic signature.
Request a signature for each document	Select if you want a signature for each attached document.
Intentions dictionary for forward Intentions dictionary for reject	Select a dictionary to enable users to select a reason from a list of options. Alternately, you can also type a custom `$value` into the field. For example: GR Approval $value(site_id) $value(some_custom_property) Reject This input would get the first workflow document (not supporting document) in the task to determine the $value then look for that dictionary config.
Intention required for forward Intention required for reject	Select to require the reason when forwarding or rejecting the electronic signature.
Audit information add-on	Configure a manual tasks audit for electronic signatures by typing additional audit information. You can use the DQL keywords `$value` and `$dqlvalue` for this field.

Field	Description
Confirmation message <*language*>	Add a custom confirmation message that appears in the client when a user is required to enter their OpenText Directory Services (OTDS) login and password to validate an electronic signature.
Digital signature	
Digital signature required	Select to enable the use of digital signature providers.
Digital signature provider	Select the digital signature provider. Only installed providers are listed.
Send document as	Select the format to send the document: Native format, PDF Configuration Export, or PDF Configuration Print. If you select PDF Configuration Export or PDF Configuration Print, you must select a PDF configuration.
Define signing order	Enable to allow the user to control the order the digital signature requests are sent. If disabled, all signature requests are sent at the same time.
Determine signee values	Defines the list of potential signers. Select either **Dictionary** or **DQL**. If you select **Dictionary**, select a dictionary. If you select **DQL**, enter a valid DQL statement to define the list of users. For example, `select user_name from dm_user`.
Subject	Enter the email subject for the digital signature email notification. If you do not enter subject text, a default subject appears.
Message	Enter the email message for the digital signature email notification. If you do not enter text, default text appears.
Allow external users	Allow external users to be sent digital signature requests. If you do not select this option, only users defined by **Determine signee values** can be sent digital signature requests.

Field	Description
Version signed document as	Determine how to update the document after signatures: same version, minor revision, or major revision.
Signed log file location	Determine where to save the digital signature log file. The available options are defined by auto linking configuration. For more information, see “Configuring Auto Link” on page 75.
Signed log file permission	Determine the permissions for the digital signature log file. The available options are defined by security configuration. For more information, see “Configuring Security” on page 111.

g. If available, open the **Process variables** tab and set the status of each process variable:

- **Hidden**: The process variable will not appear in the task page, when initiating a workflow, or when accepting or rejecting a task action.
- **Read-only**: The process variable appears as read-only in the task page, when initiating a workflow, or when accepting or rejecting a task action.
- **Mandatory**: The process variable appears as a mandatory field that the user must complete when initiating a workflow or when accepting or rejecting a task action.

Note: If you configure the workflow with performer selection, property pages, or electronic signatures, then the process variables tab displays with these configurations in sequence.

10. To configure a system task, fill out the form as described in the following table:

Field	Description
Label <*language*>	Type a label.
Transition type	Select the transition type.
Lifecycle	Select the lifecycle to populate the target state. The lifecycle evaluation will be based on the matrix mapping.
Target state	Select the lifecycle state for the content.
Apply on attachments	Select to add attachments during this process.

Field	Description
Add workflow properties	Click **Browse**, then use < and > to modify the list of additional properties. For example, you can configure a workflow task to require users to fill out a date property page. The task can be performed at different stages of the workflow by different user groups, such as the Product team, the Testing team, and the Management team. To keep and show the dates input by the different teams, configure the date property as a workflow property. If you do not configure the property, you cannot display the dates. **Note:** The values entered by the user in a manual task are stored in the workflow, not in the document. If you want to send values from the workflow to the document, you must use a lifecycle change state automatic method and use the **Add Workflow Properties** feature in the workflow configuration.
Display task in workflow manager	Select to show the task in the workflow manager.

11. Configure the performer list:

    a. Configure the properties page using the editor. The buttons function the same as in the property page. For each participant, fill out the fields shown for the selected list control as described in the following table:

Field	Description
Participant	Select the participant being modified in this object. The system can manage only one participant alias per task. **Note:** Aliases with spaces in their names are not recognized. It is not desirable to use a participant alias which is the same as the document's attribute. The participant segregation does not work properly in the workflow property page, when the DQL used to populate a synchronous list control, contains any dependency on the document's attribute using $value evaluation.

Field	Description
Default value	Type default values, such as user names.
Linked attribute	Select a property. OpenText Documentum CM sets the value of the property as the list of performers listed under the alias. You can use this property with a security template to ensure that workflow performers have access rights to the documents in the workflow. If the workflow is set to allow multiple performers and the list of performers contains groups, OpenText Documentum CM adds the individual users contained in the groups when the workflow is started. As a result, if a user is removed from a group after the workflow is started, OpenText Documentum CM does not update the list of workflow performers and the linked attribute. You must manually remove users from the list of performers to update the linked attribute.
Load asynchronously	Select to load values on click and filter as opposed to when the dialog box is opened. Use the option when the list contains a large number of values.
Label <*language*>	Type a label.
Read-only on startup	Select to disable the field in the workflow startup page.
Required on workflow start-up	Select to make the field required when the end user starts the workflow.
Type	Select DQL or Dictionary to filter the list and select or type the condition.
Rows number	Type the number of rows the field shows.
Required values count	Type the minimum number of values for accepting the performer list.
Maximum values count	Type the maximum number of values for accepting the performer list.
Sort list	Select to sort the list alphabetically.
Enable values import export	Select to enable the import and export of the list of performers in .xls format.
Popup width and Popup height	Type the size of the popup dialog.

b. To prevent workflow supervisors from changing the list of performers:

   i. In **Participant's Structure**, select **Read-only on startup** for each performer.

   ii. On the **Participants management** tab, clear **Can add or remove a task when the activity is running** and remove the performer from **Participants modifiable**.

   OpenText Documentum CM requires default values when workflow supervisors cannot change performers.

   c. Use the list controls to reorder tasks.

12. Configure the process variables for the process and all corresponding tasks:

   a. Configure the properties page using the editor. The buttons function the same as in the property page. For each process variable, fill out the fields shown for the selected list control as described in the following table:

Field	Description
Process variable	Map the selected process variable.
Default value	Not configurable. Configured in Workflow Designer.
Label <*language*>	Type a label.
Type	Only available for some process variable types. Select how to populate process variable options. Can be: **DQL**, **Dictionary**, or **Taxonomy**. If you select **DQL**, enter a DQL query in the **Query** field to control which options appear for the process variable. If you select, **Dictionary** or **Taxonomy**, select which dictionary or taxonomy with which to populate the process variable options.

   b. Some process variables controls cannot be added through Workflow Designer, such as a Password or Radio button. To add these types of controls, use the editor buttons to add a control and then map it to an existing process variable. You might need to delete a mapped process variable control first to free it to be added to the new control.

   c. Use the list controls to reorder the process variables.

   By default, each process variable is mapped to a default field, but you can configure them here. The following are details for each process variable type:

Process variable type	Single or double variable	Default mapping	Supported mapping
String, Integer, Float	Single	Text	Text field, Combo list, Editable combo list, Password, Text area, Checkbox, Radio button, Hyper link
Boolean	Single	Boolean	Boolean, radio button
Date	Single	Date	Date
String, Integer, Float, Boolean, Date	Multiple	Editable list	List, Editable list

Important
Smart View client cannot validate the field type when the user enters data. As a result, if the user enters the wrong data type, the workflow process internally fails and an exception is thrown. This applies to floating points, incorrect date types, and integers. As a workaround, the administrator can use labels to inform the user the data type to enter.

13. Click **Save**.

Note: To configure the `end_date` and `d2c_workflow_tracker` objects of a workflow, you need to run the keystore utility to configure property files. See *OpenText Documentum Content Management - Client Installation Guide (EDCCL-IGD)*.

Related Topics

"Understanding Workflows" on page 146

"Using $keywords with workflows" on page 167

6.5.1 Using $keywords with workflows

You can use `$keywords` to automatically populate text in workflow tasks and notifications. Here is a list of `$tokens` you can use and details about using them:

- Use the `$value` keyword to use property values. For example, `'$value(<document.property_name>)'` shows the property of a document.
- Use the `$alias` keyword to use a dictionary. For example: `$value(document.a_status.dictionary_name.auto)`
- For email configuration, use `'$value(document.object_name)'` to display attributes of the packaged object. The package type is selected during the process of configuring the workflow. These attributes can be accessed using "document" as the prefix. To use this with workflow configuration, use a direct reference like `$value(<property name>)`. For example: `$value(object_name)`.

- Use '$value (tracker.<some_attribute>)' to access d2c_workflow_tracker details using tracker.<some_attribute>. If the attribute data returned has any $alias, then is also processed.
- When you use either '$value(task.subject)' or '$value(task.message)' with $value, it picks your workflow configuration's current task instance details (subject and message based on the activity name) and the subject or message strings are internally processed. During this process, inner $value and $alias are considered. At this point, $value processing is different from general $value processing of email configuration. Rules that were applied for a string in Email Configuration cannot be parsed (for example, '$value(tracker.<some_attribute>)' because the value for 'task.subject' and 'task.message' are from Workflow/Task configuration.
- $foreachworkingdoc works for all the workflow related notifications which display the **Working Documents** of the workflow. For example:

  $foreachworkingdoc{$value(object_name) - $value(title) - $value(a_status)|\n}
- If a user's configured default *local* in **Preferences** is empty, $value(task.subject) and $value(task.message) are evaluated based on the fallback locale configured in the D2-JMS.proeprties file. For example: localeFallback=en
- Use $WorkflowNotes.PERFORMER to add the notes of the most recent performer or performers of the task.
- Use $WorkflowNotes.ALL to add all notes added by the supervisor and performer to date, newest first.

Chapter 7

Configuring search

7.1 Configuring Global Search Settings

Configure the properties of the search engine, type, and results to apply to every search performed in the client.

Notes

- When <*DQL Fulltext*> is enabled in search configuration, DFC generates the DQL, and the CS Query Plugin translates the DQL into an <*xQuery*>. When <*Documentum xPlore*> is enabled in search configuration, DFC generates the <*xQuery*> directly. This means that <*Order by Score*> is not calculated the same when using DQL with and without <*xPlore*> for fulltext search.
- Running quick and advanced searched with DQL can be slower than expected if all three of the following conditions are present:
  - Multiple types are included in the search criteria.
  - A custom attribute is added to the columns.
  - The search result returns a huge data set.

1. From the menu bar, navigate to **Interface** > **Search**
2. Select a dictionary as described in the following table:

Field	Description
Dictionary for object type labels	Select a dictionary to add specific labels to object types.
Alias/Locale	Select an alias or language.

3. Use the list controls to configure the **Groups allowed to create public search**. Users in the selected groups can create and save public searches.

   Note: By default, all users are allowed to create public searches. To limit the ability of creating public searches only to selected users, you must first define a group that is allowed to create public searches, add selected users to that group, and then add the group to the list of **Groups allowed to create public search**.

4. Configure the search engine as described in the following table:

Field	Description
Maximum results returned by search	Type an integer to restrict the number of maximum returned results. Type `0` to have all results shown.
Enable full text search	Select to enable full-text search. By default only simple searches are performed.
Quick search on all versions (default is CURRENT)	Select to enable quick search on all versions of content. By default, quick search is only run on current versions of content.
Indexing engine choice	Select the search engine used by OpenText Documentum CM. If you do not enable full-text searches, you do not have to select a search engine. Simple searches are performed using DQL queries and do not need an external search engine. **DQL Fulltext** is the same as **Documentum Search / xPlore Search**, except that end users cannot configure and use facets. If you use this option, end users can set the client to highlight search terms in the search results.

5. If you select a search engine other than DQL fulltext, fill out the form as described in the following table:

Field	Description
Use Documentum Search / xPlore Search for all searches (even the searches without full-text criteria)	Select to enable xPlore for all searches. You can configure the length of the search summary when configured to use this search engine.
Enable Facets	Select to enable search facets. Facets represent one or more important characteristics of an object in the OpenText Documentum CM object model. Configure facets to search large data sets without explicit queries and to avoid queries that do not return desired results. Configuring Advanced Search contains further information on how to configure properties for use as facets.
Maximum number of result by Facet	Type the number of property values retrieved by facets.

Field	Description
Limit Documentum Search / xPlore term highlighting to Dictionary values only	Select to limit term highlighting. Dictionary values are highlighted in search results, but corresponding dictionary keys (that are not identical to the search term) are not highlighted.

6. Click **Save**.

   After search results are displayed in a doclist, the default behavior for actions such as edit, check out, and cancel checkout is to perform a locate action along with the user indicated action. This results in the doc list being refreshed and focused on the document within the repository folder structure.

   To alter this behavior and allow the user to remain on a refresh view of the search results, use a menu configuration to define the menu action for Edit, Checkout, and Cancel Checkout as indicated below:

   - Set **Locate content and refresh state upon action** (default = true) to <*false*>
   - Set **Refresh state** (default = false) to <*true*>

   The refreshed view of the search results may not display the current lock icon state when xPlore full-text is used due to xPlore index configuration and latency.

   "Configuring Client Menus" on page 293 has information about creating and using menus.

Related Topics

"Configuring a Query Form Search" on page 174

"Configuring Advanced Search" on page 171

7.2 Configuring Advanced Search

Use search mapping to enable types and properties that have been defined in the repository for advanced searches. You can construct different search modules in accordance with different contexts.

Ensure the DFC query search limit (*`dfc.search.max_results_per_source`* and *`dfc.search.max_results`* in `dfc.properties`) is aligned with the *`maxResultsize`*, which can be found under **Maximum Results Filtering** section in the widget configuration or in the `D2FS.properties` file) to avoid result truncation when a facet is applied.

1. Navigate to **Go to** > **Search** from the menu bar.
2. Click **New** to create an advanced search.

   If you want to create a child advanced search that inherits the properties of an existing advanced search, select an advanced search and click **Create from**.

Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this advanced search applies. For example, adding the `QA` application would cause the advanced search to only apply to matching quality assurance cases.

4. Use the list controls to configure the list of available types for **Type**.

5. Select **Include type as search default** to set the types that will be available by default to the user for selection in the advanced search dialog box. The selected types are listed in **Default types for advanced search** list.

6. Select an available type, select a property, and use the list controls to configure the list of **Properties of selected type**.

Caution

If facet search is enabled, do not add a search type without also adding a property. If you do not add a property for a search type, an error occurs.

7. Select a property and select **Dictionary** or **DQL** to create input assistance:

   a. If you select **Dictionary**, select a dictionary from the **Dictionary** list box, then an alias or a language from the **Alias/Locale** list box.

   b. If you select **DQL**, type a DQL query in the **Query** field.

   Notes

   - DQL does not support *`$value`* configuration.
   - To ensure that a single document with multiple dates in a custom repeating date field displays as a single result in the search results:
     i. Configure `server.ini` of Documentum CM Server with `return_top_results_row_based = false` in the [SERVER_STARTUP] section.
     ii. Include `r_object_id` in the SELECT statement of your DQL query.

   c. Select **Include property as facet** to allow end users to include the property of the selected types in the facet configuration table.

The facet configuration table allows the administrator to set the display order of the facets, define the sorting of the facet values, set a default facet for search, and configure the structured facets. Using the toolbar buttons, the administrator can move the facet properties to change the display order, indent or outdent the facet properties to configure structured facets, and remove a property from facet use. The administrator can select multiple properties in the table and can simultaneously perform these operations.

The structured facets are configured by indenting a facet property and making it a child of the previously listed facet. Only one level of indenting is supported. The remaining structure is defined by ordering the indented sibling facets.

For example:

Document FormatOwner NameModified By

This structure implies, Document Format > Owner Name > Modified By, that is Category > sub-category1 > sub-category2.

Here, **Document Format** facet has a sub-category **Owner Name**, and Owner Name further has a sub-category **Modified By**.

> **Note:** If you have set up OpenText Documentum CM Content Intelligent Services (CIS), then users can enhance faceted search results using CIS taxonomies.

d. Select **Default Facet** to set a facet property as the default facet for searches.

> **Note:** In the Smart View client, default facets control which facets display by default in new advanced and quick searches. In the Classic client, default facets apply only to quick searches.

8. Click **Save**.

Related Topics

"Configuring Global Search Settings" on page 169

"Configuring a Query Form Search" on page 174

7.2.1 Improving advanced search performance

Open `D2FS.properties` and set *`ignoreSearchOrderBy`* to `true`. By default it is set to `false`.

This change ensures that the data from search is sorted by ranking. The user's specified sort column then sorts the data further.

7.3 Configuring a Query Form Search

Use query form searches to provide property templates to end users. End users can select a query form, fill out the properties, then search. The alternatives would be to perform:

- A standard search, in which end users can only input a general query.
- An advanced search, in which an end user must specify the properties they wish to use.

Note: If you perform a `select *` query, the lock symbol for a document (which indicates it is checked-out) does not appear in the query form search result.

1. Navigate to **Go to** > **Query form** from the menu bar.
2. Click **New** to create a query form.

   If you want to create a child query form that inherits the properties of an existing query form, select a query form and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this query form applies. For example, adding the `QA` application would cause the query form to only apply to matching quality assurance cases.
Label <*language*>	Type the form label.
Property page	Select the property page to display for the search from the list box. You can preview the property page by clicking the magnifying glass. **Note:** The query form does not support multiple panel property pages, hence these pages are not available in the list.

Field	Description
Default value	Select a default value template to add values to the properties. **Note:** For repeating attribute r_creation_date, if the field is non-mandatory and blank in the query form, then a default value of 01/01/1970 is used.
Ignore empty attributes	Select to remove empty attributes from DQL or search queries. When this option is checked, a specialized syntax can be used in the DQL to indicate when an attribute with no value should be excluded from the final DQL statement instead of the attribute being submitted with a NULL value: "[<optional-attribute> ? <qualifier>]" For example: select datetostring(submission_date, 'yyyy/MM/dd') as "date", object_name, product_code, submission_type, heath_authority from cd_reg_submission_info where product_code = '$value(product_code)' [health_authority ? and health_authority = '$value(health_authority)'] [from_date ? and submission_date >= date('$value(from_date.dd/MM/yyyy)', 'dd/ MM/yyyy')] [to_date ? and submission_date <= date('$value(to_date.dd/MM/yyyy)', 'dd/ MM/yyyy')] order by 1 desc
Mapping options	Select one of the following options: • **DQL query**: This option enables a DQL based query but does not support the use of facets with search results. • **Documentum Search / xPlore query**: This option enables an Advanced Search configuration that can use facets to refine the search results.

a. If you select DQL query, configure the following details:

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>DQL</td><td> Type a DQL query using the attributes of the property page. Use <code>'$value(<property name>)'</code> to retrieve the values of different properties.
Ensure that the DQL query is selective. Neither the Maximum results returned by search field in the global search configuration page nor the <code>dfc.search.max_results</code> options in <code>dfc.properties</code> affect the result size returned by query forms. End users could experience performance issues when a large number of search results are returned.
To ensure that a single document with multiple dates in a custom repeating date field displays as a single result in the search results:
i. Configure <code>server.ini</code> of Documentum CM Server with <code>return_top_results_row_based = false</code> in the [SERVER_STARTUP] section.
ii. Include <code>r_object_id</code> in the SELECT statement of your DQL query.
iii. Include <code>order by r_object_id</code>.
For query forms involving date type attributes follow the steps mentioned in the Note below.
Ensure that you include <code>r_object_id</code> or <code>a_content_type</code> in the select statement to display the icon in the query form results.
Note: Only the first column with <code>order by</code> keyword is effective. </td></tr>
</table>

Field	Description
Using repeating attribute values	You can use repeating attribute values by using the < *$repeatingvalue<repeating_attribute>>* method. For example, `Select object_name from dm_document where subject IN($repeatingvalue(authors))` **Notes** • If your DQL select clause contains any repeating attributes, and also includes a distinct function, add an `i_position` attribute in the select clause to avoid a DQL error. • If your DQL select clause contains any repeating attributes, include an attribute which identifies the objects in the select clause and in *`order by`*. For example, `select r_object_id, authors from dm_document order by r_object_id`. This method ensures object-based results return.
Add mandatory attributes in hidden mode	Select this option to retrieve properties required for content control in the search results. If you do not select this option, the search result returns an informational listing of matching content. In this state, you cannot make any changes to the retrieved content, including launching a lifecycle. If you select this option, some of the attributes included in the query may not be displayed in the results, as they are added in hidden mode. For example, `r_lock_owner` attribute. **Note:** If you want DQL search results in a Query form on a Doclist to engage with other widgets, you will need to add `r_object_id` to the query in DQL for Query form or check **Add mandatory attributes in hidden mode**.

Field	Description
Add the user column preferences	Select the option to retrieve content properties matching the column preferences of the user. If you do not select the option, added columns using properties that are not automatically retrieved for showing search results remain blank.

Note: For query forms involving date type attributes:

- Add a constraint <date attribute> is not nulldate to where clause in the DQL if <*nulldate*> values are not required in the result set. For example:

```
select r_object_id,r_modify_date from dm_document where (r_modify_date is
not nulldate) enable(return_top 10)
```

- The <*Datefloor*> function rounds a given date down to the beginning of the year, month, or day in UTC. OpenText recommends that you use “`utc`” in literal values used in equals comparison. For example, assuming both OpenText Documentum CM Application Server and Documentum CM Server are in the same GMT+5:30 time zone, the following DQL returns all records with `r_modify_date` as `25 january, 2016` in GMT +5:30 timezone:

```
select r_object_id,r_modify_date from dm_document where
datefloor(day,r_modify_date) =date('01/25/2016utc','mm/dd/yyyy')
enable(return_top 10)
```

  The *DQL Reference Guide* provides additional details for <*datefloor*> and timezone syntax.

- For the operators less than and greater than, adding timezone `utc` to the date literal changes the results as well. You can decide on the usage based on docbase timezone and business requirements. For example, assuming both OpenText Documentum CM Application Server and Documentum CM Server are in the same GMT+5:30 timezone, the following DQLs returns different results:

  Example:

```
select r_object_id,r_modify_date from dm_document where r_modify_date
>=date('02/16/2016','mm/dd/yyyy') enable(return_top 10)
```

  Includes results such as <*r_modify_date=2/16/2016 5:26:55 AM*>

  Example:

```
select r_object_id,r_modify_date from dm_document where r_modify_date
>=date('02/16/2016utc','mm/dd/yyyy') enable(return_top 10)
```

  Includes results such as <*r_modify_date=2/16/2016 5:37:57 AM*> with the addition of timezone, the second DQL returns values greater than or equal to midnight past 5:30 hours.

- If you use any date aliases such as <*’$value(r_modify_date)’*>, that do not directly specify the date literal, for example: <“2/16/2016

5:26:55”>, the system automatically translates the aliases to the UTC time (with the UTC keyword already appended). There is no requirement to convert the time to UTC time nor append the UTC keyword manually.

For example, the following will be translated properly to UTC time with UTC keyword appended:

```
select * from dm_document where r_modify_date >=
DATE('$value(r_modify_date)', 'mm/dd/yyyy') enable(return_top 10)
```

b. If you select xPlore query, configure the following details:

Field	Description
Type	Select an available object type for the search. Move the selected types to the right for use. Use the list controls to add and remove content types.
Criteria	Create search equations by using the one of the following modifiers: • `AND`: Matches content where both the search terms are found. • `OR`: Matches content where one or both terms are found. **Note:** To add another row for a property-based criteria, click **+**.
Property	The object types you select provide value assistance for the properties list box. The property attributes configured for the object types are available for selection.
Condition	For the selected property, specify a condition from the list box based on which you want to perform the search operation.

Field	Description
Value	For the selected combination of property and condition, specify the value to be searched. The search operation displays the results if the combination of property, value, and condition is satisfied. You can use the keyword `$value` (property page attribute). The value of the indicated attribute is passed from the Query Form Property page and used in the query. For example, if the query form passes the value of the 'Title' attribute to a query form value using `'$value(title)'`, the query uses that value in the search. **Note:** An xPlore query form does not support empty values, so all form values should be set as mandatory or you should select the **Ignore empty attributes** option.
Full Text	Select a search option from the list box and type the search terms you want to use for the full-text portion of the search. You can use the keyword `$value` (property page attribute). The indicated attribute is passed from the Query Form Property page and used in the query. For example, if you use the search option **Containing any words** and type the text term as `'$value(user_input_text)'`, OpenText Documentum CM fetches all the documents that contain the text you enter for the search operation. **Note:** You cannot use search-term modifiers such as `AND` and `+`.
All versions	Select if you want to search all versions of the content. If you do not select this option, only the current version is searched.
Case sensitive	Select to perform a case-sensitive search. **Note:** This feature is not supported in this release.

Field	Description
Search Path	Type the source folders to be used by the search operation. To additional folders, click **+**. To remove a selection, click **X**.
Include sub-folders	Select to include the subfolders of the selected folders. If you do not select this option, the search checks only the selected folders.
Columns	Define the columns to configure the display settings of the search results. Select the columns from the available list and move them to the right column for use.
Order by	The selected columns are displayed in the list box. Specify whether you to list the columns in the search results in the ascending or the descending order.
Properties for facet	The list of properties for facet is populated based on the object types you select. From the available list, move the properties to the right for use as facets. The selected facets automatically appear in the facet configuration table below.
Definition	You can further configure the individual property facets using these options: • **Dictionary**: If you select this option, select a dictionary from the **Dictionary** list box, and then an alias or a language from the **Alias/Locale** list box. • **DQL**: If you select this option, type a DQL query in the **Query** field. **Note:** Best practice is to include a unique attribute name and attribute type when defining a facet. The uniqueness should be at database level. For example, if you run the following query: `select distinct attr_type from dm_type where attr_name = '<attribute_name>' enable (row_based)` one row should be returned in the result.

Field	Description
Configure facet order and structure	The properties selected for using as facets are listed in the **Facet Property** column, along with their type in parenthesis. The **Sort** column provides the following options to sort a property: • Frequency: Sorts the facet values based on the number of matches in the search result. • Ascending: Sorts the facet values in the increasing order (A to Z) of their labels. • Descending: Sorts the facet values in the decreasing order (Z to A) of their labels. • Fixed: The original configured order is retained. You can use the toolbar buttons to move the facet properties to change the display order. You can configure structured facets by indenting / outdenting a facet. The indented facet becomes the child of the parent facet under which it is indented. You can select multiple properties in the table and can simultaneously perform these operations. You can also remove a property from facet use. Only one level of indenting is supported. The remaining structure is defined by ordering the indented sibling facets. **Note:** Only two levels of facets appear on mobile.

4. Add the query form to a search category so that end users can locate it for use:

   a. In the **Existing categories** browser, navigate to the folder in which you want to add the query form.

   b. Type /*<folder name>* in **Path**.

   c. Type a description for the category.

   d. You can configure the name, title, and user group permissions of a category by right-clicking the category and selecting **Modify search category**.

5. Click **Save**.

Related Topics

"Configuring Global Search Settings" on page 169

"Configuring Advanced Search" on page 171

Chapter 8

Configuring Administrator-to-User Communication

8.1 Understanding How to Communicate with Users Through OpenText Documentum CM and Email

If you have an email server to which OpenText Documentum CM connects and communicates, you can configure:

- Mailing lists that end users can use to send pre-configured batch emails.
- The ability for end users to send emails directly from the user interface instead of having to open up an email application.
- Distributions that end users can use to create an email-based distribution of content for expedited and external validation.
- Subscriptions that end users can use to receive email notifications for events, such as workflow and lifecycle tasks. An end user who creates or owns the subscription does not receive notifications even if the end user had configured to receive the subscription.

Related Topics

"Configuring the Mail Server" on page 186

"Configuring a Distribution" on page 196

"Configuring a Subscription" on page 194

"Configuring Options for Sending Emails through OpenText Documentum CM" on page 192

"Configuring Options for Sending Emails through OpenText Documentum CM" on page 192

"Configuring a Mailing List" on page 189

8.2 Configuring the Mail Server

The email server is used for sending emails.

1. Navigate to **Tools** > **Email** from the menu bar.
2. Depending on your protocol, fill out the **Email server** form as described in the following tables:

   SMTP

Field	Description
Protocol	Select **SMTP**.
Using TLS	Select to use an TLS connection.
Server	Type the server address for the email server.
Port	Type the port number for the server.
Login	Type the username for the email address to use when sending messages. **Note:** The login must have permission to send email on the mail server.
From address	Type an address to mask the sender email. If you use Office 365, this email address must be the same as what is defined by *login mail_session email_id* in client configuration.
Password	Type the password for the username. OpenText Documentum CM stores passwords as encrypted text for existing email configuration. It is recommended that you reconfigure email configurations in the client configuration.
From name	Type the name to use when sending messages.

   OAuth 2 (Microsoft 365 only)

Field	Description
Protocol	Select **oauth2**.
Tenant ID	Type the tenant ID of the organization.
Client ID	Type the client ID of the application.
Client secret	Type the secret key used by the application to request the token.

Field	Description
Proxy host	Type the proxy host if the application is behind a proxy.
Proxy port	Type proxy port number if the application is behind a proxy.
From name	Type the name to use when sending messages.
From address	Type an address to mask the sender email. This email address must have permission to send email and be the same as what is defined by *login mail_session email_id* in client configuration.

Note: The values above must be generated by you Office 365 administrator.

3. Click **Test Email**. This validates the email server details and confirms if the server is accessible. If it fails, confirm the details that you entered are correct in the client configuration.

4. Fill out the **Email reception server** form as described in the following table:

Field	Description
Protocol	Select the mail protocol used by the reception server.
Using TLS	Select to use an TLS connection.
Server	Type the server address for the email server.
Port	Type the port number for the server.

   The reception server is checked for received emails.

5. Fill out the **Accepted task account** form as described in the following table:

Field	Description
Login	Type the username for the email address to use when accepting tasks.
Address	Type an address to mask the sender email.
Password	Type the password for the username.

   When using the email reception server, OpenText Documentum CM logs into the accepted task account to accept tasks.

6. Fill out the **Rejected task account** form as described in the following table:

Field	Description
Login	Type the username for the email address to use when rejecting tasks.
Address	Type an address to mask the sender email.
Password	Type the password for the username.

When using the email reception server, OpenText Documentum CM logs into the rejected task account to reject tasks.

7. Fill out the **Administration of errors into external task inbox** form as described in the following table:

Field	Description
Send back mail on task error	Select to send mail back to the sender if the email address used by the task assignee returns an error.
Active mail history	Select to enable keeping a history of emails, including duplicate messages, in the event that a message is sent back in the above option.

8. Click **Add a notification** to add an event:

   Fill out the form for each row as described in the following table:

Field	Description
Event	Select or type the event to use from the list box. You can use audit events as a trigger for sending a notification. For example, if you want to send a notification when a workflow is started, select or type `d2_workflow_started`. **Note:** To enable d2_workflow_started, you must select the **Send notification when the workflow is started** in the **Planning /Sending of workflow** dialog box.
Subject <*language*>	Type the subject of the email.
Message <*language*>	Type the body of the email.

Notes

- You can use HTML in the body of the email.
- Emails are triggered from the D2EventSenderMailMethod in different scenarios, such as: Workflow notifications, Workflow with external task, Lifecycle Notifications, Workflow Subscriptions, Send Mail, Email.

- The system validates access to the email server and does not expect **login** and **from address** to be the same.
- Emails that are triggered by lifecycle notifications are sent via a multi-threaded process to avoid impacting the overall performance of D2LifecycleBatch. The number of threads is dynamically allocated, but you can manually define or disable the thread count by adding the following to the AppServer JVM property `-Dmail.nothreads=<x>`. `0` value disables the threading.
- You can use `$keywords` in email notifications. For more information, see “Using $keywords with workflows” on page 167.

9. Click **Save**.

Related Topics

“Understanding How to Communicate with Users Through OpenText Documentum CM and Email” on page 185

“Configuring a Distribution” on page 196

“Configuring a Subscription” on page 194

“Configuring Options for Sending Emails through OpenText Documentum CM” on page 192

“Configuring Options for Sending Emails through OpenText Documentum CM” on page 192

“Configuring a Mailing List” on page 189

8.3 Configuring a Mailing List

1. Navigate to **Go to** > **Mailing list** from the menu bar.
2. Click **New** to create a mailing list.

   If you want to create a child mailing list that inherits the properties of an existing mailing list, select a mailing list and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.

Field	Description
Applications	Add or remove the applications to which this mailing list applies. For example, adding the `QA` application would cause the mailing list to only apply to matching quality assurance cases.
Attach document	Select to allow attaching of documents.
Attach rendition with format	Click **Browse**, then use → and ← to configure the list of allowed rendition formats. The end user can only add attachments that have an allowed rendition format.

4. Click **+** to add a recipient, then select or type a recipient.

   Use the following table to understand the options shown:

Field	Description
User	Select a user from the list box.
Group	Select a user group from the list box.
Property value	Select a property to use the value of the property as the recipient.
Email address	Type an email address.

5. Fill out the content of the email as according to the following table:

   You can use the DQL keyword `'$value(`*`<property name>`*`)'`.

   The body accepts HTML.

Field	Description
Email subject *<language>*	Type the subject of the email.
Email message *<language>*	Type the body of the email.

   You can use the `$value` keyword to use property values. For example, `'$value(`*`<document.property_name>`*`)'` shows the property of a document.

   You can use the `$alias` keyword to use a dictionary.

   You can use HTML in the body of the email.

6. Click **Save**.

8.3.1 Configuring Summary Mail for Asynchronous Multi-file Lifecycle State Changes

When asynchronous state change is configured for bulk lifecycle processing (i.e., **Skip UI processing** is enabled), a mailing list may be selected to send success or failure summary information about the asynchronous state change (see **Summary mail** under “Configuring Smart View Options” in “Configuring a Lifecycle” on page 136). This mailing list is configured using the `$d2_email_core_content` parameter, which is substituted with the appropriate success/failure summary information when the email is sent. By default, the email is sent to the performing user but additional recipients can be configured.

To configure a Summary mail:

1. Create a mailing list as you would normally.
2. Enter a recognizable **Name**, e.g., **Lifecycle state change**.
3. Select **Recipients** in addition to the performing user, if desired.

   Note: Resolving a recipient with an attribute is not supported in lifecycle state change summary emails.

4. Enter an appropriate **Email subject**, e.g., **Lifecycle state change complete**.
5. Enter an appropriate Email message, making sure to include the `$d2_email_core_content` parameter enclosed in HTML tags.
6. Click **Save**.

Related Topics

“Understanding How to Communicate with Users Through OpenText Documentum CM and Email” on page 185

“Configuring the Mail Server” on page 186

“Configuring a Distribution” on page 196

“Configuring a Subscription” on page 194

“Configuring Options for Sending Emails through OpenText Documentum CM” on page 192

“Configuring Options for Sending Emails through OpenText Documentum CM” on page 192

8.4 Configuring Options for Sending Emails through OpenText Documentum CM

1. Navigate to **Go to** > **Send mail** from the menu bar.

2. Click **New** to create an email sending option.

   If you want to create a child email sending option that inherits the properties of an existing email sending option, select an email sending option and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this email sending option applies. For example, if you add the `Quality Assurance` application, OpenText Documentum CM makes the configuration available only when an end user interacts with content recognized by the `Quality Assurance` application.
Enable external recipients	Select to allow end users to add email addresses of recipients outside of the repository when sending an email through the client.
Enable attachments	Select to allow end users to add attachments when sending an email through the client. **Note:** If a user selects multiple files, the attachment option only appears if all selected documents contain some content (are not empty files), if the send mail configuration is valid, and the **Attach file** option is selected.
Email subject <*language*>	Type the subject of the email.
Email message <*language*>	Type the body of the message.

You can use DQL substitutions such as `'$value(object_name)'` in the email subject and message. To specify the smart URL for the selected content item, use `'$value(locate_url)'`, or `'$value(ios_locate_url)'` to ensure the item is opened in the iOS version of the mobile app.

Note: `'$value(ios_locate_url)'` is currently only supported in the Classic client.

You can use `$foreachdoc` in combination with `$value` to organize multiple document references in an email. For example, if a user selects several documents in a doclist, then clicks **Share**, each document can be clearly identified in the email message body:

```
Hello,
I'd like to share $foreachdoc{$value(object_name)} with you.
To locate $foreachdoc {<a href='$value(locate_url)'>$value(object_name)</a>|,}
in OpenText Documentum CM click the links or copy and paste it into your browser

$foreachdoc{Locate url for doc : $value(locate_url)|
```

Note: `$foreach` can be used in the email message body or the message subject, but it is recommended for email message body only due to maximum string length limitations in the Subject field. Single object `$` tokens like `$value(object_name)` and Send mail configuration would be evaluated based on the first selected document.

Here is a list of parameters that you can use, but values will be set conditionally based on the flow:

`docbase_name, due_date, event_name, message_text,object_name, package_id, planned_start_date , task_priority , sender_name, supervisor_name, task_name, task_number, recipient_login_name, recipient_name, platform, mail_user_name, stamp, date_sent, link_cnt, package_type, content_type, content_size, dos_extension, temp_file_name, mail_script`

4. Click **Save**.

Related Topics

“Understanding How to Communicate with Users Through OpenText Documentum CM and Email” on page 185

“Configuring the Mail Server” on page 186

“Configuring a Distribution” on page 196

“Configuring a Subscription” on page 194

“Configuring a Mailing List” on page 189

“Understanding How to Communicate with Users Through OpenText Documentum CM and Email” on page 185

“Configuring the Mail Server” on page 186

“Configuring a Distribution” on page 196

“Configuring a Subscription” on page 194

"Configuring a Mailing List" on page 189

8.5 Configuring a Subscription

1. Navigate to **Go to** > **Subscription** from the menu bar.
2. Click **New** to create a subscription.

   If you want to create a child subscription that inherits the properties of an existing subscription, select a subscription and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this subscription applies. For example, adding the `QA` application would cause the subscription to only apply to matching quality assurance cases.
Defer notifications	Select the option to send notification events to the intermediate queue **subscription_audit**. After treating the queue, D2JobSubscriptionSendMail sends the notification email to subscribed users. When using the delayed notification feature, configure the email message using `$foreachdoc{}` to list the documents. For example, `$foreachdoc{'$value(title)', '$value(r_modifier)'}` As D2JobSubscriptionSendMail sends a single message that can contain events for multiple documents, not configuring the message with the above syntax can cause confusion.
Label <*language*>	Type a label.

4. On the **Events** tab, click **Add an event** to add an event:

   a. Select an event to audit.

      You cannot use the following events for event subscriptions because they do not apply to content:

      - d2_template

- d2_taxonomy
- d2_dictionary
- d2_workflow_added
- d2_workflow_removed
- d2_workflow_sent
- d2_delegation
- d2_connect

Event names may be suffixed by one or more characters.

b. Type a DQL qualification to filter the event. If you want to configure more than one condition for an event, change the DQL qualification for the event instead of adding another instance of the event because you can only save one instance of a configuration for an event. For example, to configure an event for one of two titles, use the following query:

```
title='toto' or title 'titi'
```

5. Select the **Email** tab to configure notification parameters as described in the following table:

Field	Description
Attach document	Select to attach content to the notification message.
Attach rendition with format	Click **Browse** to add a rendition format.
Email subject <*language*>	Type the subject of the email.
Email message <*language*>	Type the body of the message.

You can use the `$value` keyword to use property values. For example, `'$value(<document.property_name>)'` shows the property of a document.

You can use the `$alias` keyword to use a dictionary.

You can use HTML in the body of the email.

6. Click **Save**.

Related Topics

"Understanding How to Communicate with Users Through OpenText Documentum CM and Email" on page 185

"Configuring the Mail Server" on page 186

"Configuring a Distribution" on page 196

"Configuring Options for Sending Emails through OpenText Documentum CM" on page 192

"Configuring Options for Sending Emails through OpenText Documentum CM" on page 192

"Configuring a Mailing List" on page 189

8.6 Configuring a Distribution

1. Navigate to **Go to** > **Distribution** in the menu bar.

2. Click **New** to create a distribution.

   If you want to create a child distribution that inherits the properties of an existing distribution, select a distribution and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this distribution applies. For example, adding the `QA` application would cause the distribution to only apply to matching quality assurance cases.
Attach document	Select to attach content to the distribution email.
Attach rendition with format	Click **Browse** to add the rendition formats you want to use when sending content.
Label <*language*>	Type a label.
Property page	Select a property page. You can click the magnifying glass to preview the page. **Note:** Distribution does not support multiple panel property pages, hence these pages are not available in the list.
Attribute which contains recipients	Select a property that contains the list of recipients.
Email subject <*language*>	Type the subject of the email.
Email message <*language*>	Type the body of the message.

Field	Description
Electronic signature	Select to require an electronic signature during when accepting a distribution and rejecting a distribution.
Intention required	Select to require the reason with the electronic signature.
Intentions dictionary	Select a dictionary to enable users to select a reason from a list of options. Alternately, you can also type a custom `$value` into the field. For example: GR Approval $value(site_id) $value(some_custom_property) Reject This input would get the first workflow document (not supporting document) in the task to determine the $value then look for that dictionary config.

You can use the `$value` keyword to use property values. For example, `'$value(<document.property_name>)'` shows the property of a document.

You can use the `$alias` keyword to use a dictionary, and you can use HTML in the body of the email.

4. Click **Save**.

Distribution is limited by the fact that it is configurable for one type only. In configuration matrix, Distribution should be checked with the context which is the same type as the type used in the property page which is configured for the distribution.

Related Topics

"Understanding How to Communicate with Users Through OpenText Documentum CM and Email" on page 185

"Configuring the Mail Server" on page 186

"Configuring a Subscription" on page 194

"Configuring Options for Sending Emails through OpenText Documentum CM" on page 192

"Configuring Options for Sending Emails through OpenText Documentum CM" on page 192

"Configuring a Mailing List" on page 189

Chapter 9

Configuring External Services

9.1 Configuring a Rendition Server Connection

You cannot configure specific connections to rendition servers through the client configuration. Use the client configuration to send events and messages to preconfigured queues.

Note: System behavior for multiple rendition requests made on a document can be controlled with `allowRenditionRequest` set in the `d2fs.properties` file. See the *OpenText Documentum CM Installation Guide* for more information.

1. Navigate to **Go to** > **Rendition server** from the menu bar.
2. Click **New** to create a rendition server connection.

   If you want to create a child rendition server connection that inherits the properties of an existing rendition server connection, select a rendition server connection and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this rendition server connection applies. For example, adding the `QA` application would cause the rendition server connection to only apply to matching quality assurance cases.
Queue name	Select the rendition server queue you want to use. For example, when using default event names used by Documentum CM Server , you can select: • **dm_autorender_win31** for legacy PDF/HTML rendition requests. • **dm_mediaserver** for other rendition requests
Priority	Select the processing priority.

Field	Description
Event	If you selected **dm_autorender_win31**, type `rendition` If you selected **dm_mediaserver**, type `dm_register_asset` for imported items and `dm_transcode_content` for other rendition requests.
Message	Type the message you want accompanying the event.
Rendition format	Select format to which the rendition is created.
Don't overwrite the rendition of a document with the same format	Select if you do not want the rendition server to overwrite a rendition if a rendition of the same format already exists.

4. Click **Save**.

Related Topics

“Configuring an Advanced Transform Services (ADTS) Rendition Server Connection” on page 200

“Enabling a Branch Office Caching Service (BOCS) Element” on page 201

9.2 Configuring an Advanced Transform Services (ADTS) Rendition Server Connection

You must configure ADTS servers to use thumbnail view mode.

Note: System behavior for multiple rendition requests made on a document can be controlled with `allowRenditionRequest` set in the `d2fs.properties` file. See the *OpenText Documentum CM Installation Guide* for more information.

1. Navigate to **Go to** > **Rendition server** from the menu bar.
2. Click **New** to create a rendition server connection.

   If you want to create a child rendition server connection that inherits the properties of an existing rendition server connection, select a rendition server connection and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.

Field	Description
Description	Type a description.
Applications	Add or remove the applications to which this rendition server connection applies. For example, adding the `QA` application would cause the rendition server connection to only apply to matching quality assurance cases.
Queue name	Select the queue **dm_mediaserver**.
Priority	Select the processing priority.
Event	Type the event `dm_register_asset`
Message	Do not type a message.
Rendition format	Select format to which the rendition is created.
Don't overwrite the rendition of a document with the same format	Select if you do not want the rendition server to overwrite a rendition if a rendition of the same format already exists.

4. Click **Save**.

Related Topics

"Configuring a Rendition Server Connection" on page 199

"Enabling a Branch Office Caching Service (BOCS) Element" on page 201

9.3 Enabling a Branch Office Caching Service (BOCS) Element

Set up BOCS elements that have been installed and configured in the client configuration to allow use by end users.

1. Navigate to **Go to** > **BOCS cache** from the menu bar.
2. Click **New** to create a BOCS element.

   If you want to create a child BOCS element that inherits the properties of an existing BOCS element, select a BOCS element and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Select a BOCS element.

Field	Description
Description	Type a description.
Applications	Add or remove the applications to which this BOCS element applies. For example, adding the `QA` application would cause the BOCS element to only apply to matching quality assurance cases.

4. Click **Save**.

Related Topics

“Configuring a Rendition Server Connection” on page 199

“Configuring an Advanced Transform Services (ADTS) Rendition Server Connection” on page 200

9.4 Configuring an OpenText Core Share Connection

Refer to the *OpenText Documentum Content Management Installation Guide* for more information about configuring the installation of the Documentum Connector for Core, where you will need to perform prerequisite tasks such as updating the keystore and configuring properties files.

Once the connector is configured, refer to “Configuring Smart View client Menus” on page 304 to establish a Smart View sub-menu that allows users of the Smart View Client to access OpenText Core Share functionality.

Chapter 10

Configuring Client Configuration Tools

10.1 Configuring Relations

Create and configure relations in OpenText Documentum CM Composer, then install them into the repository. The *OpenText Documentum CM Administrator User Guide* contains further information. You can configure specific application of relations in OpenText Documentum CM, such as the type of relation, inheritance of the relation during versioning, and so on.

Smart View supports both parent-to-child and child-to-parent relationships by type, as configured in Composer. To link new content as a child, use the **Link new content as child** option when configuring inheritance. See “Configuring Inheritance” on page 73.

You can place controlled relations in property pages, so that end users can set related contents depending on the type of relation.

1. Navigate to **Tools** > **Relation** from the menu bar.

2. Click **New** to create a relation.

   If you want to create a child relation that inherits the properties of an existing relation, select a relation and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. For each relation, fill out the row as described in the following table:

Field	Description
Relation label	Type a label for the relation to display in the client.
Relation type	Select a relation type.
Permanent link	This field is automatically set, and it indicates the `permanent_link` property of the relation as created in OpenText Documentum CM Composer. If **Permanent link** is set to `Yes`, new versions of the content creates a new, identical relation to keep the relation through versioning.
Multiple relation	Select if the relation is a one-to-many relationship.

Field	Description
Controlled relation	Select if you want the relation to only be set through the properties page. **Note:** For more information on controlled relation support, see "Support for Classic features in Smart View" on page 237.
Child version	Select the behavior of the relation for child versions of the content. Select **All versions** to have each child version link to the relation. In this case, where one parent has a relation with multiple child versions, end users see one relation in the **Relation** tab when selecting he parent. Select **Current** to have the only the current version link to the relation. In this case, where a child document has relations toward several versions of the parent document, end users see one relation in the **Relation** tab when selecting the child. Select **Unique version** to have only the version used for creating the relation link to the relation.
Type	If you want to enable relation properties, select a relation subtype. This list includes `dm_relation` and all its subtypes.
Property page	If you want users to edit relation properties, select a corresponding property page for the subtype you selected in **Type**. This property page cannot include controlled relations.

4. Click **Save**.

Related Topics

"Configuring Automated Delegation" on page 206

"Configuring Fonts" on page 207

"Configuring Access to client user interface and the client configuration" on page 207

"Configuring Client Options" on page 209

10.1.1 Relation properties

Relation properties are an optional feature that allow users to enter additional details about a relation. Depending on the subtype and how you configure the relation, users can enter these properties when creating a relation, or on the relation property page.

To configure relation properties:

1. Enable relation properties: select the `dm_relation` subtype in the **Type** column in relation configuration.
2. Create additional relation properties by adding custom `dm_relation` subtypes.
3. Create a property page with the `dm_relation` subtypes you want users to define when creating a relation. Then assign it to the relation configuration.

> **Note:** There are limitations on when users can define and edit some relation properties:
>
> - When creating a relation, users can define default `dm_relation` subtypes and custom subtype attributes.
> - On the properties page, users can edit the default `dm_relation description` attribute and custom `dm_relation` subtype attributes.

If you create relation properties, you can enable users to add them as columns to widget tables by adding the `dm_relation` subtypes to the client column configuration. You can also add them as default columns for widget tables by adding the `dm_relation` subtypes to a widget's default column configuration.

Related Topics

"Configuring Relations" on page 203

"Configuring a Property Page" on page 48

"Configuring an Internal Widget" on page 276.

"Configuring Client Column Preferences" on page 329

10.2 Configuring Automated Delegation

Use the automated delegation tool to create automated delegation on behalf of other users. For example, if a user goes on vacation for two weeks and forgets to set up self delegation using Delegations widget, another user from the team can set up the delegation on his behalf to automatically delegate all the workflow tasks from that user to specified users and groups.

1. Navigate to **Tools** > **Manage Delegations** from the menu bar.
2. Click **New** to create a delegation control.

   If you want to create a child delegation control that inherits the properties of an existing delegation control, select a delegation control and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
User/Group name	Select a user or group who can setup a delegation for other users.
Manage user/group delegations	Select the list of users or groups for whom the automated delegations can be targeted. Click **Browse**, then use the list controls to add or remove automated delegation targets.

4. Click **Save**.

Related Topics

"Configuring Relations" on page 203

"Configuring Fonts" on page 207

"Configuring Access to client user interface and the client configuration" on page 207

"Configuring Client Options" on page 209

10.3 Configuring Fonts

You can configure the fonts used in the XSL outputs produced by plugins.

1. Navigate to **Tools** > **Fonts** from the menu bar.
2. Click **Import** to add a font, then fill out the table as described in the following table:

Field	Description
Name	Type a name.
File	Locate and select the font file.
Format	Select the file format.

3. Click **OK**.
4. Type a **Label**.
5. To save the font:
   - Click **Save** to save to the repository.
   - Click **Export** to export the font in a file format you want.

Related Topics

"Configuring Relations" on page 203

"Configuring Automated Delegation" on page 206

"Configuring Access to client user interface and the client configuration" on page 207

"Configuring Client Options" on page 209

10.4 Configuring Access to client user interface and the client configuration

Configure access privileges to the client user interfaces and the client configuration based on user groups.

1. Navigate to **Tools** > **Options** from the menu bar.
2. Select **Display all groups** if you want to limit the list boxes to only the groups to which you have administrator rights.
3. Select user groups for the list boxes as described in the following table:

Field	Description
Access group for Client	Select the user group to have access to client user interfaces. If you leave this list box blank, all groups have access.
Access group for skipping SSO in Client	Select the user group which can bypass single sign-on configuration and enter credentials to access client user interfaces. If you configure SSO and leave this list box blank, users will not be able to log in from the login screen. All users will need to log in with SSO.
Access group for Client-Config	Select the user group to have access to the client configuration. If you leave this list box blank, all groups have access.
Access group for skipping SSO in Client-Config	Select the user group which can bypass single sign-on configuration and enter credentials to access the client configuration. If you leave the list box blank, no groups can use SSO to access client configuration.

4. Select a group from **Use IDP for electronic signature** if you require users to sign approvals (for example, during task completion) using their OpenText Directory Services (OTDS) login and password instead of OpenText Documentum CM's native electronic password dialog box. You can add a custom message that appears when the signature is accepted. See the **Confirmation message** field in the Property Page configuration, Lifecycle configuration, or Workflow configuration.

5. Click **Save**.

Related Topics

"Configuring Relations" on page 203

"Configuring Automated Delegation" on page 206

"Configuring Fonts" on page 207

"Configuring Client Options" on page 209

"Configuring Access to Administration Widgets in the Client" on page 332

10.5 Configuring Application Selection at Login

You can allow users to select the application they want to open when they log in. You can select an application that appears as the default selection in the dialog, and acts as the default if the user accesses objects through a URL that does not feature an application parameter.

Note: To enable application selection at login, **Load on startup** needs to configured for all the applicable repositories. See the *OpenText Documentum CM Installation Guide* for more information about modifying the `D2FS.Properties` file for this purpose.

1. In **Tools > Options** Runtime Mode section, select **Application Evaluation** as the Matrix Evaluation mode.
2. Make sure the **Allow application selection on login** checkbox is checked.
3. Make sure you designate Applications as eligible for selection at login. In **Tools > Options** (or when creating a new application), check the **Include at Login** checkbox.
4. Choose the application you want to appear as the default in the login dialog's **Default Application** drop-down list. This default will also be selected if a URL is used to access objects, and no explicit application parameter is defined in the URL.

Depending on individual permissions, users will be able to select from the range of available applications in the **Application** drop-down list in the login dialog.

10.6 Configuring Client Options

1. Navigate to **Tools** > **Options** from the menu bar.
2. Fill out the form as described in the following table:

Field	Description
Use IDP for electronic signature	For electronic signature authentication, users in the selected group will be required to enter their username and password through the IDP they have configured to use in OTDS, rather than the built-in electronic signature dialog. This requires: • OTDS as IDP or OTDS with SAML IDP. • IDP supports reauthentication for active user session • IDP supports the prompt=login option (OpenID) or ForceAuthn=true option (SAML)
Cut, copy, and paste are not allowed	Select content that you want to forbid for cut, copy, and paste. For example, you can prevent the manipulation of content with a counter in its name. To add a content set, select a set from **Contexts** and click the **>** button. To remove a set, select a set from **Contexts which cut, copy, and paste are not allowed** and click the **<** button.

Field	Description
Clients URL	Type client web application URLs to refresh the caches of application servers. Click **Add**, type the URL for the application server you want to add, and click **OK**. To remove a server, select the server and click **Delete**. To ensure cache refresh requests are sent to all back-end services, list the specific URL for every singular instance of the application servers. For example: • http://*<hostname>*:*<port>*/D2 • http://*<hostname>*:*<port>*/D2-Smartview • http://*<hostname>*:*<port>*/d2-rest • http://*<hostname>*:*<port>*/D2-BOCS • http://*<hostname>*:*<port>*/D2FS **Important** • URLs must point to a specific application server instance, not a load balancer. • Do not include the client configuration application server in this list.
Web Server URL	Type web server URLs. Distributions, Subscriptions and Tasks can have a `$value(web_server_urls#)` in the subject or body. The # is the index and is optional (default is first).
Enable the Word files comparison	Select to enable comparison of content through two different versions of Word using the Versions widget.
Properties list display mode	Select the display mode for properties: • **Name**: for only names. • **Label**: for only labels. • **Name (Label)**: for names followed by labels. • **Label (Name)**: for labels followed by names.
Use DocApp constraints in property pages	Select to use the constraints for values defined for the repository when the attributes were created in OpenText Documentum CM Composer.

Field	Description
Move all versions	Select to move all versions of the selected document. This enables the following behavior: • All versions of the selected document in the same folder are moved. • No matter if the target document is the current or a previous version, in either case, all versions of the document that are linked to the same folder are moved. • When a virtual document is selected and moved, the root document is moved with all its versions, including descendants (which also reside in the same folder as the root document).
Show “Go to location” link after successful document creation	After adding or uploading a file, replace the default **Go to overview** link with a **Go to location** link to open the file location of the new file.
Redirect URL	Type a URL to configure a redirect URL for logout. **Note:** The redirect URL must have a protocol such as `http://` or `https://`. For SSO deployments, such as SAML or OTDS, the logout URL is required to be the same web address domain as the OpenText Documentum CM servers. If you do not define a redirect URL for logout, then `D2LogoutResponse.html` is used.

3. Click **Save**.

Related Topics

“Configuring Relations” on page 203

“Configuring Automated Delegation” on page 206

“Configuring Fonts” on page 207

“Configuring Access to client user interface and the client configuration” on page 207

10.7 Enabling Actions for Intelligent URLs

To prevent end users from performing unintended actions, such as deleting content, actions are disabled for use in intelligent URLs by default. You can change the setting for an action to enable it for your end user.

1. Navigate to **Tools** > **Options** from the menu bar.
2. Use the list controls to add or remove **Allowed actions** in the **Allowed actions in URL** section.
3. Click **Save**.
4. Navigate to **Tools > Refresh Cache** to ensure that your changes take effect on the client.

10.8 Renaming a User or Group with a Script on the Documentum CM Server

The following section describes how to change user or group names using the rename user script.

1. Retrieve the `RenameUser.bat` and `RenameUser - Example.bat` scripts from the `d2userrename/User_Rename_Config.zip` file, which can be found in the `D2-Config/utils` folder.
2. Edit the `RenameUser - Example.bat` script as necessary: `-docbase_name`, `-user_name`, and `-password`, then complete the rename parameters by supplying the `-from` and `-to` name information.

   - To rename users, add the correct `-docbase_name`, `-user_name`, and `-password`, then complete the rename parameters by supplying the `-from` and `-to` name information.
   - To rename a group, change the `– operation` to “Rename Group” and update the `-from` and `-to` as required.

   Notes

   - If you would like to test the rename job before you run it, set the `-test` flag to `true`.
   - If the user you want to rename has special characters (such as accented characters) in their original name, instead of setting the parameters in the `RenameUser-Example.bat` and running it, enter the same command in a command prompt directly after navigating to the utility location and run it from that location.
   - The script ignores content custom attributes that happen to contain user names. The method `D2RenameUserOrGroup` uses a schema configuration to determine which fields to update. The configuration is stored in objects of type called `d2_schema_configuration`. If you create an

extension in the object for custom object model extensions, it should be picked up by the method and do the job. For example:

List all current configs:

```
select r_object_id,object_name,table_names,column_names,is_
auditable,is_included_in_object_names,is_repository_object,is_
valid from d2_schema_configuration enable(ROW_BASED)
```

- If you use a Linux script, arguments that have spaces need to be enclosed within `\` delimited quotes: `\"`.

  For example:

```
RenameUser.sh argument1 \"Argument 2\" argument3 \"Argument 4\"
```

- After the rename operation is completed, rename operation report generated at %temp% location and report generated by CS at <Documentum>/dba/logs/<000xxx>/sysadmin/userrenamedoc.txt, both of them can be verified for more details on what objects have been updated with the rename operation.

3. Save the changes in `RenameUser - Example.bat`

4. Run the `RenameUserTest - Example.bat` script.

The script should log the outcome of the operation, and should also send a report to the instigating user's InBox in OpenText Documentum CM. The reports can also be found in the `/System/SysAdmin/Reports` folder in the repository.

After you run the utility, check the report generated in the user temp location with the operation name and the `userrenamedoc.txt` report generated in *`<Documentum>`*`/dba/logs/<000xxx>/sysadmin`. These reports list which objects were updated by the rename option.

You must also check either the user or group rename as follows:

- Rename user: Check the report and status of the dm_UserRename job in DA to verify that CS completed the renaming the relevant users. Query dm_user with user_name in the database to ensure the user_name has been updated to the new value.
- Rename Group: Check the report and status of dm_GroupRename job in DA to verify that CS completed naming the relevant groups. Query dm_group with group_name in the database to ensure the group name has been updated to the new value.

After successfully renaming a user name ensure:

- The user can still log in using their original user ID and password (the user login name, or user ID, is not updated by this process).
- The new user name is shown in the main window.
- The user can continue to navigate the same folders and documents in the repository with the same roles and permissions.

- The user's InBox works as before, and any in-flight workflow tasks assigned to the user are preserved.
- Any documents that were checked-out to the user prior to the rename operation are now locked under the new user name, and the user is able check these in as normal.

Note: The Rename Utility’s process of picking `d2c_workflow_tracker` files varies based on the XQuery search status. If the XQuery is enabled, it is expected to add the `d2c_workflow_tracker` files to indexing. If the XQuery is not enabled, the system internally processes the tracker files.

Chapter 11

Auditing and Monitoring

11.1 Configuring Audit Events

You can configure specific audit events.

1. Navigate to **Tools** > **Documentum audit** from the menu bar.
2. Select **Audit** to log an audit event whenever a user connects to OpenText Documentum CM.
3. Use the list controls to select document types. An audit event logs whenever content of the selected types is deleted.
4. Click **Save**.

Related Topics

"Configuring Audit" on page 217

"Monitoring" on page 222

"Configuring Audit" on page 217

11.2 Configuring Audit

Configure the audited events and properties. The *OpenText OpenText Documentum CM Server System Object Reference* contains further information on auditing non-system events.

1. Navigate to **Go to** > **Audit** from the menu bar.
2. Click **New** to create an audit template.

   If you want to create a child audit template that inherits the properties of an existing audit template, select an audit template and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.

Field	Description
Applications	Add or remove the applications to which this audit template applies. For example, adding the `QA` application would cause the audit template to only apply to matching quality assurance cases.

4. Use the list controls to configure the list of **Audited events** and **Displayed events**.

   The Audit widget only shows events in the **Displayed events** list.

   If you place events in the **Displayed events** list, only the listed events display in the Audit widget.

   Select **Limit the display to client events** if you want to remove non-system events from the events lists.

   Select **Extended display** if you want to use extended messages to describe event information in the Audit widget in place of standard columns.

5. Use the list controls to configure the list of **Audited properties**:

   a. Select **Display previous and new value** if you want to display the value change.

   b. For a repeating property, select **Display only changes** if you want to omit events that resulted in no change.

6. Click **Save**.

Note: Audit configurations require a `Type`–based context mapping in the configuration matrix. See “Configuring Contexts” on page 19.

Related Topics

“Configuring Audit Events” on page 217

“Monitoring” on page 222

“Configuring Contexts” on page 19

11.3 Audit Event Reference

Server Audit Event	Comparable Audit Event(s)
dm_acquire	d2_workflow_acquired_task
dm_addnote	d2_workflow_added_note
dm_addrendition	d2_rendition_import, d2_rendition_request
dm_assemble	d2_vd_convert_to_vd
dm_bp_attach	d2_change_state
dm_checkin	d2_checkin
dm_checkout	d2_checkout
dm_delegatedworkitem	d2_workflow_delegated_task
dm_destroy	d2_destroy
dm_disassemble	d2_vd_convert_to_simple_doc
dm_freeze	d2_vd_create_snapshot
dm_getfile	d2_view, d2_export
dm_insertpart	d2_vd_add_component
dm_lock	d2_edit
dm_removepart	d2_vd_remove_component
dm_removerendition	d2_rendition_destroy
dm_save	d2_create, d2_save_properties
dm_signoff	d2_workflow_signoff
dm_startworkflow	d2_workflow_started
dm_unlock	d2_cancel_checkout
dm_updatepart	d2_vd_move_component

Full List of Audit Events	
c2_controled_print	d2_relation_create
c2_controlled_print_recall	d2_relation_destroy
d2_add_inherited_component	d2_rendition_destroy
d2_annotate	d2_rendition_import
d2_autolink	d2_rendition_request
d2_bin_restore_antecedant	d2_security
d2_bin_restore_rendition	d2_send_mail
d2_bin_restore_version	d2_set_template
d2_cancel_checkout	d2_thumbnail_get

Full List of Audit Events	
d2_change_state	d2_vd_add_component
d2_change_state_failed	d2_vd_binding_component
d2_change_state_signoff	d2_vd_convert_to_simple_doc
d2_change_state_success	d2_vd_convert_to_vd
d2_checkin	d2_vd_create_snapshot
d2_checkout	d2_vd_move_component
d2_create	d2_vd_remove_component
d2_delegation_delegated_task	d2_view
d2_delegation_undo_delegated_task	d2_view_inline
d2_destroy	d2_workflow_aborted
d2_destroy_failed	d2_workflow_acquired_task
d2_digital_signature_request_cancelled	d2_workflow_added_note
d2_digital_signature_request_declined	d2_workflow_added_pseudo_task
d2_digital_signature_request_sent	d2_workflow_auto_aborted
d2_digital_signature_request_signed	d2_workflow_auto_acquired_task
d2_digital_signature_signed_document_retri eved	d2_workflow_auto_delegated_task
d2_digital_signature_signer_declined	d2_workflow_auto_forwarded_task
d2_digital_signature_signer_forwarded	d2_workflow_auto_rejected_task
d2_digital_signature_signer_signed	d2_workflow_auto_sent_mail
d2_digital_signature_signing_log_retrieved	d2_workflow_delegated_task
d2_distribution_accept	d2_workflow_forwarded_task
d2_distribution_accept_signoff	d2_workflow_received_mail
d2_distribution_launch	d2_workflow_rejected_task
d2_distribution_prepare	d2_workflow_removed_pseudo_task
d2_distribution_reject	d2_workflow_scheduled
d2_distribution_reject_signoff	d2_workflow_send_mail_failed
d2_distribution_sendmail	d2_workflow_sent_mail
d2_distribution_stop	d2_workflow_signoff
d2_distribution_update	d2_workflow_started
d2_edit	o2_apply
d2_export	d2_share_to_core_create
d2_import	d2_share_to_core_update

Full List of Audit Events	
d2_mass_update	d2_share_to_core_delete
d2_mass_update_signoff	d2_workflow_completed_auto_task
d2_print	d2_workflow_rerun_auto_task
d2_properties_save	d2_workflow_resumed_task
d2_properties_save_signoff	d2_workflow_paused_task
d2_properties_view	

The following table describes email notification event details:

Event	The client configuration name	Required configuration	Trigger
dm_startedworkitem	on task receive **Note:** "on task receive" changes to "dm_startedworkitem" after saved.		A workflow task is sent.
d2_workflow_started		Run the "D2JobWFLaunchScheduledWorkflows" job.	A scheduled workflow starts.
d2_workflow_not_finished		Run the "D2JobWFWorkflowsNotifications" job.	The workflow passes its due date.
dm_selectedworkitem	dm_selectedworkitem	Register this event.	
dm_completedworkitem	dm_completedworkitem	Register this event.	
dm_terminateworkflow	dm_terminateworkflow	Register this event.	

11.4 Monitoring

You can access the monitoring information by navigation to **Help** > **Monitoring** from the menu bar. This opens the **Monitoring** dialog box. The following table describes the information presented in the tabs:

Tab	Description
D2 (re)Loading	Shows the state of OpenText Documentum CM on application servers and checks that the URL is responsive.
Summary	Shows information about the JVM running OpenText Documentum CM on the operating system.
VM	Shows information about the JVM running OpenText Documentum CM on the virtual machine.
Memory	Shows information about the memory status for the JVM running OpenText Documentum CM.
Threads	Shows information about the threads for the JVM running OpenText Documentum CM.
Classes	Shows information about the classes for the JVM running OpenText Documentum CM.
Documentum	Shows information about active sessions and OpenText Documentum CM resources.
D2 Cache	Shows the number of caches in use along with all statistical information.
ACS/BOCS	Shows the URL status of ACS and BOCS configured in OpenText Documentum CM Administrator.

Click **Refresh** to update all monitored information.

Related Topics

"Configuring Audit Events" on page 217

"Configuring Audit" on page 217

Chapter 12

Using Configurations

12.1 Understanding Configurations

You can back up all repository configurations for backup purposes as well as to transfer configurations through the importing feature. The actions you can take regarding configurations are:

- Backup
- Restore
- Export
- Import

You can view the version of the current configuration by navigating to About > Client Solution > Configuration.

Related Topics

"Backing Up a Configuration" on page 224

"Restoring a Configuration" on page 225

"Exporting a Configuration" on page 226

"Importing a Configuration" on page 227

"Importing a Configuration Zip Using a Command Line" on page 234

"Resetting the Configuration" on page 235

12.2 Configuring session timeout

By default, if a user's session is inactive for 30 minutes, the session times out and the user will have to login again.

If you want to change the number of minutes before a session times out, adjust the *<session-config>* value in the `<appserver>\webapps\D2-Config\WEB-INF\web.xml` file. For example, to set the inactive time-out to 20 minutes, adjust the `web.xml` file as follows:

```
<session-config>
        <session-timeout>20</session-timeout>
</session-config>
```

The default For *<session-config>*, is `30`. If you enter `0`, the session will not time out.

If you want to display a message before the user's session times out, you can use the *client_warning_session_timeout* parameter in the same file. For example, if you want to display a warning message two minutes before a user's session times out, add the following to the `web.xml` file:

```
<context-param>
        <param-name>client_warning_session_timeout</param-name>
      <param-value>2</param-value>
</session-config>
```

After modifying the `web.xml` file, you must restart the app server.

12.3 Backing Up a Configuration

1. To create a new configuration object:

   a. Navigate to **File** > **Save configuration** > **Check in as new object** from the menu bar.

   b. Select an application or select **All applications** to define the scope of the configurations you want to save.

   c. Type a name and description of the configuration.

   d. Click **OK**.

2. To save a configuration as a new version of an existing configuration object:

   a. Navigate to **File** > **Save configuration** > **Check in configuration base** from the menu bar.

   b. Select and edit a name.

   c. Select the version.

   d. Type a description.

   e. Click **OK**.

Note: If you are using client configuration with the Microsoft Internet Explorer browser, the configuration is saved as a `.zip` file. If you are using a modern Chromium-based browser, the configuration is saved as an `.xml` file.

Related Topics

"Understanding Configurations" on page 223

"Restoring a Configuration" on page 225

"Exporting a Configuration" on page 226

"Importing a Configuration" on page 227

"Importing a Configuration Zip Using a Command Line" on page 234

"Resetting the Configuration" on page 235

12.4 Restoring a Configuration

1. Navigate to **File** > **Restore configuration** from the menu bar.
2. Select a configuration.
3. Select the version you want to use.
4. Click **OK**.
5. Select options as described in the following table:

Option	Description
Full import with current config reset	Import the entire configuration and erase the current configuration.
Full import without actual config reset	Import the entire configuration without erasing the current configuration. New contexts are placed between the lowest-priority context and the default context. Existing contexts keep their position.
Include Context-Configuration matrix selections	Import the toggled status of the configuration matrix.
Overwrite existing elements	Replace existing configurations with the imported configuration.
Do not overwrite the autonaming values in the new configuration	Keep existing values of autonaming counters.
Do not overwrite the cache URLs	Keep the cache URLs.
Do not overwrite the mail server configuration	Keep the mail server URLs.

6. Select a context from **Import the context before this context** to select the relative context when importing contexts to the configuration matrix.
7. Select all modules you want to import.
8. Click **OK**.

Note: If you are using client configuration with the Microsoft Internet Explorer browser, the configuration is saved as a `.zip` file. If you are using a modern Chromium-based browser, the configuration is saved as an `.xml` file.

Related Topics

"Understanding Configurations" on page 223

"Backing Up a Configuration" on page 224

“Exporting a Configuration” on page 226

“Importing a Configuration” on page 227

“Importing a Configuration Zip Using a Command Line” on page 234

“Resetting the Configuration” on page 235

12.5 Exporting a Configuration

You can export a configuration as a .zip archive or a single .xml file. The .zip archive method allows you to export all of the Property Configurations and select individual property configurations for export. The XML method exports all of the Property Configurations, but does not allow you to select individual property configurations.

Note: The XML configuration import/export does not support Business DocApp packages.

1. Navigate to **File** > **Export configuration** to export as .zip, or **File** > **Export XML configuration** from the menu bar.
2. Select an application or select **All elements** to define the scope of the configurations you want to export. To export all configurations, select **Full config export**. If you are exporting to .zip archive, you are also permitted to select individual property configurations. A suggested file name will be populated and will include a timestamp based on the App server local time.
3. Click **OK**.

Related Topics

“Understanding Configurations” on page 223

“Backing Up a Configuration” on page 224

“Restoring a Configuration” on page 225

“Importing a Configuration” on page 227

“Importing a Configuration Zip Using a Command Line” on page 234

“Resetting the Configuration” on page 235

12.6 Importing a Configuration

You can import both .zip archive and .xml file configurations.

Note: The XML configuration import/export does not support Business DocApp packages.

1. Navigate to **File** > **Import configuration** to import a .zip file, or **File** > **Import XML configuration** from the menu bar.
2. Select the configuration Zip or XML file you want to import.
3. Click **Open**.
4. Select options as described in the following table:

Option	Description
Full import with current config reset	Import the entire configuration and erase the current configuration
Full import without actual config reset	Import the entire configuration without erasing the current configuration. New contexts are placed between the lowest-priority context and the default context. Existing contexts keep their position.
Include Context-Configuration matrix selections	Import the toggled status of the configuration matrix.
Overwrite existing elements	Replace existing configurations with the imported configuration.
Do not overwrite the autonaming values in the new configuration	Keep existing values of autonaming counters.
Do not overwrite the cache URLs	Keep the cache URLs.
Do not overwrite the mail server configuration	Keep the mail server URLs.

5. Select a context from **Import the context before this context** to select the relative context when importing contexts to the configuration matrix.
6. Select all modules you want to import.
7. If you are importing updates to existing configurations, select **Report extension configuration warnings** to generate a PDF report that lists the configurations that were changed.
8. Click **OK**.
9. After the configuration is imported, navigate to **Tools > Refresh Cache** to ensure that your changes take effect on the client.

Note: `D2_documentset` (also known as Contexts) config types feature the attribute `order_no` which determines the position from left to right where the context appears in client configuration. In some scenarios, the same `document_set` might have a different `order_no` between merging configurations, and the merged `order_no` might overlap an already imported `d2_documentset` during the import process. In this scenario, the second `d2_documentset` would be pushed to the end of the context list.

Related Topics

"Understanding Configurations" on page 223

"Backing Up a Configuration" on page 224

"Restoring a Configuration" on page 225

"Exporting a Configuration" on page 226

"Importing a Configuration Zip Using a Command Line" on page 234

"Resetting the Configuration" on page 235

12.6.1 Maximum sizes of imported files

You can configure the maximum allowable sizes of configuration files that are imported into client configuration by editing the following properties in the D2–Config.properties file.

Description	Property name	Default Size (in KB)
Maximum size of a C2 file	importC2FileMaxSize	50000
Maximum size of a dictionary file	importDictionaryFileMaxSize	50000
Maximum size of a registered table	importRegisterFileMaxSiz	50000
Maximum size of a taxonomy file	importTaxonomyFileMaxSize	60000
Maximum size of a configuration file (XML or Zip)	importFileD2WebFileMaxSiz e	50000
Maximum size of a file uploaded to WidgetView	importFileX3ConfigFileMaxS ize	50000

12.6.2 Configuring new file types allowed for import in client configuration

You can configure the allowable file types that can be imported in client configuration using the D2–Config.properties file.

1. Find the Magic Byte of the file type you want to add. This can be found using a web search. For example the Magic Byte for a XLS file is `D0 CF 11 E0 A1 B1 1A E1`.

   Note: If a Magic Byte for the new file type cannot be found, it cannot be added to the allowable file list. Contact D2 Engineering for further instruction.

2. Navigate to the # **Define allowed File Types** section in the D2–Config.properties file and add the new file type. Use capital letters and use the following format:

   `FILE_TYPE_XLS=XLS`.

3. Navigate to the# **DEFINE MAGIC BYTES HERE** section in the D2–Config.properties file and add the new file magic byte information. Use capital letters and comma separated hexadecimal characters. For example: `MAGIC_BYTE_XLS=0xD0,0xCF,0x11,0xE0,0xA1,0xB1,0x1A,0xE1`.

4. Locate where the new import file will be used.

Section title	Property name	Default file types
#### import C2 File ####	IMPORT_C2_FILE_ALLOWED_FILE_TYPES_DEFAULT	PNG, JPG
	IMPORT_C2_FILE_ALLOWED_FILE_TYPES_FOR_C2_TOC_C2_XSL_C2_LAYER	PDF, PNG, JPG, SVG, XML, XSL
	IMPORT_C2_FILE_ALLOWED_FILE_TYPES_FOR_C2_MERGE	PDF
	IMPORT_C2_FILE_ALLOWED_FILE_TYPES_FOR_C2_WATERMARK	PDF, PNG, JPG, SVG
#### import Dictionary ####	IMPORT_DICTIONARY_ALLOWED_FILE_TYPES	XLS, XLSX, XML, CSV
#### import File D2 Web ######	IMPORT_FILE_D2_WEB_FILE_TYPES	ZIP, XML, TTF, OTF
#### import File X3 Config ####	IMPORT_FILE_X3_CONFIG_FILE_TYPES_X3_IMAGE	PNG, JPG, SVG, GIF

Section title	Property name	Default file types
	IMPORT_FILE_X3_CONFIG_FILE_TYPES_X3_SPACE	XML
#### import Register ####	IMPORT_REGISTER_FILE_TYPES	XLS, XLSX, CSV
#### import Taxonomy ####	IMPORT_TAXONOMY_FILE_TYPES	XLS, XLSX, XML, CSV

5. Add the new file type to the end of the list using a comma. For example: to add the XLS file type as allowable file type for D2 Web `IMPORT_FILE_D2_WEB_FILE_TYPES=ZIP, XML, TTF, OTF, XLS`.

12.7 Running a Report on a Single XML Configuration

You can run a report that shows the details of the current XML configuration or a separate XML configuration file that can be imported.

1. Navigate to **File** > **Generate XML specification**.

2. Select the configuration you want to report on. **Current Configuration** or **Import Configuration**. If you select **Import Configuration**, browse to the location of the specification XML file.

3. Fill out the form as described in the following table:

Field	Description
Applications	Select the application for which the specification is generated.
Display elements without application	Select to display configuration components that are not assigned to applications.
Include dictionaries	Select to include dictionaries.
Include taxonomies	Select to include taxonomies.
Include registered tables	Select to include registered tables.

4. Click **OK**.

An XML Specification HTML report is generated. To browse through the report, click any of the configuration elements in the report’s table of contents to drill down for more detail.

12.8 Running a Report on Multiple XML Configurations for Delta or Merge

You can run a report that shows the deltas between two exported XML configuration files, three configuration files with a common ancestor, or the current configuration and exported XML configurations. For example, with a three-way compare you could run a report on the current configuration, an original baseline, and a new baseline to determine the differences between the three. You can also optionally create a merged file that can help you create a new configuration.

Note: The delta/merge utility is available for use with XML configurations from version 4.7 and higher.

1. Navigate to **File** > **Generate XML configuration delta**.
2. Select the configurations you want to compare by choosing a first, second, or third configuration. Select **Current Configuration** or**Import Configuration** and browse to the exported XML configuration file(s).

   Note: If you select a third configuration to compare, you must designate one of the three configurations as a **Common Ancestor**. Select that file from the drop-down at the bottom of the dialog.

3. Select your desired **Output Mode** for the report: **Differences** or **Merge**.
4. Click **OK** to generate your chosen report.
5. Save the file, then open it in your browser for review.

   The **Configuration Diff** report lists all the deltas between your chosen configurations. It only shows items that have been modified between configurations, or only show up in certain configurations. Differences will be indicated with a small colored box next to modified/added attributes and XML content, representing which configuration these items appear in. The ancestor configuration is indicated as a blue box, the edit1 configuration is indicated by a green box, and the edit2 configuration is indicated by a yellow box. You can navigate through the report table of contents, or by clicking the **Next** and **Previous** buttons which will rotate through the modified configurations. Only one configuration will be shown at a time, and it will appear at the bottom of the page.

   The **Configuration Merge** report looks very similar to the diff report, except it only includes the configurations that require manual conflict resolution. If the configurations are able to auto merge completely, nothing will be shown in the report table of contents. The **Next Conflict** button scrolls through all of the conflicts that require resolution, and the **Apply Changes** button generates the final merged configuration for saving. If conflicts have not been resolved, the **Apply Changes** action notifies the user and jumps to the next unresolved conflict. As with the diff report, the user can see which configurations each change belongs to by the color box indicators. The merge report might display a scenario where a single radio button pair will be shown. Selecting one or the other will change the content to be merged in the final merged configuration.

Both reports might highlight a scenario where a control has moved within the configuration. These moved controls are highlighted with a red name. To resolve these conflicts, select to remove at least one of the sibling pairs by hovering over the control and clicking the red Remove button.

Note: The user-settings xml config records the date and date-time format as numerical pattern keys, which appear in the delta xml files. For example:

In the configuration type `d2_user_preferences_config`, subtype `d2c_preferences`, the following pattern might be observed:

`dateformat_datetime`:

Merge Value	Client Configuration Value
1	Jun 12, 2017 2:17:03 PM
2	Monday, June 12, 2017, 2:17:03 PM BST

`dateformat_date`:

Merge Value	Client Configuration Value
1	Jun 12, 2017
2	2017-06-12

Additionally, there are some cases in the XML export where a Default Value is listed as a blank/empty string where the string is defined as something specific to the config. For example, for **Facets** in **Search** config **Blank** refers to frequency:

`d2_search_config`:

Attribute	Default Value in Merge	Default Value EN
`facet_sort`	BLANK	Frequency

12.9 Running configuration management scripts

Three command line utilities are available that allow you to automate configuration management activities on the server where the client configuration is deployed:

- `D2ExportConfigXmlUtil`: Exports configurations in XML format.
- `D2ImportConfigXmlUtil`: Imports XML format configurations.
- `D2DeltaConfigXmlUtil`: Reports on the deltas between multiple XML format configuration exports.
- `D2XmlSpecificationUtil`: Creates a configuration export report.

Note: The following import commands are not supported in the `D2ImportConfigXmlUtil` tool:

- Include imported modules' relative matrix switches
- Report extension configuration warning

1. In a command shell, navigate to the directory where the client configuration web application has been deployed, and then to the `d2configxml` subdirectory.
2. Run the desired utility:
   - In Windows: `D2`*<xxx>*`ConfigXmlUtil.cmd`
   - In Linux: `D2`*<xxx>*`ConfigXmlUti.sh`
3. See below for example usage with required and optional parameters. Running a utility with no arguments displays usage help:

Example 12-1: Commands

D2ExportConfigXmlUtil.cmd [-login *<superUserLoginName>*] [-password *<password>*] [-docbase *<docbase>*] [-file_path *<filepath>*]

```
Required Parameters:
  -login: docbase superuser loginname.
  -password: docbase password for superuser.
  -docbase: docbase name.
  -file_path: pathname of output file

Optional Parameters:
  -app: name of desired application, NONE for no application (default: ALL).
  -noAppEl: include configuration elements that do not have applications,
 T or F (default: T).
  -type: names of desired configuration type elements each separated by a
';' and surrounded by double quotes ex: "x3_widget_config;x3_space_config"
(default: all configuration elements).
```

D2ImportConfigXmlUtil.cmd [-login *<superUserLoginName>*] [-password *<password>*] [-docbase *<docbase>*] [-file_path *<filepath>*] [-full_import *<true|false>*] [-reset *<true|false>*]

```
Required Parameters:
  -login: docbase superuser loginname.
  -password: docbase password for superuser.
  -docbase: docbase name.
  -file_path: pathname of import D2 configuration file
  -full_import: full or custom import - Boolean
  -reset: reset the existing configuration - Boolean (default: false)

Optional Parameters:
  -overwrite: Overwrite the existing configuration - Boolean (default: false).
  -keep_naming_counter: Keep autonaming counters - Boolean (default: false).
  -keep_url_cache: Keep cache URL values - Boolean (default: false).
  -keep_mail_server: Keep mail server URL values - Boolean (default: false).
  -import_context_before: Specifies the relative context to import when importing
contexts. This parameter is case sensitive and you must specify the context value in
double quotes.

Module List:
For a custom import, inquire the list of module type which must be imported
followed by the list of module config names.
-module_type "module config name": The module config name have to be between
```

```
quote. A list of name can be put separate by the character '|'.
Example:
-d2_documentset "documentSet name" -d2_workflow_config
"workflow config name 1|workflow config name 2|workflow config name 3"
```

`D2DeltaConfigXmlUtil.cmd [-login` *<USERNAME>*`][-password` *<PASSWORD>*`][-docbase` *<DOCBASE>*`][-first` *<FIRST_CONFIG>*`][-second` *<SECOND_CONFIG>*`] [-result` *<RESULT_OUTPUT_PATH>*`]`

```
Required Parameters:
  -login: docbase superuser loginname.
  -password: docbase password for superuser.
  -docbase: docbase name.
  -first: First D2 Config XML file Path. Provide string 'current' for current
repo configuration to be used
  -second: Second D2 Config XML file Path.
  -third: Third D2 Config XML file Path
  -ancestor: Common ancestor file for three-way compare file Path
  -result: Result file path
Optional Parameters:
  -locale: The desired language output messages.
  -merge: IS_MERGE_MODE (true/false)
```

`D2XmlSpecificationUtil.cmd [-login` *<USERNAME>*`][-password` *<PASSWORD>*`] [-docbase` *<DOCBASE>*`][-config` *<CONFIG_PATH>*`][-result` *<RESULT_OUTPUT_PATH>*`]`

```
optional params: [-locale LOCALE_STRING]
```

12.10 Importing a Configuration Zip Using a Command Line

1. Open a DOS or Linux console.
2. Type the following line:

```
java -cp <[classpath]> com.emc.d2.api.config.batch.D2ConfigImport <[-login
login_name]> <[-password password]> <[-docbase docbase_name]> <[-config_file
file_config_path]> <[-full_import true|false]> <[optional parameters]> <[modules
list]>
```

The following table describes the parameters:

Parameter	Description
`classpath`	Classpaths contain OpenText Documentum CM libraries, DFC libraries, D2-Config\WEB-INF\lib*, and the `dfc.properties` file.
`login`	Repository connection login
`password`	Repository connection password
`docbase`	Repository name
`config_file`	Configuration file

Parameter	Description
`full_import`	Full or custom import based on true or false, respectively
`reset`	Reset of existing configuration
`overwrite`	Overwrite the existing configuration (default: false)
`keep_naming_counter`	Keep autonaming counters (default: false)
`keep_url_cache`	Keep cache URL values (default: false)
`keep_mail_server`	Keep mail server URL values (default: false)
`help`	Show a help message with a list of parameters
`import_context_before`	Specifies the relative context to import when importing contexts. This parameter is case sensitive and you must specify the context value in double quotes.

Related Topics

"Understanding Configurations" on page 223

"Backing Up a Configuration" on page 224

"Restoring a Configuration" on page 225

"Exporting a Configuration" on page 226

"Importing a Configuration" on page 227

"Resetting the Configuration" on page 235

12.11 Resetting the Configuration

You can erase the current configuration and revert to default settings. This can be done from the client configuration user interface, or by using a command line utility.

Note: The command line reset option is suggested if the reset is expected to be long-running, or needs to run overnight.

To reset using the client configuration interface:

1. Navigate to **File** > **Reset configuration** from the menu bar.
2. Click **OK**.

To reset using the command line utility:

1. In a command shell, navigate to `\D2-Config\utils\d2configresetutil`.
2. Run the following:

   `D2ConfigResetUtil.cmd [-login superUserLoginName] [-password password] [-docbase docbase]`

   Required parameters:

   `-login`: docbase superuser login name

   `-password`: docbase password for superuser

   `-docbase`: docbase name

Related Topics

"Understanding Configurations" on page 223

"Backing Up a Configuration" on page 224

"Restoring a Configuration" on page 225

"Exporting a Configuration" on page 226

"Importing a Configuration" on page 227

"Importing a Configuration Zip Using a Command Line" on page 234

Chapter 13

Configuring the Client

13.1 Understanding the Smart View client

Multiple separate client interfaces are available:

- Smart View: A client that uses Open Text's tile-based Smart View user interface.
- Classic: The legacy, widget-based user interface.
- Mobile: A mobile user interface available on mobile browsers and mobile app.

If desired, a Smart View menu action to switch to the Classic view (and vice-versa) can be included in your solution.

If configured, users can switch to the Classic client at any time from the Smart View client. In the **Profile** menu, click **Classic View**.

if configured, users can switch to Smart View at any time from Classic. In the **Profile** menu on the menu bar, click **Smart View**.

Consult the following charts to understand important differences between the clients. For additional, specific information about feature and configuration support differences, see “Smart View Support Reference“ on page 513.

13.2 Support for Classic features in Smart View

This table lists Classic features and to what degree they are supported in Smart View.

Classic Feature	Smart View Support	Notes
Logging In	Supported	
Session timeout warning	Supported	
Apple iPad	Supported - see notes	Smart View is the preferred client on the iPad and provides a better, responsive, user experience compared to Classic.
Workspace Gallery	Supported - see notes	Smart View uses a single Landing Page from which a user can access available features.

Classic Feature	Smart View Support	Notes
Menus	Supported - see notes	Smart View menus are configurable, similar to Classic.
Workspaces; Add/Remove/ Change	Partially Supported - see notes	Smart View uses a single landing page from which a user can access available features. The administrator can add, remove, or change tiles or landing pages per contexts.
Workspaces; Change Theme	Supported	
Workspaces: Resizing, expanding, collapsing	Not Applicable	Smart View uses a single landing page from which a user can access available features.
Widgets; Adding, removing, replacing, moving	Partially Supported - see notes	Smart View uses a single landing page from which a user can access available features. The administrator can add, remove, or change tiles or landing pages per contexts.
Widgets; Exporting the list to Excel	Supported - see notes	Smart View supports the option to zip and download files and folders with a manifest file. Users can export the manifest only, which includes file properties.
Change the list columns	Supported	
Use column filters	Supported	
Sorting by column	Supported	
Supports different Widgets	Supported	
Search; Quick	Supported	
Search; Query form	Supported	
Searches; Running a saved search	Supported	
Searches; Creating a saved search	Supported	
Searches; Running a query form search	Supported	
Searches; **Show Criteria** before running search	Supported	

Classic Feature	Smart View Support	Notes
Searches; Facets	Supported - see notes	Smart View supports using facets as part of search results as well as in the doclist itself. Facets also support multi-selection in the Smart View.
Searches; Advanced	Supported	
User Preferences	Partially Supported - see notes	User Settings supports the enablement of Client Manager, the setting of Checkout and Temp paths, Accessibility toggle, Sort Folder/Files toggle, and Language for the user interface.
Viewing content	Supported	
Creating content	Supported	
Editing content	Supported	
Importing content	Supported	
Importing content as new version	Supported	
Exporting content	Supported	
Checking in	Supported	
Checking out	Supported	
Cancel Checkout	Supported	
Deleting content	Supported	
Renaming content	Supported - see notes	Available in Property Page.
Viewing content permissions	Supported	
Viewing content properties	Supported	
Copying and pasting content	Supported	
Copying and linking content	Supported	
Cross Repository support	Not Available	
Creating a folder	Supported	
Importing a folder’s structure	Supported	
Exporting a folder and its content	Supported - see notes	Smart View supports the option to zip and download files and folders, which allows users to export folders and their contents.
Deleting a folder	Supported	

Classic Feature	Smart View Support	Notes
Renaming a folder	Supported	
Viewing folder permissions	Supported	
Viewing folder properties	Supported	
Copying a folder using drag and drop	Not Available	
Creating a cabinet	Supported	
Convert to Virtual Document	Supported	
Locate	Supported	
Comment	Supported	
Favorite	Supported	
Subscribe	Not Available	
Send email	Supported	
Send email with file attachment	Not Available	
Copy link to clipboard	Supported	
Advanced Lifecycles	Supported - see notes	Advanced Lifecycles are specific to OpenText Documentum CM and configured in client configuration.
Lifecycles (dm_lifecycle)	Partially Supported - see notes	Neither Classic or Smart View clients support interacting with *dm_lifecycles* in the user interface. If you migrate from Webtop or other clients that support *dm_lifecycles*–based lifecycles, we recommend that you migrate to Advanced Lifecycles (*d2_lifecycle_config*). If your *dm_lifecycle* does not require a user interface, then it is supported. However, we still recommend switching to Advanced Lifecycles because you can configure them in client configuration.
Send to Workflow	Supported	
Distribution	Not Available	

Classic Feature	Smart View Support	Notes
Mass Update	Partially Supported - see notes	Mass update support for all controls except grid fields.
Filter	Partially Supported - see notes	Filters are supported on doclists and collections. The Favorites widget uses an **All version** toggle to show or hide all versions of items.
PDF Configuration Support		
Viewing with PDF Configuration	Partially Supported - see notes	In Smart View, documents with a configured PDF view display as in Classic, but there is no menu action to let the user choose which configuration to use.
Exporting with PDF Configuration	Partially Supported - see notes	In Smart View, documents with a PDF export configuration export as in Classic, but there is no menu action to let the user choose which configuration to use.
Printing with PDF Configuration	Partially Supported - see notes	In Smart View, documents with a PDF print configuration print as in Classic, but there is no menu action to let the user choose which configuration to use.
Recalling a controlled print	Not Available	
Recycle Bin Support	Supported	
Copying and Pasting Content between Repositories	Not Available	Not Available
Understanding Paste as Link Action in Multiple repositories	Not Available	
Copying and Linking Content between Repositories	Not Available	
Copying Content from OpenText Documentum CM to Desktop using Drag and Drop	Not Available	

Classic Feature	Smart View Support	Notes
Drag and Drop behaviors from desktop to OpenText Documentum CM	Partially Supported - see notes	Files can be dragged into the Smart View client to start an **Import File** action. A file can be dragged into the client onto a single file to start an **Add Version** action. A folder can be dragged to start import folder structure.
Drag and Drop behaviors from Outlook to OpenText Documentum CM	Supported - see notes	Supported in modern browsers (excluding Microsoft Internet Explorer.
Importing Content as a new version	Supported	
Importing an email and attachments	Supported	
Creating an XML file to Fill Properties During Import	Not Available	
Exporting content from the repository list widget	Supported	
Exporting content from the checked out widget	Supported	
Exporting content from your favorites widget	Supported	
Supports Presets Profiles during creation flows	Supported	Preset profile configuration allows creation flow to have a preset profile configured to streamline creation flows.
Supports Recent Profiles during creation flows	Supported	Preset profile configuration displays recent creation profiles the user has used to streamline repetitive creation flows.
Controlled Relations	Partially Supported - see notes	Controlled relations are supported for **Import**, **Create**, and **Standard** property pages. They appear on **Add Version**, **Lifecycle**, and **Workflow** property pages but they are not supported.

13.3 Configuring mobile

OpenText Documentum CM Mobile is a lightweight mobile app that links directly to a repository, allowing users to browse, access, search, and add content on any iOS or Android device, from anywhere.

The mobile browser client allows mobile users to access a repository from mobile browsers such as Safari (iOS) and Chrome (Android).

To understand how to install mobile client, follow the instructions in the *OpenText Documentum Content Management Installation Guide*. More information about mobile can be found in the *OpenText Documentum CM Client and Mobile Release Notes*.

Information about mobile configuration can be found throughout this guide:

- Established property pages can be used with mobile, but some elements work differently, which might require you to configure a mobile-specific version of the page. For more information, see “Configuring a Property Page” on page 48.
- Smart View and mobile share a landing page definition file. In the case of mobile, both the main menu and the landing page are defined in the Landing page xml file. For more information about using the `targetPlatform` tag to make tiles appear on the mobile Landing page, and to define the menu, see “Designing a Smart View and Mobile Landing Page” on page 260 and “Understanding Mobile Main Menu in the Smart View Landing Page Config” on page 269.
- The mobile app includes barcode scanning functionality that must be set up in a property page and the Landing page xml before it can be accessed on the device. See “Adding a Mobile Barcode Scan tile to the Landing Page XML” on page 270.
- The default Smart View Action Toolbar menu item **Download** is required for the Mark Offline feature of the mobile app. Removing the **Download** menu item from this and other toolbars will cause **Mark Offline** to disappear from applicable mobile menus. See “Configuring Smart View client Menus” on page 304 for more information about menus.

Note: For information on security configuration to manage the Mobile app’s behavior, see section 6.2 “MDM Settings” in *OpenText AppWorks Gateway - Installation and Administration Guide (OTAG-H-IGD)*.

13.3.1 Cookie storage setting

For version 23.2 and later, to use the mobile app, you must update how cookies are stored by browsers in the Smart View deployment.

To do so, set the *`samesite`* property to `none` in the `rest-api-runtime.properties` file. For example:

```
rest.security.client.token.cookie.samesite=none
```

Notes

- This change is only required if you use the mobile app. Access on a mobile browser does not require this update.
- Setting the *`samesite`* property to `none` does not impact client security.
- After you set the *`samesite`* property to `none`, users must access Smart View using HTTPS. If users access Smart View using HTTP, they will not be able to log in. This is a browser security restriction.

13.3.2 Force password-based authentication

When Smart View is configured with OTDS authentication and **Force password based authentication (non-OAuth)** is enabled in Appworks Gateway, then you must configure Smart View's *`rest.security.auth.mode`* property in the `rest-api-runtime.properties` file with `ct-otds_tikect-otds_token`.

13.3.3 Two–factor authentication

Two-factor authentication (2FA) can be enabled for the Mobile client using OTDS configuration settings. OTDS has an embedded two-factor authentication solution that uses a time-based one-time password (TOTP) client application to obtain the authentication codes required to log in to the Mobile client. Two-factor authentication can also be implemented with a third-party two-factor authentication provider instead of OTDS' embedded solution. For more information about implementing two-factor authenticating, see section 3.8.8 "Enabling two-factor authentication" in *OpenText Directory Services - Installation and Administration Guide (OTDS-IWC)*.

13.4 Configuring a Client Workspace

Classic View and Smart View both have the concept of workspaces, however, their application and configuration are separate.

Classic View workspaces define a container of widgets that allows users to personalize functionality for availability and convenience. Smart View workspaces are a content container that is created from a workspace type. Workspaces allow Smart View users to collaborate and share information, including metadata, documents, and images, and typically represent a meaningful entity in an organization.

If you configure a workspace for Classic View, that configuration does not impact Smart View. Similarly, if you configure a workspace for Smart View, it does not impact Classic View.

13.4.1 Configuring a Workspace for Smart View

The majority of Smart View workspace configuration is done in the Admin Console. See *OpenText Documentum Content Management - Workspaces Administration Help (EDCADC-H-ABW)* for more information.

There are some minor configuration requirements in client configuration. Here is a list of Smart View workspace configurations:

- Add and configure a default workspace widget on a Smart View landing page: “Configuring an Internal Widget” on page 276. The widget name is `BusinessWorkspaceWidget`.
- Create a property page to correspond to a workspace: “Configuring a Property Page” on page 48. To create a property page for a workspace:
    1. Under **Document type** select the name of the workspace that you created in Admin Console (*OpenText Documentum Content Management - Workspaces Administration Help (EDCADC-H-ABW)*).
    2. For each attribute you created in Admin Console, create a corresponding property object.
- Add workspace menus: “Configuring Smart View client Menus” on page 304. These menus appear in workspaces by default.
- Add a dynamic workspace menu to the **+Menu** toolbar to allow users to create any type of workspace they have permission to create: “Dynamic Menu Config Java Class Reference” on page 301. The relevant Java class is `MenuBWSTypesList`.

Smart View workspace limitations

- Creation profiles:
    – In a Smart View workspace, the standard and extended creation profiles that display match the document types assigned in Admin Console to the folders

in the workspace type's templates. If no document type is assigned to a folder, the default creation profiles display based on the priority order, as described in “Configuring Creation Profiles” on page 120. For more information about assigning document types to a folder in a workspace type template, see section 3.3 “Add templates to a workspace type” in *OpenText Documentum Content Management - Workspaces Administration Help (EDCADC-H-ABW)*.

- Do not configure a standard or extended creation profile with `dm_bws` or its subtypes. Smart View workspaces cannot be created using a creation profile. This is an incorrect configuration and can cause errors.

- Auto-linking:
  - Auto linking configurations do not apply to Smart View workspaces or their contents.
  - Do not include a workspace, the `dm_bws` type, or its subtypes in an auto link path. Smart View workspaces cannot be created with an auto link. This is an incorrect configuration and can cause errors.
- Security configurations do not apply to Smart View workspaces or their contents.

13.4.2 Configuring a Workspace for Classic View

1. Navigate to **Widget view** > **Workspace** from the menu bar.
2. Click **New** to create a workspace.

   If you want to create a child workspace that inherits the properties of an existing workspace, select a workspace and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this workspace applies. For example, adding the `QA` application would cause the workspace to only apply to matching quality assurance cases.
Label <*language*>	Type a label.
Description <*language*>	Type a description.
Workspace structure file	Select the workspace XML to use as this template configuration.

Field	Description
Preview file	Select the preview image, if any, to use as a thumbnail so the end user has an idea of what the workspace looks like. If you do not select or import a thumbnail, a thumbnail automatically generates one.
Default theme	Select the default color theme used for the workspace. You must select a theme. Define the default theme in the global registry docbase so that it is applied to the login screen and workspace gallery. Initially, the docbase settings are not retrieved and the theme does not display (please verify if this is correct). Ensure that the default theme is specified both for the global registry docbase setting and for retrieval. If multiple themes are checked in the default column in client configuration, the theme from the first row is applied, by default.

4. Select whether the workspace uses multiple views. If you select **Multiple view**, use the list controls to add, remove, or reorder workspace views. For each view:

   a. Type the label for the view in **View** <*language*>.

   b. Click **Browse**, select a workspace XML, then click **OK**.

   c. Click **View** to preview the workspace template.

5. Select **Lock workspace layout** to prevent end users from modifying the widgets contained in a workspace. Select this option to prevent end users from removing widgets that are necessary for context-sensitive widget operations. For example, you can configure the Locate widget so that when an end user selects content within the Locate widget, the Doclist widget displays.

6. To configure one or more events that cause a workspace view to active:

   a. Use the list controls for **Add focus events for the selected workspace view** to add, remove, or reorder actions and events. List of Widget Communication Channels contains a list and description of the events, actions, and their parameters.

   b. Select an event to configure its **Additional event parameters**.

      For example, if you add and select the event `D2_EVENT_WIDGET_FOCUS` to the `TasksWidget`, you can type the parameter `widgetType==BrowserWidget`. When the end user selects the Document browser widget, the selected workspace view becomes visible and active, then sets focus to the Document browser widget if it is found in the new view.

      You can also type the parameter `config==` and set the value to the **Name** of the widget. You can use this parameter to differentiate between instances of the same widget. For example, you can create a Document browser widget with the name `Word document browser` and another browser widget with the name `PDF document browser`. If you use `widgetType==BrowserWidget` the focus event for both widgets triggers. If you use `config=="Word document browser"` you can specify the specified widget.

7. Click **Save**.

Related Topics

"Designing a Workspace for Classic" on page 249

"Understanding the Border Layout Container" on page 250

"Understanding the Tab Container" on page 252

"Understanding the Accordion Container" on page 253

"Understanding the Horizontal and Vertical Containers" on page 254

"Configuring a Client Theme" on page 255

"Configuring an External Widget" on page 286

"Configuring an Internal Widget" on page 276

"Configuring Client Menus" on page 293

"Configuring Client Column Preferences" on page 329

"Configuring Display Filters" on page 326

"Configuring Custom Icons for Rendition Formats and Content Types (Classic)" on page 319

"Configuring Icons and Text Color for Content Properties in List Widgets (Classic)" on page 321

"Refreshing and Reloading" on page 332

"Configuring Access to Administration Widgets in the Client" on page 332

13.4.2.1 Designing a Workspace for Classic

There is no user interface to create a workspace layout. It must be done directly by editing a user-created XML file.

The workspace XML must contain the following object:

```
<?xml version="1.0" encoding="UTF-8"?>
<root xsi:noNameSpaceSchemaLocation="SpaceSchema.xsd" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance">
  <space>
    <border-layout-container>
  </space>
</root>
```

There must only be one `<root>` tag at the beginning of the file, which contains one `<space>` tag.

The `<space>` tag requires at least one `<border-layout-container>`.

All containers may contain itself, other containers, or widgets. The following table lists the available containers and possible subcomponents:

Container	Subcomponents
Border layout container	Containers and widgets
Tab container	Widgets
Accordion container	Widgets
Vertical and horizontal containers	Containers and widgets

Related Topics

"Understanding the Border Layout Container" on page 250

"Understanding the Tab Container" on page 252

"Understanding the Accordion Container" on page 253

"Understanding the Horizontal and Vertical Containers" on page 254

"Configuring a Workspace for Classic View" on page 246

"Understanding the Border Layout Container" on page 250

"Understanding the Tab Container" on page 252

"Understanding the Accordion Container" on page 253

"Understanding the Horizontal and Vertical Containers" on page 254

13.5 Understanding the Border Layout Container

The border layout container is the main frame of the workspace. Border layout containers are composed of data nodes and content nodes.

Data nodes are exclusive to border layout containers and define the location and size of container elements. The following table describes the attributes you can use:

Attribute	Description
`layoutregion`	Defines the location of a container element. • `NORTH` for north (top of the container) • `SOUTH` for south (bottom of the container) • `EAST` for east (right of the container) • `WEST` for west (left of the container) • `CENTER` for center (middle of the container)
`size`	Defines the default container size. You can define sizes in two ways, and the size is automatically detected. If you type a value greater than 1, the value is considered an absolute size in pixels. For example, `100` would be 100 pixels. If you type a value less than 1, the value is considered a relative size. For example, `0.5` would be 50%, or half the size of the window. **Note:** The `maxSize` and `minSize` attributes are no longer applicable during resizing the container.
`split`	Determines enabling or disabling of resizing of the container. Set `split` to `TRUE` to enable resizing of the container. Set `split` to `FALSE` (default value) to disable resizing of the container.

There must be a content node for each data node, and they define the elements used in a container. If the element is a widget, you must specify the name of the widget. If the element is a container, you must specify the name of the container. The number of elements that can be defined is limited to 5 for display reasons.

The following image contains an example code of a border layout container with data and content nodes:

```
<?xml version="1.0" encoding="UTF-8"?>
- <root xsi:noNamespaceSchemaLocation="SpaceSchema.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <space>
    - <border-layout-container>
      - <data>
          <border-layout-data split="true" layoutregion="CENTER" />
          <border-layout-data split="true" layoutregion="EAST" size="480"/>
          <border-layout-data split="true" layoutregion="WEST" size="260"/>
        </data>
      - <content>
        - <tab-container>
          - <content>
              <widget config="doclist"/>
              <widget config="docgallery"/>
            </content>
          </tab-container>
        - <border-layout-container>
          - <data>
              <border-layout-data split="false" layoutregion="NORTH" size="100"/>
              <border-layout-data split="false" layoutregion="CENTER"/>
            </data>
          - <content>
              <widget config="quicksearch"/>
            - <accordion-container>
              - <content>
                  <widget config="locations"/>
                  <widget config="versions"/>
                  <widget config="renditions"/>
                  <widget config="relations"/>
                  <widget config="audits"/>
                </content>
              </accordion-container>
            </content>
          </border-layout-container>
          <widget config="browser"/>
        </content>
      </border-layout-container>
    </space>
  </root>
```

Related Topics

"Designing a Workspace for Classic" on page 249

"Understanding the Tab Container" on page 252

"Understanding the Accordion Container" on page 253

"Understanding the Horizontal and Vertical Containers" on page 254

"Configuring a Workspace for Classic View" on page 246

"Designing a Workspace for Classic" on page 249

"Understanding the Tab Container" on page 252

"Understanding the Accordion Container" on page 253

"Understanding the Horizontal and Vertical Containers" on page 254

13.6 Understanding the Tab Container

The tab container is a type of horizontal layout with tab items organized horizontally.

Tab containers contain a single content node. Add a widget to the layout by entering the name of the widget in the `<widget config>` tag. There is no limit to the number of widgets you can put in a tab container. You do not need to define a size, as the tab container uses the size as determined by its parent container.

The following image contains an example of a tab container:

```
- <tab-container>
  - <content>
      <widget config="doclist"/>
      <widget config="docgallery"/>
    </content>
  </tab-container>
```

Related Topics

"Designing a Workspace for Classic" on page 249

"Understanding the Border Layout Container" on page 250

"Understanding the Accordion Container" on page 253

"Understanding the Horizontal and Vertical Containers" on page 254

"Configuring a Workspace for Classic View" on page 246

"Designing a Workspace for Classic" on page 249

"Understanding the Border Layout Container" on page 250

"Understanding the Accordion Container" on page 253

"Understanding the Horizontal and Vertical Containers" on page 254

13.7 Understanding the Accordion Container

The accordion container is a type of vertical layout with tab items organized vertically.

Accordion containers contain a single content node. Add a widget to the layout by entering the name of the widget in the `<widget config>` tag. There is no limit to the number of widgets you can put in an accordion container. You do not need to define a size, as the accordion container uses the size as determined by its parent container.

The following image contains an example of a accordion container:

```
- <accordion-container>
  - <content>
      <widget config="locations"/>
      <widget config="versions"/>
      <widget config="renditions"/>
      <widget config="relations"/>
      <widget config="audits"/>
    </content>
  </accordion-container>
</content>
```

Related Topics

"Designing a Workspace for Classic" on page 249

"Understanding the Border Layout Container" on page 250

"Understanding the Tab Container" on page 252

"Understanding the Horizontal and Vertical Containers" on page 254

"Configuring a Workspace for Classic View" on page 246

"Designing a Workspace for Classic" on page 249

"Understanding the Border Layout Container" on page 250

"Understanding the Tab Container" on page 252

"Understanding the Horizontal and Vertical Containers" on page 254

13.8 Understanding the Horizontal and Vertical Containers

Use horizontal and vertical containers to organize elements horizontally and vertically, respectively.

Horizontal and vertical containers can contain containers and widgets, which are placed in a content node. Resizing is disabled for horizontal and vertical containers.

The following image contains an example of a horizontal container:

```
- <horizontal-container>
   - <content>
        <widget config="browser"/>
        <widget config="doclist"/>
     </content>
  </horizontal-container>
```

The following image contains an example of a vertical container:

```
- <vertical-container>
   - <content>
        <widget config="taskfolders"/>
        <widget config="tasks"/>
     </content>
  </vertical-container>
```

Related Topics

"Designing a Workspace for Classic" on page 249

"Understanding the Border Layout Container" on page 250

"Understanding the Tab Container" on page 252

"Understanding the Accordion Container" on page 253

"Configuring a Workspace for Classic View" on page 246

"Designing a Workspace for Classic" on page 249

"Understanding the Border Layout Container" on page 250

"Understanding the Tab Container" on page 252

"Understanding the Accordion Container" on page 253

13.9 Configuring a Client Theme

You can configure color themes that end users can select and use on their Classic View workspaces. You can define the colors of the background, logo, borders, and so on.

This feature is especially useful in a multi-environment (development/testing/production). You can create three themes, one for each environment. For example, the development environment can be red based, testing can be green based, and production can be default based; all within a single configuration. This allows for deploying the same application to all environments, and then simply choosing the appropriate theme for the environment, instead of deploying the application and then editing it for each environment.

1. Navigate to **Widget view** > **Theme** from the menu bar.
2. Click **New** to create a theme.

   If you want to create a child theme that inherits the properties of an existing theme, select a theme and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this theme applies. For example, adding the `QA` application would cause the theme to only apply to matching quality assurance cases.
Label <*language*>	Type a label.

4. Configure the application header as described in the following table:

Field	Description
Logo	Select a logo from the list box. If you want to import a new logo, click **Browse** and fill out the **Import** dialog box. Use an image with a maximum dimension of 250 pixels width by 50 pixels height.
Background	Select a background color using the **Color selection** dialog box.
Normal	Select a color for the menu icons, labels, and buttons in the normal state using the **Color selection** dialog box.

Field	Description
Hover	Select a color for the menu icons, labels, and buttons for when an end user mouses over the menu using the **Color selection** dialog box.

5. Configure the workspace tabs as described in the following table:

Field	Description
Tab label	Select a color for the workspace tab label and workspace menu button using the **Color selection** dialog box.
Border	Select a color for the horizontal border below the workspace tabs using the **Color selection** dialog box.

6. Configure the widget tabs, borders, and item selection as described in the following table:

Field	Description
Tab label - Active	Select a color for the tab label, menu button, and button label for an active widget using the **Color selection** dialog box.
Tab label - Inactive	Select a color for the tab label and menu button for an inactive widget using the **Color selection** dialog box.
Tab background - Active	Select a color for the tab background and buttons for an active widget using the **Color selection** dialog box.
Tab background - Inactive	Select a color for the tab background for an inactive widget using the **Color selection** dialog box.
Use transparent tab background	Select this option to alter the theme to use transparent widget tabs and a larger tab font.
Selected Item - Active	Select a background color for a selected item in a grid, menu, list, and tree for an active widget using the **Color selection** dialog box. This color does not apply to the text color of the selected item. Depending on the color selected, the text color displays as black or white for visibility.

Field	Description
Selected Item - Inactive	Select a background color for a selected item in a grid, menu, list, and tree for an inactive widget using the **Color selection** dialog box. This color does not apply to the text color of the selected item. Depending on the color selected, the text color displays as black or white for visibility.
Border - Active	Select a color for the panel border for an active widget using the **Color selection** dialog box.

7. Configure the application background (workspace and application frame) as described in the following table:

Field	Description
Single color	Select this option to use a solid color background. Select a color using the **Color selection** dialog box.
Gradient	Select this option to use a gradient color background. Select a **Start color** and an **End color** for the gradient using the **Color selection** dialog box.
Image	Select this option to use a background image. Select a background image from the list box. If you want to import a new background image, click **Browse** and fill out the **Import** dialog box.
Use white dialog background	Select this option if you want dialog boxes to use a white background. If you do not select this option, dialog boxes use the application background.

8. Click **Save**.

Related Topics

"Configuring a Workspace for Classic View" on page 246

"Configuring an External Widget" on page 286

"Configuring an Internal Widget" on page 276

“Configuring Client Menus” on page 293

“Configuring Client Column Preferences” on page 329

“Configuring Display Filters” on page 326

“Configuring Custom Icons for Rendition Formats and Content Types (Classic)” on page 319

“Configuring Icons and Text Color for Content Properties in List Widgets (Classic)” on page 321

13.10 Classic Client Cache Management

To improve performance, OpenText Documentum CM uses Ehcache to cache the required client configuration objects to reduce the communication between Content Server and the database. OpenText Documentum CM provides a host of cache managers to cache client configuration objects and other useful metadata.

The Cache Manager loads the `d2-cache.xml` file from the `C6-Common.jar` file with the default settings. Any `<cache>` elements specified in this `d2-cache.xml` file can be overwritten by specifying these `<cache>` elements in the `d2-cache.xml` file in the `D2\WEB-INF\Classes` folder.

Note: Caching DFC objects or the `workflow_config` cache can cause timer exceptions when they use dynamic sessions. Session leaks have also been reported on preference objects when they re-attached sessions after data-fetching from the cache.

13.11 Configuring a Smart View Client Landing Page

The client configuration allows you to define the landing page that users of Smart View and mobile see when they log in.

The elements are defined in an XML landing page structure file, which is then selected and applied to the configuration matrix through client configuration. Mobile uses the same XML file to define its landing page. You can define the default tiles, widgets and their locations, sizes, and colors. Other settings such as persistent search field on the landing page header can also be established. See “Designing a Smart View and Mobile Landing Page” on page 260 for more information.

1. Navigate to **Widget view** > **Smart View Landing Page** from the menu bar.
2. Click **New** to create a landing config.

   If you want to create a landing page that inherits the properties of an existing landing page, select a landing page and click **Create from**.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this landing page applies. For example, adding the QA application would cause the landing page to apply to matching quality assurance cases only.
Label <*language*>	Type a label.
Description <*language*>	Type a description.
Default Landing Page	If multiple landing pages are configured and mapped in the configuration matrix, select this option to make this landing page the default when a user first logs in. When a user logs in again, whatever landing page they accessed last will open.
Landing Structure File	Select the landing page structure XML file to use as this landing page configuration. Click **Download** to download the XML file for viewing. Click **...** to import a new landing structure file. See "Designing a Smart View and Mobile Landing Page" on page 260 for more information about designing a landing page.
Default theme	Select the default color theme for the landing page. See "Configuring a Smart View client Theme" on page 274 for more information about designing a theme.

4. Click **Save**.

13.12 Designing a Smart View and Mobile Landing Page

An XML file is required to define a Smart View Landing Page. You can find an `example_landingPage.xml` file and the `U4Landing.xsd` with detailed comments in the D2–Config.war under `utils/d2smartview`. You can use this XSD in any XML editor to design the landing page and to validate it for errors.

See "Configuring a Smart View Client Landing Page" on page 258 for more information about selecting the landing page configuration xml file you have prepared.

The landing page configuration file controls the positioning and dimensions of the tiles and widgets on the landing page. Additionally, it features a Settings section that controls the behavior of landing page UI elements such as a persistent search field in the header.

The landing page XML must contain the following object:

```
<?xml version="1.0" encoding="utf-8"?>
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation='U4Landing.xsd'>
```

There must only be one `<root>` tag at the beginning of the file, which contains one `<space>` tag. There are also tags for `<default-widgets>` and `<settings>`. The `<space>` tag requires at least one `<flow-layout-container>`, which can hold several `<tile-container>` or `<widget-container>`.

A *tile container* is a solid tile on the landing page that redirects the user to some other destination based on its function within the Smart View. For example, a Saved Search tile will run a search and take the user to search results.

A *widget container* exposes core widget functionality on the landing page itself. Example widgets include Favorites, Recently Accessed, Saved Searches, and Checked-Out.

The order/grouping in the XML file determines the position of these tiles/widgets automatically, from top left to bottom right.

13.13 Understanding Tile Container in the Smart View Landing Page Config

`tile-container` is a container of one or more tiles or action-tiles. Each tile represents a DoclistWidget, QueryObject or an actionable item.

The following table describes the attributes you can use:

Attribute	Description
`size`	The size of the container. Currently, the only selection is `full`.
`tile`	An individual tile within a TileContainer that hosts features such as doclists, or a Workflow overview.
`action-tile`	`action-tile` is used to launch an action. Several types of actions are supported: • `UploadFile` • `CreateFile` • `CreateUploadFile` • `UploadFolder` • `MobileScanAndUpload` The titles for these tiles are picked from Smart View Menu Configuration. **Notes** • To use `CreateUploadFile`, a Smart View preset profile configuration should be available to the logged-in user. • At least one preset profile must be enabled for a mobile user to enable upload. Without a preset, the **Upload** tile, the **Scan and upload** tile, and the **Upload** menu option are disabled on mobile. • `MobileScanAndUpload` is only supported on the mobile apps.

Attribute	Description
`type`	Mandatory. For `tile`, this can be: • `DoclistWidget`: a standard doclist where content is listed. • `QueryFormObject`: a query form where the user enters search details. • `QueryObject`: a saved query that runs a static search and displays the results. • `OpenURLWidget`: a link to an internal or external URL. • `MobileBarCode`: add a tile in the mobile application that opens the device camera to scan a barcode. • `MobileQRCode`: add a tile in the mobile application that opens the device camera to scan a QR code (using either XML or JSON data). • `WorkflowOverviewWidget`: add a tile that allows you to search for Workflows. • `AdminAviatorWidget`: add an administration tile in an administration workspace in Smart View. For `action-tile` this can be: • `CreateFile` • `UploadFile` • `CreateUploadFile` • `UploadFolder`

Attribute	Description
`targetPlatform`	Shows or hides Landing page tiles or widgets depending on the platform: the standard Smart View desktop (`desktop`) or mobile (`mobile`). If you do not designate a `targetPlatform`, the tiles or widgets will potentially appear on both platforms, subject to support. The following tiles and widgets are supported on mobile, and will appear on the mobile Landing page: • Favorites • Recently accessed • Doclist shortcut • Collection • Import file, import with preset, scan and upload • DQL shortcut • Query form • Barcode scan • QR Code scan • Saved search • Tasks
`name`	Mandatory. For a doclist widget, the `name` is the Doclist Widget Name in client configuration. For QueryObject, `name` is the Public Search Name. Like DoclistWidget, the name is the Query Form Name in client configuration. The label shown in the landing page tile for both a DoclistWidget and Query Form widget will use the localized label in the corresponding client configuration object the entry references.
`panelsize`	Controls the size of the pane that appears if the tile opens a pane, such as if it uploads a file or initiates a lifecycle. The available options are: • **S**: Small. The pane fills 1/3 of the screen. • **M**: Medium. The pane fills 2/3 of the screen. • **L**: Large. The pane fills most of the screen. If you do not define a `panelsize`, the default size is **S**.

Attribute	Description
`presetConfigName` and `presetUniqueID`	`CreateFile` and `UploadFile` action-tiles can contain Presets that act as shortcuts for creating or importing specific types of content. The values for `presetConfigName` and `presetUniqueID` are taken from the Preset configuration explained in "Configuring Smart View Preset Profiles" on page 106. The tile title on the Landing Page is taken from the **Label** field in Preset configuration.
`theme`	Each tile can be styled by either a pre-defined theme of complimentary and 508/WCAG AA accessible color combinations or specific color (`types` and `color`), or by specifying a specific hex color code for background and text colors. Available preset `types`: • `"stone1"` (#5f6e77) • `"teal1"` (#005873) • `"pink1"` (#780050) • `"indigo1"` (#4f3690) • `"stone2"` (#3f4950) • `"teal2"` (#078db3) • `"pink2"` (#c83370) • `"indigo2"` (#2e3d98) Available preset `color`: • `"shade1"` • `"shade2"` • `"shade3"` • `"shade4"` See Figure 13-1 as an example of how these color settings display.

Note: If **Type** or **Name** is invalid, or the user does not have the required permission, the tile is still shown on the landing page, but no content appears in the tile except an error icon or other error message.

The following is an example of some `TileContainer` implementations with several individual tiles:

```
<tile-container size="full">
                    <tile name="PM Sandbox Find Documents by Status"
type="QueryFormObject">
                        <theme color="shade1" type="teal1"/>
                    </tile>
                    <tile name="PM Sandbox Find Checked-Out Documents"
type="QueryFormObject">
                        <theme color="shade3" type="teal1"/>
                    </tile>
```

```
                <tile name="Quick doc search" type="QueryObject">
                    <theme color="shade4" type="teal1"/>
                </tile>
            </tile-container>
            <tile-container size="full">
                <action-tile type="CreateUploadFile">
                    <theme color="shade1" type="indigo2"/>
                </action-tile>
                <action-tile type="CreateFile">
                    <theme color="shade2" type="indigo2"/>
                </action-tile>
                <action-tile type="UploadFile">
                    <theme color="shade3" type="indigo2"/>
                </action-tile>
                <action-tile type="UploadFolder">
                    <theme color="shade4" type="indigo2"/>
                </action-tile>
            </tile-container>
```

Figure 13-1: Color templates

13.14 Understanding Widget Container in the Smart View Landing Page Config

`widget-container` allows you to define individual widgets on the landing page. Each landing page widget is a widget defined in client configuration **Widget View**.

The following table describes the attributes you can use:

Attribute	Description
`type`	The type of widget. Available types are: • `AdminCheckedOutDocumentWidget` • `AdminWorkflowWidget` • `CheckoutDocumentWidget` • `CollectionWidget` • `FavoritesWidget` • `RecentlyAccessedWidget` • `SearchWidget` • `SharedDocumentWidget` • `TasksWidget` • `WorkflowOverviewWidget` • `RecycleBinWidget`
`name`	The widget name as defined in client configuration. The label shown in the Landing Page is based on the localized label provided in the corresponding configuration referenced by the entry.

Note: Both type and name are mandatory. If type or name is invalid or the user doesn't have permission to the widget, the widget container is still shown on the landing page, but no content appears in the widget except an error icon or other error message.

The following is an example of a `widget-container` implementation. The name of the widget needs to match the widget name you defined in client configuration **Widget View**:

```
<widget-container>
  <widget type="FavoritesWidget" name="Favorites" />
</widget-container>
```

13.15 Understanding Default Widgets in the Smart View Landing Page Config

A standard list of widgets is included in the sample landing page xml under default-widgets, but you can remove them.

Note: If excluded, users of the landing page will not see widgets such as Versions, Relations, and Renditions when they view the properties of content. Additionally, if some widgets (such as `vdoc–widget` for Virtual Documents) are not present, the Action and Inline Action menu items associated with the widget will not appear to the user, even if they are defined in the Smart View menu configuration.

To remove a default widget, delete the corresponding row from the xml file, as described in the following table:

Attribute	Description
`version-widget`	For showing versions of a document, the name of `DetailsVersionWidget` from client configuration has to be provided. If not provided, the user interface does not show the option to see Versions.
`relation-widget`	For showing relations of an object, the name of `DetailsRelationsWidget` from client configuration has to be provided. If not provided, the user interface does not show the option to see Relations.
`rendition-widget`	For showing renditions of a document, the name of `DetailsRenditionsWidget` from client configuration has to be provided. If not provided, the user interface does not show the option to see Renditions.
`doclist-searchresults-widget`	This is the doclist widget where the search results are shown for quick search. **Important** `doclist-searchresults-widget` is a default widget. If it is not provided, the landing page will not load.
`doclist-widget`	This is the default doclist widget that appears when user is browsing using DRLs, or uses Locate. **Important** `doclist-widget` is a default widget. If it is not provided, the landing page will not load.
`bravacsr-widget`	This is the default widget for the BravaCSR viewer.
`bravaenterprise-widget`	This is the default widget for the Brava! Enterprise viewer. For more information on this viewer, and setting up the `BravaEnterpriseViewer` widget, see “About the BravaEnterpriseViewerWidget” on page 390.
`otiv-widget`	This is the default widget for OpenText Intelligent Viewing.

Attribute	Description
`vdoc-widget`	This is the Virtual Document widget that allows the user to view and interact with Virtual Document and Virtual Document Snapshot outlines. **Note:** If this widget is not present, Virtual Document Action and Inline Action menu items associated with the widget will not appear to the user.

`tasks-widget`
 	
`task-attachments-widget`
	These two Task widgets are required in order for the user to view the Task lists and Task detail views in both Smart View and mobile.
	
	Note: If a `tasks-widget` is not defined in the Smart View Landing Page configuration xml file, the Task Details load using URL navigation (by bookmark or shared URL, for example) will not succeed, and if `task-attachment-widget` is not configured, then Task Details page will not display. This configuration is also required for the Workflow Details view.

`collection-widget`	This is the Collections widget, which allows users to see collections of files.

`all-workflows-widget`
 	
`my-workflows-widget`
	The `my-workflows-widget` widget is required to show a list perspective of workflows. `all-workflows-widget` is required to show a list of all workflows in the system. The value of the widget tag is the name you entered for the `WorkflowOverviewWidget` in the widget configuration. See “Configuring an Internal Widget” on page 276 for more information on setting up this widget.

`audits-widget`	The `audits-widget` allows users to view the audited events of a document. This widget refers to an instance of the `DetailsAuditWidget` widget.
`locations-widget`	Displays location information for items in the repository.

Note: If default `doclist-widget` or `doclist-searchresults-widget` are not provided, the landing page will not load.

The order of the default widget tags in the `.xml` file is strict. Departing from the example listed below might cause Landing Page elements to malfunction, or cause errors when you attempt to upload the `.xml` in client configuration. For example, place `all-workflows-widget` before `my-workflows-widget`. If these widgets are not defined in the Smart View Landing Page configuration .xml file, then the list of My Workflows, All Workflows and Workflow Details and load using URL navigation (by bookmark, shared URL or URL shortcut tile, for example) will not succeed.

The following is an example of a `default-widgets` implementation:

```
<default-widgets>
          <version-widget>Innovate Versions</version-widget>
          <relation-widget>Innovate Relations</relation-widget>
          <rendition-widget>Innovate Renditions</rendition-widget>
          <doclist-searchresults-widget>Innovate Doclist</doclist-searchresults-widget>
          <doclist-widget>Innovate Doclist</doclist-widget>
          <favorites-widget>Innovate Favorites</favorites-widget>
          <collection-widget>Innovate Collections</collection-widget>
          <bravacsr-widget>Innovate BravaCSR</BravaCSR-widget>
          <vdoc-widget>Innovate Virtual Documents</vdoc-widget>
          <tasks-widget>Innovate Tasklist</tasks-widget>
          <task-attachments-widget>Innovate WF Task Attachments</task-attachments-widget>
          <bravaenterprise-widget>Innovate Brava Enterprise Viewer</bravaenterprise-
widget>
          <otiv-widget>Innovate OTViewer</otiv-widget>
          <all-workflows-widget>Innovate WF Overview</all-workflows-widget>
          <my-workflows-widget>Innovate WF SV My Workflows</my-workflows-widget>
          <audits-widget>Innovate Audit</audits-widget>
          <locations-widget>Innovate Locations</locations-widget>
</default-widgets>
```

13.16 Understanding Mobile Main Menu in the Smart View Landing Page Config

You can define menu selections and the order that they appear in the mobile app's main menu using the landing page xml file. Some selections will appear by default (Home and Offline (in the case of the mobile app), but you can configure the Favorites and Recently Accessed selections.

Attribute	Description
`menu-item name`	The title of the menu selection, as you want it to appear in the main menu.
`type`	The type of menu item. The selections are `FavoritesWidget` and `RecentlyAccessedWidget`.

Note: The selections will appear in the menu in the order that they are listed in the landing page xml file.

The following is an example of a `mobile-menu` implementation:

```
<mobile-menu>
   <menu-item name="WG Favorites" type="FavoritesWidget"/>
   <menu-item name="WG Recently Accessed" type="RecentlyAccessedWidget"/>
</mobile-menu>
```

13.17 Adding a Mobile Barcode Scan tile to the Landing Page XML

To add a Barcode scan tile to the landing page of the mobile application, you need to configure a property page, create a query form that can handle the input from the scanner, then add a tile container to the Landing Page XML file.

1. Navigate to **Go to** > **Property page** from the menu bar.
2. Click **New** to create a barcode property page. Complete all of the required fields, including the creation of an attribute called `bar_code` (which will also be referenced in the query form and the landing page .xml).
3. Save the barcode property page.
4. Navigate to **Go to > Query form** from the menu bar.
5. Click **New** to create a Query Form. Complete all of the required fields, including the selection of the barcode property page. Select **DQL query** in the **Mapping** options, then add a statement similar to the following to the DQL field:

   `select r_object_id from d2mob_barcode where bar_code='$value(bar_code)'`
6. Save the query form.
7. In your Landing Page XML file, add a tile container that references the DQL query as the barcodekey, in this example `bar_code`:

```
<tile-container size="full">
      <tile type="MobileBarCode" barcodekey="bar_code" name="BarCode"
targetPlatform="mobile">
          <theme type="teal1" color="shade4"/>
      </tile>
  </tile-container>
```

   Note: The targetplatform is optional. This type of tile is restricted from appearing in the desktop Smart View client landing page.

8. Save the landing page .xml.

13.18 Adding and configuring a Mobile QR Code Scan tile to the Landing Page XML

To add a QR code scan tile to the landing page of the mobile application, you need to configure a QR code data extractor, create a query form using client configuration that can handle the input from the scanner, and then add a tile container to the Landing Page XML file.

13.18.1 Creating a QR code data extractor

To create a new QR code data extractor

1. Navigate to **Go to** > **QR Code Data Extractor** from the menu bar.
2. Click **New** to create a QR Code Data Extractor.
3. Enter the QR code data extractor properties.
4. Select a data type: XML or JSON.
5. Enter the details for your selected data type:
   - XML: Enter a parameter name and its path. You can add or remove parameters using the **+** or **—** buttons.
   - JSON: Enter a parameter name and its path. You can add or remove parameters using the **+** or **—** buttons.

     Note: For more information about XML and JSON formatting, see "XML and JSON format" on page 271.

   Note: You can add up to 10 parameters.
6. Click Save.

13.18.1.1 XML and JSON format

This section provides examples of XML and JSON QRCode configurations as well as links to external sites with additional information for each language.

XML example:

```
<?xml version="1.0" encoding="UTF-8"?>
<laptops>
    <brand2>
        <specifications>
            <hardDiskSize>256GB</hardDiskSize>
            <ram>8GB Soldered DDR4-2400</ram>
            <resolution>11.6" HD (1366x768) IPS 250nits Anti-glare</resolution>
            <batteryType>MobileBatt 2014: 13.9 hr</batteryType>
            <batterySize>Integrated 42Wh</batterySize>
            <usbType>USB 3.1 Gen 1</usbType>
            <customerCareNo>1876-123-456</customerCareNo>
        </specifications>
```

```
    </brand2>
</laptops>
```

Param name	XML Path Value
hard_disk	//laptops/brand2/specifications/hardDiskSize
ram	//laptops/brand2/specifications/ram
resolution	//laptops/brand2/specifications/resolution
battery_type	//laptops/brand2/specifications/battery_type
battery_size	//laptops/brand2/specifications/battery_size
usb_type	//laptops/brand2/specifications/usb_type
customer_care_no	//laptops/brand2/specifications/ customer_care_no

Note: Param names correspond to column names in the QRCode table.

For more information about XML and XPath for XML, see the following pages:

- XML tutorial: https://www.w3schools.com/xml/default.asp.
- XPath reference: https://www.w3schools.com/xml/xpath_intro.asp.

JSON example:

```
{
 "laptops":{
  "brand2": {
    "specifications": {
      "hardDiskSize": "256GB",
      "ram": "8GB Soldered DDR4-2400",
      "resolution": "11.6\" HD (1366x768) IPS 250nits Anti-glare",
      "batteryType": "MobileBatt 2014: 13.9 hr",
      "batterySize": "Integrated 42Wh",
      "usbType": "USB 3.1 Gen 1",
      "customerCareNo": "1876-123-456"
    }
  }
 }
}
```

Param name	JSON Path Value
hard_disk	$.laptops.lenovo.specifications.hardDiskSize
ram	$.laptops.lenovo.specifications.ram
resolution	$.laptops.lenovo.specifications.resolution
battery_type	$.laptops.lenovo.specifications.batteryType
battery_size	$.laptops.lenovo.specifications.batterySize
usb_type	$.laptops.lenovo.specifications.usbType
customer_care_no	$.laptops.lenovo.specifications.customerCare No

Note: Param names correspond to column names in the QRCode table.

For more information about JSON and JSONPath, see the following pages:

- JSON tutorial: https://www.w3schools.com/js/js_json_intro.asp.
- JSONPath reference: https://www.w3resource.com/JSON/JSONPath-with-JavaScript.php.

13.18.2 Creating a QR code query

To create a QR code query:

1. Navigate to **Go to > Query form** from the menu bar.
2. Click **New** to create a Query Form and complete all of the required **Properties** fields.
3. Under **Query / Form mapping**, select **QR Code** as the **Query form type**.
4. Select the required QR Code Config.
5. Save the query form.

13.18.3 Adding the QR code scanner tile to a mobile landing page

To add the QR code scanning tile to a mobile landing page

1. Open your Landing page XML file.
2. Add a tile container that references a queryform with the QRCode type. For example:

```
<tile-container size="full">
  <tile name="[<queryform name>]" type="MobileQRCode">
    <theme type="teal1" color="shade4"/>
  </tile>
</tile-container>
```

   Note: The targetplatform is optional. This type of tile is restricted from appearing in the desktop Smart View client landing page.

3. Save the landing page .xml.

13.19 Understanding Miscellaneous UI Settings in the Smart View Landing Page Config

The following user interface settings can be turned on or off from the landing page xml file, as described in the following table:

Setting	Description
`search-focus-header-enabled`	This entry is required in the XML otherwise the XML is considered invalid. If set to `true`, the search field in the landing page header bar is enabled and centered in the header. If set to `false`, the search field is collapsed on the right side of the header. User activates the field by clicking the Search icon.

Note: The `<disable-perfect-scrollbars>` and `<persistent-messages>` settings supported in previous releases have been deprecated and consolidated into an Accessibility Mode setting in the Smart View client's User Settings. You will need to remove the settings from landing page xml files for releases later than 20.2. An invalid file upload xml error will appear if they are present in the xml file.

13.20 Configuring a Smart View client Theme

You can configure color themes that you can apply to Smart View client landing page. You can define the colors of the page background, logo, and icon visual attributes.

1. Navigate to **Widget view** > **Smart View Theme...** from the menu bar.
2. Click **New** to create a theme.

   If you want to create a child theme that inherits the properties of an existing theme, select a theme and click **Create from**.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this theme applies. For example, adding the `QA` application would cause the theme to only apply to matching quality assurance cases.

Field	Description
Label *<language>*	Type a label.

Note: Watch the **Landing Preview** and **Login/Logout Preview** areas update as you customize to give you an idea of your overall theme.

4. Configure the landing page **Header** as described in the following table:

Field	Description
Logo	Select a logo from the list box. The default is the OpenText logo. If you want to import a new logo, click **Browse** and fill out the **Import** dialog box. Use an image with a maximum dimension of 250 pixels width by 40 pixels height.
Background	Select a header background color. Click either **Single color** or **Gradient**. Customize by clicking the color boxes beside the radio buttons and choosing from the **Color selection** dialog box. Customize the Gradient level by entering a value between 0–360 in the **Degree** field.
Icons	Select a color for the header icons in the normal state using the **Color selection** dialog box.
Icons border	Select a color for the menu icons, labels, and buttons for when an end user mouses over the menu using the **Color selection** dialog box.

5. Configure the landing page’s **Search Box**, **Application Background**, and **Breadcrumb Trail** as described in the following table:

Field	Description
Background	Select a background color. Click either **Single color** or **Gradient**. Customize by clicking the color boxes beside the radio buttons and choosing from the **Color selection** dialog box. Customize the Gradient level by entering a value between 0–360 in the **Degree** field.
Icon/Text	Select a color for the font or icons using the **Color selection** dialog box.

6. Configure the **Login/Logout** page as described in the following table:

Field	Description
Background	Select a background color. Click either **Single color** or **Gradient**. Customize by clicking the color boxes beside the radio buttons and choosing from the **Color selection** dialog box. Customize the Gradient level by entering a value between 0–360 in the **Degree** field.
Text	Select a color for the font using the **Color selection** dialog box.
Welcome	Type a welcome message for the login page.

7. Click **Save**.

13.20.1 Enabling a Smart View Theme in a Multi-Repository Environment

If the target repository for your Smart View theme is different from the Global Repository, follow these steps:

1. Install all related DARs on the Global Repository.
2. Configure dfc.properties on client configuration to point to Global Repository as well, if it is not already set.
3. Log into the Global Repository using client configuration as install owner.
4. Create a Smart View theme as described in this section (it will be the theme across all Repositories using this Global Repository) and don't choose any **Applications**.
5. Map the theme against the default context in the Creation Matrix.
6. Restart the repository and appserver.

13.21 Configuring an Internal Widget

Internal widgets are user interface components used to display internal repository information.

Note: Widgets that you create for the Smart View client need to be referenced in the Smart View landing page configuration file before they appear on the landing page. See "Designing a Smart View and Mobile Landing Page" on page 260 for more information.

1. Navigate to **Widget view** > **Widget** from the menu bar.
2. Click **New** to create a widget.

If you want to create a child widget that inherits the properties of an existing widget, select a widget and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this widget applies. For example, adding the `QA` application would cause the widget to only apply to matching quality assurance cases.
Label <*language*>	Type a label.
Description <*language*>	Type a description.
Widget type	Select the widget type from the list box containing internal widgets. List of Widgets contains more information about the widgets.
Preview file (Classic client only)	Select the preview image to use as a thumbnail preview. If you do not select or import a thumbnail, a default image displays.

4. You can configure a customized set of default columns for a selected widget.

   Note: You can only configure the widgets that allow users to customize column preferences in the client. This includes all content widgets such as Document List, Favorites, Checkout, Recycle Bin, Locations, Versions, Renditions, Relations, Relation Properties, Virtual Document, Virtual Document Snapshot, Publication, Workflow Task List, Task Attachment, and Thumbnails Gallery widgets. Rest of the widgets do not allow column configuration.

   a. Select the **Customize default widget columns** to change the standard column configuration.

   b. Select the **Type** to list all the properties or aspects of the selected document type.

   c. Use the list controls to add, remove, or reorder the properties of the default columns.

      You can click the **Restore Default Columns** button to reset the column configuration to the original configuration for the selected widget type.

      Note: The administrator should also add these default column properties to the **Display Configuration** to avoid any issues during column customization in the client. For example, if a user removes a column that is not part of the **Display Configuration**, that column cannot be restored.

d. For Classic client, select the **Apply default repository content assistance columns from this widget** to show the columns from the selected widget in the content assistance dialog. This checkbox has no effect in the Smart View client. When the user opens a targeted browse, the columns shown in the assistance dialog reflect the current container the user is accessing, or, when the user's current location cannot be determined, fall back to the default container doclist columns as defined in the Smart View landing page configuration file.

Note: Classic client only: Workflow document and attachment picker will honor doclist column selection only if **Apply default repository content assistance columns from this widget** is checked in the widget configuration. If more than one widgets with this checkbox checked are mapped in the configuration matrix, only the first widget in the matrix will be used to show the columns in the content assistance dialog.

e. You can configure a column as the default sort column in the `DoclistWidget`. Any non-repeating attribute of `dm_sysobject` can be configured as a sort column from the default column list by selecting the **Include as default sort column** option. If no columns are configured in the `DoclistWidget`, the default sort columns are: object_name ascending and r_object_id ascending.

Once added, the direction of the attribute is ascending, by default. Use the drop down list in the **Sort Direction** section to select ascending/descending. Date attributes are treated as oldest first/newest first.

Note: The order is always top down. Only if there is a conflict in the previous attribute is the next attribute with the direction considered.

Use the operation buttons to reorder the list of sort columns or remove a column from the list.

f. To configure the maximum number of doclist/docgallery results in the `DoclistWidget`:

i. In the **Maximum Results Filtering** section, set the **Limit max results to** number. If specified with any valid number above 0, all the doclist/ docgallery results are truncated to this number. If no number is entered here, OpenText Documentum CM falls back to the global maximum result size, as specificed in the *`maxResultsSetSize`* paramter of the `d2fs.properties` file.

ii. **For Classic client only:** The **Customize max results filter columns** drop down lists the filter columns when overall results exceed the configured maximum result size. Any non-repeating string-based `dm_sysbobject`, `dm_folder` attributes can be configured in this section by selecting the type from the drop down and moving the attribute to right section. Use the operation buttons to reorder the list of filter columns to change how they appear in the filter drop down. If none are configured, the default filter columns are: `object_name`, `title`, `subject`, `a_status`, `r_creator_name`, and `r_modifier` of the `dm_sysobject`.

Notes

- When users enter text to filter results, items that contain the text anywhere in the string and not just at the start of the string display.
- These settings can be configured per widget.

To update the column types for a list of preferences object, with or without default recreation, run `D2JobPreferencesUpdate` job. For more information, refer to "Default Column Preferences for the Client" on page 331.

5. For Classic client, you can add and configure events that cause a widget to become active:

   a. Use the list controls for **Add focus events** to add, remove, or reorder actions and events. List of Widget Communication Channels contains a list and description of the events, actions, and their parameters.

   b. Select an event to configure its **Additional event parameters**.

      For example, you can add and select the event `D2_EVENT_WIDGET_FOCUS` to a `TasksWidget` and set the parameter and value `widgetType==TaskFoldersWidget`. When the end user selects the Tasks browser widget, the Workflow tasks widget displays if the widget exists in the current workspace view.

      To differentiate between instances of the same widget, use the parameter `config==` and set the value to the **Name** of the widget. For example, you can create a Document browser widget with the name `Word document browser` and another browser widget with the name `PDF document browser`. If you use `widgetType==BrowserWidget` the focus event for both widgets triggers. If you use `config=="Word document browser"` you can specify the specified widget.

6. If you are configuring a `SearchFormWidget`, select a configured query form from the **Query form** list box.

   Each instance of the search query form widget can only pertain to a single type of query form. Use a naming convention that makes it clear to both administrators and end users that the widget is configured for a specific query form. For example, you can use the widget name `SQF QA General Documents` to indicate that the widget uses the QA General Documents query form.

7. If you are configuring a `DocgalleryWidget` or a `DoclistWidget`, fill out the **Parameters** fields as described in the following tables.

   Classic and Smart View client parameters:

Field	Description
Enable breadcrumb	Select to show a container path breadcrumb at the top of the widget. **Note:** If your Classic client workspace does not include a browser tree widget or your Smart View client doclist is not configured to show a tree view, you can **Enable breakcrumb** to allow the user to navigate back to a parent folder.
Hide search bar	Select to hide search toolbar from the Doclist, DocGallery, and SearchWidget widgets.
Content type	Select **DQL** to configure the widget to show a list of content based on a DQL query.

Field	Description
DQL Query	Type the DQL query the widget uses for retrieving its list of content. You can use the `$value` wildcards: • `'$value(r_object_id)'`: Add the selected content. • `'$value(all)'`: Add filtered content. You can use `'$value(all)'` when the widget shares a context with a filter configuration that is set to show all versions. The system then finds and replaces `'$value(all)'` with `(all)` at runtime. Therefore, you must place `'$value(all)'` after the type in the DQL query. For example: `from` *<type>* `'$value(all)'` • `'$value(default_attributes)'`: Add the attributes you want shown as columns. • `'$value(preferences_attributes)'`: Add user preference attributes. The DQL generated by `'$value(preferences_attributes)'` includes a preceding comma, for example `,object_name,a_status,r_modify_date,r_modifier` As a result, `'$value(preferences_attributes)'` must never: • Be the only attribute in the DQL query. • Be the first attribute in the DQL query. • Have a comma between the preceding attribute and itself in the DQL query. If the DQL query is invalid, the system ignores the query and does not show an error message to indicate a problem with the query. The widget shows an empty list of content. **Notes** • If you use the custom attribute in the query, it is mandatory to use `r_object_type` in DQL select query. • If you want DQL search results in a Query form on a Doclist to engage with other widgets, you

Field	Description
	will need to add `r_object_id` to the query in DQL for Query form or check **Add mandatory attributes in hidden mode**.
Override event subscribe	Select to override the events for which the widget refreshes.
Subscription events	Use the list controls to add or remove the events that trigger the widget to refresh.
Override event publish	Select to override the events for which the widget outputs information.
Loading events	Use the list controls to add or remove the events that trigger when the widget loads.
Select events	Use the list controls to add or remove the events that trigger when the end user selects content. Make sure the event matches the content type being shown by the widget. For example, if you configure the widget to show folders, you must use `D2_EVENT_SELECT_FOLDER` or the widget shows an error. If you configure the widget to show a mix of content, use `D2_EVENT_SELECT_OBJECT` or create a custom event to work with the widget.
Double-click events	Use the list controls to add or remove the events that trigger when the end user double-clicks content.
Browser widget starting path	If the Classic workspace has a browser widget using a starting path, then when this Classic doclist is configured to display home cabinet or favorites, it must also have the same starting path. This field is not used by Smart View.

Smart View client parameters:

Field	Description
Disable comment column	This is a checkbox used by the Smart View client only. Select to remove the user comment column from Smart View client doclist widgets.

Field	Description
Enable tree view	This is a setting used by the Smart View client only. Select to turn on the tree view navigation pane for this doclist in Smart View. The user can toggle the view in the interface.
Enable thumbnail view	This is a setting used by the Smart View client only. Select to turn on the content thumbnail gallery view for the doclist in Smart View. The user can toggle the view in the interface.
Default view mode	This is a setting used by the Smart View client only. Select the default appearance of the list of documents in a document list widget. These are the options: • List: Displays a list of filenames. • Pinned: Displays a list of filenames and details in a customizable table. Column widths can be resized in this view and it supports horizontal scrolling. • Grid: Displays thumbnails of files. **Enable thumbnail view** must be enabled for this option.
Starting path	This is a setting used by the Smart View client only. Input a valid path, existing at runtime, as a starting point for browsing with the doclist widget. For example, entering /My Cabinet will start the doclist within the specific My Cabinet cabinet location. Use `$USER` to add a user's login name to the path. Use `$USERhome` to add a user's home directory to the path. **Note:** Locate will not be possible if the location path does not match the starting path.
Disable location column	This is a setting used by the Smart View client only. Select to remove the location column from Smart View client widgets such as the Doclist, Renditions, and Relations.

8. If you are configuring a `BrowserWidget`, type the path to a cabinet or folder in the **Starting path** field. The end user can only browse the entered cabinet or folder when using the widget.

9. If you are configuring a `PropertiesWidget`:

   a. To open properties in edit mode instead of view mode, select **Enable auto edit**.

   b. To broadcast a `D2_ACTION_REFRESH_DOCUMENT` event, select **Enable refresh**. Widgets that listen to the refresh event do not automatically change locations even if the content moves to a different folder based on the updated property values input by the end user.

      If you do not select this option, the widget broadcasts the more resource-consuming `D2_ACTION_LOCATE_OBJECT` event. If the content moves to a different folder based on the updated property values input by the end user, widgets that listen to the locate event automatically move to the new location of the content.

10. If you are configuring a `FavoritesWidget` or `DetailsLocationsWidget`, define the **Folder selection** and **Double-click on folder** parameters by selecting the desired behavior for each:

    - **Open Folder**: Opens the target folder.
    - **Locate Folder**: Sends a locate event for other widgets to act on.
    - **Open and Locate Folder**: Opens the target folder and sends a locate event.

11. If you are configuring a `BrowserWidget`, `DoclistWidget`, or `DocgalleryWidget` you can expose the user’s personalized items in the widget navigation, which allows them to access often-used items more quickly. The following parameters are available:

    - **Show home cabinet**: adds the user’s home cabinet as a selectable node in the widget navigation.
    - **Show favorites folders**: adds a **Favorites** node in the widget navigation that will include the users Favorited folders.
    - **Show favorites documents**: (not available in `BrowserWidget`): Adds a **Favorites** node in the widget navigation that will include the users Favorited documents.

12. If you are configuring a `RecentlyAccessedWidget`, by default, matrix evaluation determines what events and objects to list in the widget. To customize this behavior, select an audit config from the list.

    This is a setting used by the Smart View client only. For more information, see “Auditing and Monitoring“ on page 217.

13. If you are configuring a `WorkflowOverviewWidget`, additional Smart View client parameters are available that allow you to control how Workflows are represented in Workflow tiles on the Landing Page. It is possible to configure

several different widgets (with different labels) so that multiple Workflow tiles appear on the Landing Page. For example, you could create a My Workflows tile that includes only workflows that a user is personally involved in, and All Workflows that allows broad access to existing workflows.

Existing Classic client `WorkflowOverviewWidget` instances can be used for the Smart View client, but they need to be updated with `D2WorkFlowOverviewColumnsMethod` using DA or idql:

```
method='D2WorkFlowOverviewColumnsMethod',arguments='-docbase_name
d2repo'
```

Note: The `-docbase_name` argument is mandatory for the method.

- **Label**: the **Label** you provide for the widget becomes the name of the Workflow tile in the Smart View client's landing Page.
- **Workflow templates**: this control allows you to select existing workflow templates in the system to be used for search with the All Workflows query form. Additionally, this field also limits the list of displayed workflows and summary for a workflow supervisor operating in the context of My Workflows.
- **Limit user searches to**: this control limits whose workflows can be searched using the All Workflows query form. If nothing is selected for this field, workflows of any user in the system could be found using the query form.

14. If you are configuring a `AdminCheckedOutDocumentWidget` complete the following Smart View parameters:

    - **Disable location column**: Select to remove the location column from the widget.
    - **Default view mode**: Select the default appearance of the list of documents in a document list widget. These are the options:
        – List: Displays a list of filenames.
        – Pinned: Displays a list of filenames and details in a customizable table. Column widths can be resized in this view and it supports horizontal scrolling.
        – Grid: Displays thumbnails of files. **Enable thumbnail view** must be enabled for this option.
    - **DQL Predicate**: The portion of the DQL statement after the `FROM` clause. Here is the syntax: *<type_name>* `WHERE` *<condition>*

      For example: `dm_document WHERE a_status != 'Approved'`
    - **Minimum permit**: Defines the filtering of objects with minimum permission when using the DQL predicate.

    Note: Configuring a `AdminCheckedOutDocumentWidget` requires the Smart View client to be approved as a privileged client. To approve it, the administrator must run the `d2privilegedclient` utility for Smart View. For help, see "d2privilegedclient" on page 621 in the Utilities Reference.

15. Click **Save**.

Related Topics

"Configuring a Client Theme" on page 255

"Configuring a Workspace for Classic View" on page 246

"Configuring an External Widget" on page 286

"Configuring Client Menus" on page 293

"Configuring Client Column Preferences" on page 329

"Configuring Display Filters" on page 326

"Configuring Custom Icons for Rendition Formats and Content Types (Classic)" on page 319

"Configuring Icons and Text Color for Content Properties in List Widgets (Classic)" on page 321

13.22 Configuring an External Widget

External widgets are user interface components used to display external web information using a third-party web application. You can configure external widgets to communicate with content properties.

1. Navigate to **Widget view** > **Widget** from the menu bar.
2. Click **New** to create a widget.

   If you want to create a child widget that inherits the properties of an existing widget, select a widget and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description for the widget in client configuration.
Applications	Add or remove the applications to which this widget applies. For example, adding the `QA` application would cause the widget to only apply to matching quality assurance cases.

Field	Description
Label <*language*>	Type a label.
Description <*language*>	Type a description for the widget to appear in the Client Widget Gallery.
Widget type	Select **ExternalWidget**.
Widget url	Type either a static or dynamic URL. **Note:** Add `encodeDollarData=true` to any Widget URL. If this value is not added, special characters will not be displayed properly. Add `locale=$LOCALE` to any Widget URL to use the same language currently used in the Classic client interface.
Bidirectional communication	Select if the widget communicates with the portal using an Open Ajax Hub. **Note:** : Ensure that you specify the full URL for the external widget to work on any browser.
Communication channel	Use the list controls to select a communication channel for transmitting information between widgets and repositories. List of Widget Communications Channels contains further information.
Preview file	Select the preview image, if any, to use as a thumbnail so the end user has an idea of what the widget looks like. If you do not select or import a thumbnail, a default image displays.

Field	Description
Unload Protect	Users can set or clear unload protect attribute in client configuration. If you set unload protect attribute to TRUE, then the system blocks certain operations that unload the IFRAME on external widgets, and loses the state. The region collapse and widget drag or drop options are disabled. The option to drag the tabs is disabled for external widgets. However, you can drag the other tabs in the same container. The option to collapse the border layout is disabled. If the user attempts to collapse a region that contains a widget that is enabled with unload protect, then the region displays “Operation is blocked by an external Widget” error.

4. Click **Save**.

Note: If you invoke *D2_ACTION_DM_TICKET_GENERATE*, an encrypted *DM_TICKET* is generated, which external widgets can use to connect to OpenText Documentum CM. This encrypted *DM_TICKET* is only known by OpenText Documentum CM and the App Server decrypts it before passing it to DFC > Content Server. The encrypted *DM_TICKET* cannot be passed to Content Server directly. If you want to send the *DM_TICKET* from OpenText Documentum CM to Content Sever directly, you must generate a plain *DM_TICKET* by invoking *D2_ACTION_DM_TICKET_GENERATE* with *is_secure* set to `false`.

By default, *is_secure* is set to `true`.

Smart View behavior

External widgets are supported in Smart View with the following functionality:

- Users can access configured external widgets from a toolbar menu item. To add this to the toolbar, you must configure a **MenuSVExternalWidgetList** dynamic menu item.
- External widgets appear as a side panel.
- The widgets can listen to the following events:
  - `D2_EVENT_SELECT_OBJECT`: Get the information of an object.
  - `D2_EVENT_SELECT_OBJECTS`: Get the information of multiple objects.
  - `D2_EVENT_DM_TICKET_GENERATED`: Get the `dm_ticket` generated by Smart View using the `D2_ACTION_DM_TICKET_GENERATE` send event.
- The widgets can publish the following events:
  - `D2_ACTION_DM_TICKET_GENERATE`: Request Smart View for a new `dm_ticket`.

– `D2_ACTION_SEARCH_DOCUMENT`: Request a search.

Related Topics

"Configuring a Client Theme" on page 255

"Configuring a Workspace for Classic View" on page 246

"Configuring an Internal Widget" on page 276

"Configuring Client Menus" on page 293

"Configuring Client Column Preferences" on page 329

"Configuring Display Filters" on page 326

"Configuring Custom Icons for Rendition Formats and Content Types (Classic)" on page 319

"Configuring Icons and Text Color for Content Properties in List Widgets (Classic)" on page 321

13.23 Understanding the Content Transfer Framework

The java-free Content Transfer Framework (WSCTF) has been developed to provide rich content transfer features in OpenText Documentum CM without using java applets. The thin client as a content transfer operation mode is still available, with limited functionality as described below. The framework is installed with OpenText Documentum CM on the application server. A browser-specific extension and the CTF Native Application is installed on end user machines at first login.

The available modes are:

- **None**: sets a *thin client* as the content transfer plugin. This is limited native functionality that is available through the client browser.

  Supported in: Classic and Smart View clients.

- **WSCTF**: sets Client Manager version 2 as the content transfer plugin. This is a browser extension-free implementation that uses web sockets to communicate with the app server.

  Supported in: Classic and Smart View clients.

Note: Downloading large files or large quantities of files in thin client mode could impact performance. To download large quantities of files, it is recommended to use WSCTF mode or set up a special customization that is designed to manage deep exports.

You can define which browser modes are available to clients by configuring `settings.properties`. See *Configuring OpenText Documentum CM Client* in the *Installation Guide* for more information. If you do not define mode(s) in `settings.`

`properties`, None (thin client) will be the default mode for all clients. On individual clients, you can check which mode is active in **Info** > **User Settings** in Classic or **My Profile** > **User Settings** in Smart View.

See “Content Transfer Framework supported feature list” on page 290 for more information on the content transfer features that are supported in each mode.

Note: Microsoft Edge browser is supported in Client Manager v2 mode, but to use it, each client needs to enable **Automatically Detect Intranet Network** in **Control Panel > Internet Options > Security > Local Intranet > Sites**.

13.24 Content Transfer Framework supported feature list

The OpenText Documentum CM Content Transfer framework is supported in both the Classic and Smart View clients, with some differences in individual feature support, as shown in the following table:

Action	Thin client behavior (Classic and Smart View)	Client Manager (WSCTF) behavior (Classic)	Client Manager (WSCTF) behavior (Smart View)
User experience during file edit and check-in	Uses standard browser file download, so a user is typically prompted to open or save the file. The exact behavior depends on the browser. During check-in, the user is prompted to locate the file before the upload.	Classic allows users to set a directory for checkout, seamlessly downloads a file to that directory, and opens it for editing. On check-in, Classic looks in the same directory and uploads the file without prompting the user to find it. The user convenience use cases for Classic client streamline the following features: • Add Version • Checkout • Cancel Checkout • View/Edit • Download	Smart View allows users to set a directory for checkout, seamlessly downloads a file to that directory, and opens it for editing. On check-in, Smart View looks in the same directory and uploads the file without prompting the user to find it. The user convenience use cases for Smart View streamline the following features: • Add Version • Checkout • Cancel Checkout • View/Edit • Download

Action	Thin client behavior (Classic and Smart View)	Client Manager (WSCTF) behavior (Classic)	Client Manager (WSCTF) behavior (Smart View)
Folder Export	No support	WSCTF not required. Supported in the following Browsers: Supported in Chrome Supported in Firefox Supported in Edge Supported in Safari	No support
Copy link in clipboard	Supported	WSCTF not required. Supported in the following Browsers: Supported in Chrome Supported in Firefox Supported in Edge Supported in Safari	WSCTF not required. Supported in the following Browsers: Supported in Chrome Supported in Firefox Supported in Internet Explorer Supported in Edge Supported in Safari
Native annotations	No support	Supported in Chrome Supported in Firefox Supported in Edge	No support
File comparison for MS Office documents	No support	Supported in Chrome Supported in Firefox Supported in Edge	No support
Folder Import	Supported in Smart View Client, not Classic client Supported in Chrome Supported in Firefox Supported in Edge Supported in Safari	Supported in Chrome Supported in Firefox Supported in Edge Supported in Safari	WSCTF not required. Supported in the following Browsers: Supported in Chrome Supported in Firefox Supported in Edge Supported in Safari

Action	Thin client behavior (Classic and Smart View)	Client Manager (WSCTF) behavior (Classic)	Client Manager (WSCTF) behavior (Smart View)
Folder drag-in from desktop	Supported in Chrome Supported in Firefox Supported in Edge Supported in Safari	WSCTF not required. Supported in the following Browsers: Supported in Chrome Supported in Firefox Supported in Edge Supported in Safari	WSCTF not required. Supported in the following Browsers: Supported in Chrome Supported in Firefox Supported in Edge Supported in Safari
File drag-in from desktop	Supported through standard HTML5 thin client drag and drop and file upload functionality.	Supports	WSCTF not required. Supported in the following Browsers: Supported in Chrome Supported in Firefox Supported in Internet Explorer Supported in Edge Supported in Safari
File drag-out to desktop	No support	WSCTF not required. Supported in the following Browsers: Supported in Edge running in Internet Explorer mode	No support
File drag-in from Microsoft Outlook (emails, emails with attachments, or just attachments)	No support	Supported in Chrome Supported in Firefox Supported in Edge Supported in Safari	WSCTF not required. Supported in the following Browsers: Supported in Chrome Supported in Edge Supported in Safari
File compression during content transfer	No support for upload. Support for download via OpenText Documentum CM app server, but not when transfer uses ACS or BOCS.	No support	No support

Action	Thin client behavior (Classic and Smart View)	Client Manager (WSCTF) behavior (Classic)	Client Manager (WSCTF) behavior (Smart View)
Parallel Streaming Download from ACS/BOCS	No support	Supports	No support
Bulk import of files with supporting XML files to populate properties	No support	Supports	No support
Export values from Property Page	Supports, browser handles the download.	Supports	No support
Print content using right-click or inline action menu	Content is downloaded so that the user can print it using the native application.	Content is printed directly to default printer	Content is downloaded so that the user can print it using the native application.

13.25 Configuring Client Menus

You can modify the position, label, display conditions, and contents of menus.

Note: The **Cut**, **Copy**, and **Paste** menus are internally hidden by default and cannot be made visible on the user interface. If you need to have **Cut**, **Copy**, and **Paste** menu items visible on the **Main** menu <*“Content”*>, you must create these new menu entries and remove the existing ones.

1. Navigate to **Go to** > **Menu Classic** from the menu bar to configure the following menus:

   - Menu bar (New, Import, Content, User, Info)
   - Right-click on Location widget
   - Right-click on Publishing Components widget
   - Right-click on Publishing Overlays widget
   - Right-click on Publishing Details widget
   - Right-click on Relation widget
   - Right-click on Rendition widget
   - Right-click on Snapshots widget
   - Right-click on Version widget
   - Right-click on Task detail widget
   - Right-click on a list widget
   - Right-click on Admin Dictionary widget

- Right-click on Admin Group widget
- Right-click on Admin Taxonomy widget
- Right-click on Admin User widget
- Right-click on Delegation widget
- Right-click on Distribution widget
- Right-click on TOC Editor dialog
- Right-click on Multisearch widget
- Right-click on Publishing Manager widget
- Right-click on Publishing Editor widget
- Right-click on SaveSearch category dialog
- Right-click on Tasks list widget
- Right-click on Task folder widget
- Right-click on Search widget
- Right-click on VD widget
- Right-click on Snapshot in VD widget
- Right-click on Workflow widget
- Right-click on Recycle Bin Admin widget
- Right-click on Recycle Bin widget

2. Click **New** to create a menu.

   If you want to create a child menu that inherits the properties of an existing menu, select a menu and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this menu applies. For example, adding the `QA` application would cause the menu to only apply to matching quality assurance cases.

Note: The default menu items don’t support adding duplicate menu conditions from the server side. From the client configuration user interface-side, they still allow adding multiple `$value` menu conditions, but they only work for custom menus (menu items created using the **Menu** or **Java menu** button in client configuration menu configuration).

4. You can add menu buttons as described in the following table:

Action	Description
Add separator	Add a line to partition the menu.
Add menu item	Add a menu entry.
Add sub-menu item	Add a menu as a submenu that an end user can view by expanding the parent menu entry.
Add dynamic menu item	Add a menu entry that runs a Java class. You can only add dynamic menu items to context menus. For more information about available dynamic menus, see "Dynamic Menu Config Java Class Reference" on page 301.

5. Configure menu entries:

   a. Configure the appearance of the selected menu entry as described in the following table:

Field	Description
Hide this entry	Select to hide the menu items of the specified application. **Note:** This setting does not apply to dynamic menu items.
Label <*language*>	Type a label.
Panel Size	Controls the size of the pane that appears when performing certain menu actions, such as uploading a file or initiating a lifecycle. The available options are: • **S**: Small. The pane fills 1/3 of the screen. • **M**: Medium. The pane fills 2/3 of the screen. • **L**: Large. The pane fills most of the screen. If you do not select a size, the default size is **S**.

   b. Configure the actions assigned to the selected menu entry and relevant parameters as described in the following table:

Field	Description
Action	Select a Javascript action for the button to perform. Based on the action selected, a list of associated parameters is populated. You need to select a value for these parameters to perform the selected action. For example, selecting `Publish event : "$value(event)"` action populates the **Event** parameter, and you need to select an event (such as `D2_ACTION_COPY_LINK_IN_CLIPBOARD_LOCATE`) to perform the selected action.

Field	Description
Message	Type additional parameters to be read by services. This field is used to extend menu action behaviors for some actions or service customizations. **Example 1**: Selecting `Show dialog with manager : “CreateFolderManager”` from the **Action** field allows you to define the **Type** of folders the user can create in the client (including a default folder type). • `No message`: default behavior • `root_object_type`: display object type and any children folder types • `only_object_type`: limit folder type to only that type and do not display folder type selection dialog • `list_object_type`: display only the listed folder types • `default_object_type`: can be combined with `root_object_type` and `list_object_type` using `!!` to indicate what value the folder type selection dialog will default to. • The following syntax sample would populate the **Type** field with three different folder type selections, with folder type 2 as the default: list_object_type==<dctm folder type1>,<dctm folder type2>,<dctm folder type3>!!default_object_type==<dctm folder type2> **Note:** You will need to include `!! locateAndRefresh==true` to facilitate locate and refresh functionality in doclist widget in client configuration. When this parameter is added, the doclist will be refreshed to show your users the new folder they have created. **Example 2**: Selecting `D2 Lifecycle Change State Operation $(operation) using the target state $value(targetState)` from the **Action** field allows a `dialog==Document properties`

Field	Description
	message to be shown, where `Document properties` is the property page name.
Sub menu	Select a menu to be shown as a submenu. For example, selecting `MenuContextAdminTaxonomy` from the submenu list box adds the most recent version of the menu for **Right-click on Admin Taxonomy widget** as a submenu.
Java class	Select the Java class to run if the menu item is dynamic.
Locate content and refresh state upon action	When you are configuring a Lifecycle action right-click menu, set this flag to control if a locate action is performed along with the user indicated action. Default is `True`. Setting this flag (along with **Refresh state** and **Refresh widget** to `True`) to `False` allows the user to remain in the current location with focus on the last object selected in the doclist. The status of the document also remains unchanged or is not refreshed.
Virtual Document Handling	Select the Virtual Document root and descendant check-in behavior: • **Root**: default behavior where only the root document is checked in. • **Root & Descendants**: automatic check in of the root and all of its descendants without prompting the user. • Both **Root** and **Root & Descendants** checked: Virtual Document check in dialog appears to the user, allowing them to select their preferred check-in flow.

c. Configure the display conditions of the selected menu entry as described in the following table:

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Display conditions</td><td>Use display conditions to define the rules and scope of the menu entry.

Click Browse to add display conditions.

Click Delete to remove display conditions.

Use the list controls to move conditions up and down the list.

Notes
Some conditions are not universally applicable to all contexts. For example, the condition The Workflow is running will not appear in the right-click menu of a document that is involved in a workflow. This condition only applies when the user right-clicks a workflow in the Workflow widget.
You can add multiple <code>'$value()'</code> menu conditions, but these only work for custom menus–menu items created using the Menu or Java menu option in client configuration menu configuration.</td></tr>
<tr><td>Display disabled instead of hidden</td><td>Select whether to show the menu entry as grayed out or hide the menu entry from view.</td></tr>
<tr><td>Invert the condition</td><td>Select to invert the condition.

For example, you want to have a particular cabinet shown to everyone except for users who have the consumer role. You set the display condition <code>user has client capability equals consumer</code> and then select to invert. The display condition then changes to <code>user has not client capability equals consumer</code></td></tr>
</table>

d. If you select a display condition that uses a DQL query, fill out the three fields as described in the following table:

Field	Description
Condition valid on	Select whether the DQL query applies: • If at least one selected content fits the condition. • If all selected content fit the condition.
Property	Select the property for the DQL query.
Value	Type the parts of the DQL query after the `select` and `from` conditions. The DQL query must use the wildcard `'$value(r_object_id)'` to refer to the selected object or objects. For example: `dm_document (all) where r_object_id='$value(r_object_id)' and a_status='Draft'` The system automatically generates the `select` and `from` parts of the DQL query.

6. Drag and drop menu items to reorder.
7. Click **Save**.

Related Topics

“Configuring a Client Theme” on page 255

“Configuring a Workspace for Classic View” on page 246

“Configuring an External Widget” on page 286

“Configuring an Internal Widget” on page 276

“Configuring Client Column Preferences” on page 329

“Configuring Display Filters” on page 326

“Configuring Custom Icons for Rendition Formats and Content Types (Classic)” on page 319

“Configuring Icons and Text Color for Content Properties in List Widgets (Classic)” on page 321

13.26 Dynamic Menu Config Java Class Reference

Dynamic menus are menus that use a Java class to populate the list of options in the menu. For example, a list of available workflows for a given document.

The following table lists which dynamic menus you can use and where they are supported.

Java class name(s)	Supported clients	Summary
MenuBWSTypesList	Smart View	Creates a dynamic menu with a list of Smart View workspaces that a user can create, depending on their access.
MenuContextTasksList	Smart View Classic	Creates a dynamic menu with a single child menu based on a found `IDfQueueItem` in the supplied context. Sets the action and label attributes consistent with discovered workflow actions, either FORWARD or REJECT.
MenuDocumentC2Export MenuDocumentC2Print MenuDocumentC2View	Classic	Creates a dynamic menu with child items for each found, valid config menu item from the supplied context, for the specified config type (Export, Print, View). The child menu items have an incrementing ID attribute like `genid_x` and a label attribute derived from the config item's label string. The Action attribute is derived from the relevant action object (Export, Print, View) combined with the config name resulting in an action specific to the config type.

Java class name(s)	Supported clients	Summary
MenuDocumentD2Distribution	Smart View	Creates a dynamic menu with one child item for each config name that is common to all objects known to the supplied context. Sets the menu item label to the label associated with the config name and sets the action to be performed to `prepareDistribution(`*<config name>*`)`
MenuDocumentD2LifeCycleAttach	Smart View Classic	Provides dynamic menu items labeled for each Applicable Entry State name. The action attribute is set to perform an `operationD2LifeCycle` specifying the transition start config string and the state name.
MenuDocumentD2LifeCycleNextStates	Smart View Classic	Provides a dynamic menu option listing and allows action on next state names, confirmations, dialogs, actions, and events for each next state name derived from the current context.
MenuDocumentD2Workflow	Smart View Classic	Creates a dynamic menu with entries representing each `D2WorkflowConfig` object common to the set of objects represented by the current context. Each menu item is composed of a label representing the object's label attribute (or, if there is no label, an alternate subject or object name is used) and an action attribute that performs `launchD2Workflow(`*<current item's object name>*`)`

Java class name(s)	Supported clients	Summary
MenuDocumentLifeCycleAttach	Smart View Classic	Queries the dm_policy table to select object name, object_id, subject, and definition_state. Creates a dynamic menu with one child node per row returned by the query, setting the menu label to the object_name, the status to the subject, and the action to operationLifeCycle('Attach', <<*object_id*>>) unless the definition_state is not equal to 2, in which case, the menu item is marked as disabled.
MenuSVExternalWidgetList	Smart View	Allows a menu item to call an external widget. External widgets are user interface components used to display external web information using a third-party web application.
MenuToolsMassUpdate	Smart View Classic	Builds up XMLNodes representing each potential item retrieved in the query. When one of the dynamic menu items is selected, massUpdate(objectName) is run.
MenuUserD2SVLandingPageConfigs	Smart View	Adds a dynamic list of landing pages to the User menu. The dynamic menu lists all the configured landing pages that are enabled in the configuration matrix.

13.27 Configuring Smart View client Menus

You can modify the display conditions and contents of various menus in Smart View client.

To understand differences in menu element support (such as Actions) between Smart View and Classic, see "Smart View: Menu Config Actions and Events Support" on page 518.

Note: New default menu actions, their order, or placement within any menu are not automatically applied to existing menu configurations when you upgrade. If you have existing menu configurations, you can create a new menu to view and compare new defaults and then either add, change, or move new elements to your existing configurations or create new menu configurations to replace your existing ones.

1. Navigate to **Go to** > **Menu Smart View** from the menu bar to configure the following menus:

   Detail contextual menus

   These menus are accessible through contents' **Property** pages.

   - Relation widget
   - Rendition widget
   - Version widget

   Contextual menus

   Action Toolbars
   Activated when a user moves the pointer over content, or selects a checkbox beside content.

   The Inline Action Toolbar appears when the user moves their pointer over content in a variety of UI contexts (in a doclist grid view, or in a minimized Landing Page tile, for example). This pop-up menu shows several action icons, and ends, if required, with an overflow button that displays the remaining applicable actions. The number of icons that appears in the Inline Action Toolbar is dynamic between a maximum of 5 and a minimum of 2, including the overflow icon (when needed) and based on the configuration of available actions, permissions, and the number of columns in the node list view.

   The **Send to workflow** menu item appears in the root of Action Toolbars, and exposes a dynamic sub-menu of Workflows that are applicable to the user and the content they selected. See "Understanding Workflows" on page 146 for more information on creating Workflows that appear in this sub-menu.

   Note: The default Smart View Action Toolbar menu item **Download** is required for the Mark Offline feature of the mobile app. Removing the **Download** menu item from this and other toolbars will cause **Mark Offline** to disappear from applicable mobile menus. The menu item must also be promoted.

+ Menu
Accessible from doclist containers, where the menu can be activated by clicking a **+** icon. This menu is generally used to add new items to the doclist (content or folders, for example).

If your users require the ability to indicate an individual object will be used as a source for inheritance during the creation process, then it is possible to add and use the existing **+ Menu** items for Add Item or Import File to the Action Toolbar menu.

User Menu
Accessible from the **User Profile** menu. Includes **About OpenText Documentum CM**, where users can view version information.

VDoc Action Toolbar
Accessible from the Virtual Document Outline user interface, when the user is viewing a Virtual Document.

Note: If a `vdoc-widget` is not defined in the Smart View Landing Page configuration xml file, the menu items that are established in the VDoc menus will not appear to the end user. See “Understanding Default Widgets in the Smart View Landing Page Config” on page 266 for more information.

Tasks
Accessible from the various Tasks views (detail view, list view) as actions the users can perform on individual Tasks.

Note: If a `tasks-widget` is not defined in the Smart View Landing Page configuration xml file, the Task Details load using URL navigation (by bookmark or shared URL, for example) will not succeed, and if `task-attachment-widget` is not configured, then Task Details page will not display. See “Understanding Default Widgets in the Smart View Landing Page Config” on page 266 for more information.

Default Task menu items related to Task Notes and Supporting Files can be removed from the user’s view in the Smart View client if you disable the **Notes** and **Supporting Files** tabs in the workflow configuration. See “Configuring a Workflow” on page 149 for more information.

Task Documents Actions Toolbar
Accessible from the **Working Files** and **Supporting Files** tabs in the Task Details and Workflow Details views. These are actions users can perform on the documents that appear in those tabs.

Note: If a `tasks-widget` is not defined in the Smart View Landing Page configuration xml file, the Task Details load using URL navigation (by bookmark or shared URL, for example) will not succeed, and if `task-attachment-widget` is not configured, then Task Details page will not display. See “Understanding Default Widgets in the Smart View Landing Page Config” on page 266 for more information.

Default Task menu items related to Supporting Files can be removed from the user’s view in the Smart View client if you disable the **Supporting Files** tab in the workflow configuration. See “Configuring a Workflow” on page 149 for more information.

Workflow Actions Toolbar
Accessible from the Workflow list view and the Workflow detail view. These are actions that users can perform on active Workflows.

Collections
Accessible from the container menu, the collection’s title on the **Collections details** page. Allows users to perform actions on the collection.

Collection Items Actions Toolbar
A menu for items in a collection, accessible when users move the pointer over content or selects the checkbox besides content. Allows users to perform actions on the contents of a collection.

Workspace Toolbars
Accessible from within a workspace. Allows the user to perform actions on the workspace.

Workspace Item Toolbars
A menu for items within a workspace, accessible when users moves the pointer over content or selects the checkbox besides content. Allows users to perform actions on the workspace content.

Workspace + Menu
Accessible from within a workspace, where the menu can be activated by clicking a **+** icon. Allows user to add new items or folders to the workspace.

Note: Promoted actions appear in the menu list before non-promoted actions.

Additionally, in order for the Comment column to appear in doclists, a Comment menu item must be added to the Action toolbar, and it must be promoted. See “Comment menu item” on page 312 for more information.

2. Click **New** to create a menu.

   If you want to create a menu that inherits the properties of an existing menu, select a menu and click **Create from**.

3. Fill out the menu form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this menu applies. For example, adding the `QA` application would cause the menu to only apply to matching quality assurance cases.

4. Add individual menu elements by clicking the following toolbar buttons:

Note: You can re-order menu entries and separators by dragging and dropping them to different locations within the same menu.

Button	Action	Description
	Add separator	Add a line to partition items in the menu. Select an existing item in a menu, then click **Add separator**. The separator appears after the selected menu item.
	Add menu item	Add a menu entry.
	Add sub-menu item	Adds a sub-menu item to one of the existing Detail contextual menus or Contextual menus. **Note:** Menus are restricted to three levels in the Smart View client (for example, a sub-menu of a sub-menu is not permitted). Sub-menu items are always non-promoted.
	Delete menu item	Delete an existing menu item or separator.

Button	Action	Description
	Add dynamic menu item	Add a menu entry that runs a Java class. The dynamic menu item supports a number of Java classes, including the following: • `MenuDocumentD2Workflow` • `MenuContextTasksList` • `MenuUserD2SVLandingPageConfigs` An example of a dynamic menu is the listing of workflows that appears in the default action menu **Send to Workflow**. For a full list of available dynamic menus, see “Dynamic Menu Config Java Class Reference” on page 301.

5. Configure individual menu entries:

   a. Click a menu entry in one of the Detail contextual menu or Contextual menu drop-down lists. Configure the appearance of the selected menu entry as described in the following table:

Field	Description
Hide this entry	Select to hide the menu items of the specified application. **Note:** This setting does not apply to dynamic menu items.
Label <*language*>	Type a label for the menu entry that appears to end users..
Shortcut	This is not supported in the Smart View client.

   b. Configure the actions assigned to the selected menu entry and relevant parameters as described in the following table:

Field	Description
Action	Select a Javascript action for the menu selection to perform. Based on the action selected, a list of associated parameters is populated. You need to select a value for these parameters to perform the selected action. For example, selecting `Publish event : "$value(event)"` action populates the **Event** parameter, and you need to select an event (such as `D2_ACTION_COPY_LINK_IN_CLIPBOARD_LOCATE`) to perform the selected action.
Message	Type additional parameters to be read by services. This field is used to extend menu action behaviors for some actions or service customizations. See "Examples of common Smart View menu additions" on page 312for some implementation examples.
Sub menu	Select a menu to be shown as a submenu. For example, selecting `MenuDocumentShare` from the**Sub menu** list box adds the most recent version of the menu for sharing documents to an external service as a submenu.
Locate content and refresh state upon action	When you are configuring a Lifecycle action right-click menu, set this flag to control if a locate action is performed along with the user indicated action. Default is `True`. Setting this flag (along with **Refresh state** and **Refresh widget** to `True`) to `False` allows the user to remain in the current location with focus on the last object selected in the doclist. The status of the document also remains unchanged or is not refreshed.
Alternate online Help Server URL Root	Configure an alternate online help server and options for Smart View client by entering a URL root. **Note:** By default, the Smart View help is hosted by OpenText. You should not need to modify this setting.

Field	Description
Entry is promoted	This checkbox is only valid for the contextual menu on these widgets: Doclist, Checkout, Recently Accessed, Favorites and Search results. In other widgets like Rendition, Relations, Versions and other menus this setting is ignored and the actions are always treated as promoted. When selected, applicable menu entries will appear first in the Smart View client. Entries that are not promoted will appear later, according to their relative locations in the menu list described at the beginning of this topic. Best practice for menu building is to place no more than 10 promoted menu entries at the top of a menu list to ensure efficient display in the client. **Note:** A menu entry with the 'Comments' action must be marked as promoted to be shown in the doclist. If it is not promoted, comments appear only in the doclist widget toolbar.

c. Configure the display conditions of the selected menu entry as described in the following table:

Field	Description
Conditions	Use display conditions to define the rules and scope of the menu entry. Click **...** to add display conditions. Click **X** to remove display conditions. Use the list controls to move conditions up and down the list.
Display disabled instead of hidden	**Note:** This checkbox is not currently supported in the Smart View client. Select whether to show the menu entry as grayed out or hide the menu entry from view.
Capability	Select the user role associated with this menu entry.

Field	Description
Invert the condition	Select to invert the condition. For example, you want to have a particular cabinet shown to everyone except for users who have the consumer role. You set the display condition `user has client capability equals consumer` and then select to invert. The display condition then changes to `user has not client capability equals consumer`
Node	Select the node target if you have selected the Condition **Selection is into node with**.

d. If you select a display condition that uses a DQL query, fill out the three fields as described in the following table:

Field	Description
Condition valid on	Select whether the DQL query applies: • If at least one selected content fits the condition. • If all selected content fit the condition.
Property	Select the property for the DQL query.
Value	Type the parts of the DQL query after the `select` and `from` conditions. The DQL query must use the wildcard `'$value(r_object_id)'` to refer to the selected object or objects. For example: `dm_document (all) where r_object_id='$value(r_object_id)' and a_status='Draft'` OpenText Documentum CM automatically generates the `select` and `from` parts of the DQL query.

6. Drag and drop menu items to reorder.

7. Click **Save**.

13.28 Understanding promoted and non-promoted actions

Promoted items are returned first for display in a Smart View menu before non-promoted items. When the menu is displayed the first time the order in configuration is not maintained, a second request is made to retrieve the non-promoted items. After the menu is displayed the second time, the entries appear as configured. This helps improve menu display performance by ensuring only the most frequently used and important actions are immediately available while secondary actions are loaded for any given object in a just-in-time manner. Best practice for menu building is to place no more than 10 promoted menu entries at the top of a menu list, then non-promoted items to ensure efficient display in the client.

13.29 Examples of common Smart View menu additions

Many Smart View menu selections are provided by default. Additional menu items will need to be added and customized by you. Here are some examples of common custom menu items and how they are added to the **Menu Smart View** interface.

13.29.1 Comment menu item

In order for the Comment column to appear in doclists, a Comment menu item must be added to the Action toolbar, and it must be promoted.

In the Action Toolbar, add a new item with the following elements:

Label: Add Comment

Conditions: Not required (optional)

Action: `Show dialog with manager`

Manager: `CommentManager`

Entry is promoted: Selected

13.29.2 Add Folder menu item

Adding this item to the Action Toolbar allows you to define the **Type** of folders the user can create in the client (including a default folder type).

Label: Add Folder

Action: `Show dialog with manager : “CreateFolderManager”`.

Message: this field is used to define the folder types:

`No message`: default behavior

`root_object_type`: display object type and any children folder types

`only_object_type`: limit folder type to only that type and do not display folder type selection dialog

`list_object_type`: display only the listed folder types

`default_object_type`: can be combined with `root_object_type` and `list_object_type` using `!!` to indicate what value the folder type selection dialog will default to.

Example **Message** syntax:

The following syntax sample would populate the **Type** field with three different folder type selections, with folder type 2 as the default:

```
list_object_type==<dctm folder type1>,<dctm folder type2>, <dctm folder
type3>!!default_object_type==<dctm folder type2>
```

13.29.3 Preset Profile menu item

Adding this item to the **+** menu allows you to include a configured Preset Profile for uploading new material to the repository.

Label: [A label that describes the type of content the preset handles]

Action: `Show dialog with manager : “CreateManager”`.

Message: This field is used to select the Presets you have configured.

The following syntax sample would add a Preset Profile to the menu:

```
presetConfigName==[Name of your preset profile from config]!!
presetUniqueID==[Preset Unique ID from config]
```

13.29.4 Open URL menu item

Allows you to add a URL link as an Action Toolbar menu item.

Label: [A label that describes the link target]

Action: `Publish event : “D2_ACTION_OPEN_URL”`

Message: allows you to pass parameters that control the target URL, a $value, and browser window features like width and height.

The following syntax sample would add a menu item that would allow you to search Google for the selected repository `object_name`, with defined browser window dimensions and name:

```
url==https://www.google.com/?q=$value(object_name)!!windowName==gsearch!!
windowFeatures==width=500,height=700
```

Additionally, the URL provided can be relative to the current app server. In the following sample, the URL would open and run a search defined by an `objectID`:

```
url==/D2-Smartview/ui/#search/by-id/0801b2768000969f
```

13.29.5 Dump menu item

Allows you to add an Action Toolbar menu that dumps content data for diagnostic purposes.

Label: Dump

Action: `Publish event:”D2_ACTION_DUMP”`

13.29.6 Send to Workflow sub-menu workflow selections

This default item in the Action menu displays all of the workflows appropriate to the logged in user in a dynamic submenu. Users can choose a workflow from the submenu to initiate a workflow for the logged-in user and selected content.

13.29.7 Workflow history menu item

This entry allows you to add a Workflow History menu item that shows workflow activity for selected content. You can add this menu item to Action toolbars, VDoc Action toolbar and Task Documents Actions toolbar.

Label: Workflow History

Action: `Publish event:"D2_ACTION_SHOW_WORKFLOW_HISTORY"`

Condition: Object has workflow history

13.29.8 Lifecycle direct state transition menu item

This entry allows you to add a menu item that users can access to move content directly to a lifecycle state, skipping the usual progression of steps. You can add this menu item to Action toolbars, VDoc Action toolbar, Relations or Versions toolbars, task doc action toolbar, and workflow action toolbar..

Note: As a prerequisite, each state you want to add as a direct menu item must be designated as a **Direct state** in the lifecycle configuration, as described in "Configuring a Lifecycle" on page 136.

Label: *<A suitable name for the lifecycle state>*

Action: `Lifecycle change state operation: "<name of operation>" using the target state "<name of target state>"`. For example, `Lifecycle change state operation: "Detach" using the target state "State1"`

Operation: `<Name of lifecycle operation>`. For example, `Detach`, `Promote`, `Demote`, `Suspend`, `Resume`, or `Start`.

Target state: *<name of target state>*. This name must match the state in your lifecycle configuration.

(Optional) **Locate content and refresh state upon action**: *<<true>>* or *<<false>>*, enable this selection if you want the user to be notified of where the content is located and what the state is after the direct state transition.

13.29.9 Brava! Compare menu items

If you have the Brava! Enterprise viewer installed and configured for use with the Smart View client, you can add several menu items that allow users to compare two pieces of content in the viewer.

Label: Text Compare, Graphical Compare, or Compare With

Action **Action**: `View selected content(s): "D2_ACTION_CONTENT_VIEWER_COMPARE_TEXT"`, `View selected content(s): "D2_ACTION_CONTENT_VIEWER_COMPARE_GRAPHIC"`, or `View selected content(s): "D2_ACTION_CONTENT_VIEWER_COMPARE_WITH"`

Parameter **Action**: (one parameter action is required per Action, to match the compare target (text, graphical, or compare with) `D2_ACTION_CONTENT_VIEWER_COMPARE_TEXT`, `D2_ACTION_CONTENT_VIEWER_COMPARE_GRAPHIC`, or `D2_ACTION_CONTENT_VIEWER_COMPARE_WITH`

Conditions: `Selection can be viewed` and `Selection has content`.

13.29.10 Intelligent Viewer compare menu items

If you have Intelligent Viewer installed and configured for use with the Smart View client, you can add several menu items that allow users to compare two pieces of content in the viewer.

Label: Text Compare, Graphical Compare, or Compare With

Action **Action**: `View selected content(s): "D2_ACTION_CONTENT_VIEWER_COMPARE_TEXT"`, `View selected content(s): "D2_ACTION_CONTENT_VIEWER_COMPARE_GRAPHIC"`, or `View selected content(s): "D2_ACTION_CONTENT_VIEWER_COMPARE_WITH"`

Parameter **Action**: (one parameter action is required per Action, to match the compare target (text, graphical, or compare with)`D2_ACTION_CONTENT_VIEWER_COMPARE_TEXT`, `D2_ACTION_CONTENT_VIEWER_COMPARE_GRAPHIC`, or `D2_ACTION_CONTENT_VIEWER_COMPARE_WITH`

Note: Before adding the Intelligent Viewer compare menu items, they must be enabled in the Intelligent Viewing widget (see “Configuring OpenText Intelligent Viewing“ on page 409).

13.29.11 View item in a new window menu items

You can configure a menu item to allow a user to open an item in a new window with a specified viewer.

Label: Open in a new window, Open with <*viewer*> in a new window

Action **Action**: `:view selected Content(s)`

Parameter **Action**: `:D2_ACTION_CONTENT_VIEW_POP_OUT`

Message: `:viewer==<viewer-widget-name>`

Notes

- <*viewer-widget-name*> is the name of the viewer as defined in D2-Config. For example, if the widget is named `myBravaCSRWidget`, then you would use `myBravaCSRWidget` in place of <*viewer-widget-name*>. Brava Enterprise, Brava CSR, and Intelligent Viewing viewers are all supported.
- If you do not specify which viewer to use, the item opens with the default viewer.
- Users can save or share the URL of the new window to allow easy access to the item.

13.29.12 Add to collection menu items

You can configure a menu item to allow a user to add an item to a collection.

Label: Add to collection

Action **Action**: `Publish event : "D2_ACTION_ADD_TO_COLLECTION"`

Parameter **Event**: `D2_ACTION_ADD_TO_COLLECTION`

13.29.13 Add a static or dynamic list of landing pages to the User menu

You can configure the User menu to include additional landing pages. You can manually add each landing page or create a dynamic list of all configured landing pages.

Fixed list example:

Label: Admin

Action **Action**: `Switch to Smartview landing page config : "admin_landingpage"`

Parameter **Smartview landing page config**: `admin_landingpage`

Add all of the landing pages you want to appear in the User menu.

Dynamic list example:

Label: <dynamic landing page list>

Java class: com.emc.d2fs.dctm.menu.dynamicmenus.MenuUserD2SVLandingPageConfigs

13.30 Defining content on click opening behavior in the Smart View client

When a user clicks content in the Smart View client interface, the content opens in the default viewer. If the viewer is not available, or the content cannot be displayed by the viewer, the browser downloads the content. This behavior can be configured with several different opening actions, which can be ranked in priority order.

1. Navigate to **Go to** > **Menu Smart View** from the menu bar.
2. Select an existing menu, or click **New** to create a new menu.
3. In the **On click defaults** section, select content opening **Actions** from the list and move them to the **Defaults** box in your preferred order of processing by clicking **>**. For example, placing **View in place viewer** first, then **Download**, will attempt to show the content in the Smart View in place viewer as the primary action. If that is not possible, the content will then be downloaded by the browser.

The following **Actions** are available:

- **Display overview**: The document overview screen is shown with content summary information. If the user clicks **Open** from this screen, the document opens in a viewer (if supported), or is downloaded.
- **Download**: The document is downloaded by the user’s browser. PDF export configuration and WSCTF behaviors are honored.
- **Download, override PDF Configuration**: The document is downloaded by the user’s browser. PDF export configuration rules are not applied. WSCTF behaviors are honored.
- **View, override PDF Configuration**: The document opens in the viewer if the file type is supported, and is downloaded if it is not. PDF view or export configuration rules are not applied. WSCTF behaviors are honored
- **View in place viewer**: The document opens in the preview in place viewer if the file type is supported, and is downloaded if it is not.
- **View in mini viewer**: The document opens in a preview pane which can be resized.
- **View in full viewer**: The document opens in the full feature viewer if the file type is supported, and is downloaded if it is not.
- **View native content**: The document opens in its original format. For example, if a PDF rendition of a Word document exists, the PDF opens by default. This options opens the original format of the document.

13.31 Customizing document link behavior in Smart View

You can customize the behavior of document links that are created using **Copy link** and **Share** in Smart View.

By default, these links open a document overview page where a user can access the current version of the document. If the link was to an older version of the document, the user can access the older version by clicking **View original** under the version number listed on the page.

To change this behavior, you can use the following parameters:

- `launchViewerIfCurrent=true` : If the link is to the current version of the document, automatically open the document based on viewer configuration.
- `preventAutoSwitchToCurrent=true` : If a link is sent to an older version, the use can access that version from the overview page by default. To access the current version, the user can click **View current**.

Add these to the parameters to the message field of the **Copy link** or **Share** menu configuration. Here is an example using both parameters:

```
add_url_params==launchViewerIfCurrent=true&preventAutoSwitchToCurrent=true
```

Note: Do not use !! to construct a URL.

For more information on configuring a menu item, see "Configuring Smart View client Menus" on page 304.

13.32 Configuring Custom Icons for Rendition Formats and Content Types (Classic)

You can configure custom icons for renditions formats and content types in the list and thumbnail widgets. If OpenText Documentum CM finds that the configuration for the content type and rendition format conflicts for given content, it uses the configuration for the content type.

You can create a custom icon configuration with or without property styling. Configuring Icons and Text Color for Content Properties in List Widgets contains more information on configuring property styles.

1. Navigate to **Widget view** > **Graphics** from the menu bar.

2. Click **New** to create a graphics configuration.

   Note: In the configuration matrix, you can configure graphics only for group or user contexts.

   If you want to create a child graphics configuration that inherits the properties of an existing graphics configuration, select a graphics configuration and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name.
Description	Type a description.
Applications	Add or remove the applications to which this graphics configuration applies. For example, adding the `QA` application would cause the graphics configuration to only apply to matching quality assurance cases.

4. Use the list controls to add, remove, and reorder rendition formats. The system gives icon priority to lower rows. Fill out the row for each rendition as described in the following table:

Field	Description
Format Name	Select the rendition format from the list box.

Field	Description
List Icons	Select an icon for list widgets from the list box. If you want to import a new icon, click **Browse** and fill out the **Import** dialog box. Use a 16x16 pixel image at 72 DPI.
Thumbnail Icons	Select an icon for thumbnail widgets from the list box. If you want to import a new icon, click **Browse** and fill out the **Import** dialog box. Use a 64x64 pixel image at 72 DPI.

5. Use the list controls to add, remove, and reorder content types. The system gives icon priority to lower rows. Fill out the row for each content type as described in the following table:

Field	Description
Type Name	Select the type from the list box.
List Icons	Select an icon for list widgets from the list box. If you want to import a new icon, click **Browse** and fill out the **Import** dialog box. Use a 16x16 pixel image at 72 DPI.
Thumbnail Icons	Select an icon for thumbnail widgets from the list box. If you want to import a new icon, click **Browse** and fill out the **Import** dialog box. Use a 64x64 pixel image at 72 DPI.

6. Click **Save**.

Related Topics

“Configuring a Client Theme” on page 255

“Configuring a Workspace for Classic View” on page 246

“Configuring an External Widget” on page 286

“Configuring an Internal Widget” on page 276

“Configuring Client Menus” on page 293

“Configuring Client Column Preferences” on page 329

“Configuring Display Filters” on page 326

"Configuring Icons and Text Color for Content Properties in List Widgets (Classic)" on page 321

13.33 Configuring Icons and Text Color for Content Properties in List Widgets (Classic)

You can configure an icon and color-coding for content properties in the columns of list widgets in Classic. For example, you can add an icon to the document status property and change the text color to show blue for `Draft`, green for `Accepted`, and red for `Rejected`.

You can configure property styling with or without custom icon configurations. Configuring Custom Icons for Rendition Formats and Document Types contains more information on custom icon configurations.

1. Navigate to **Widget view** > **Graphics** from the menu bar.
2. Click **New** to create a property style.

   If you want to create a child property style that inherits the properties of an existing property style, select a property style and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name.
Description	Type a description.
Applications	Add or remove the applications to which this property style applies. For example, adding the `QA` application would cause the property style to only apply to matching quality assurance cases.

4. Use the list controls to add and remove properties. Fill out the row for each property as described in the following table:

Field	Description
Property	Type the name of the property.
Value	Type the value of the property. For example, you can type `a_status` in the **Property** and `Draft` in the **Value** to configure a style for the `Draft` lifecycle state.

Field	Description
Icon	Select an icon from the list box. If you want to import a new icon, click **Browse** and fill out the **Import** dialog box. Use a 16x16 pixel image at 72 DPI. For example, if you configure the `Draft` value to show a pencil icon, the document status column shows the icon followed by the value.
Icon only	Select this option to replace the property value with the icon. For example, if you configure the `Draft` value to show a pencil icon and select **Icon only**, the document status column only shows the icon. The end user can still filter and search content based on the property because the option hides but does not remove the value.
Text Color	Select the color of the property text using the **Color selection** dialog box. You do not need to select a color if you selected **Icon only** because the value does not display.

5. Click **Save**.

Related Topics

"Configuring a Client Theme" on page 255

"Configuring a Workspace for Classic View" on page 246

"Configuring an External Widget" on page 286

"Configuring an Internal Widget" on page 276

"Configuring Client Menus" on page 293

"Configuring Client Column Preferences" on page 329

"Configuring Display Filters" on page 326

"Configuring Custom Icons for Rendition Formats and Content Types (Classic)" on page 319

13.34 Configuring Icons and Text Color for Content in List Widgets (Smart View)

You can configure an icon for content formats, types, and properties in the columns of list widgets in Smart View. Additionally, you can configure color-coding based on content properties.

For example, you can configure a specific icon for PDF files or change the text color of files based on a status property (blue for `Draft`, green for `Accepted`, or red for `Rejected`).

You can access these configuration options from **Widget view** > **Smart View Graphics**.

The **Graphics** list on the left of the page lists all of the graphic configurations. You can select a configuration and modify it, or create a new configuration by clicking **New**. You can also create a new configuration that inherits the settings of an existing configuration by clicking **Create from**.

This table describes the graphic configurations properties:

Field	Description
Name	The name of the graphic configuration.
Parent name	If the graphic configuration was created from an existing configuration, that configuration appears here.
Description	The description of the graphic configuration.
Applications	Add or remove the applications to which this property style applies. For example, adding the `QA` application would cause the property style to only apply to matching quality assurance cases.
Classic graphic configuration	Select a Classic graphic configuration to import those settings. Changing these settings in the Smart View configuration does not affect the Classic configuration.

You can configure the icons for content based on format, type, and properties and color-coding based on properties. You can also create conditional phrases for when to apply these formatting options. You can get details on these conditional phrases below.

Click the **Format**, **Type**, or **Properties** tab to customize the formatting options for these content details.

The following table describes the graphic options:

Option	Description
Format name Type name Property	Select which format, type, or property to configure its icon or color-coding.
Condition	Create a conditional phrase to control when the formatting applies to the content. See below for more information on these conditional phrases. If you do not enter a condition, the formatting applies to all instances of the selected format, name, or property. You can create multiple conditions. Click **+** to add a condition and **x** to delete a condition. These conditions are evaluated in order, from top to bottom. Reorder them using the up and down arrows.
Icon	Display an icon if the item meets the condition. For example, if you configure the `Draft` value to show a pencil icon, the document status column shows the icon followed by the value. Click **...** to select a file from your computer. Use a 16x16 pixel image at 72 DPI. Only `.svg` files are supported.
Icon only (Properties)	Select this option to replace the item name with the icon if it meets the condition. For example, if you configure the `Draft` value to show a pencil icon and select **Icon only**, the document status column only shows the icon. The end user can still filter and search content based on the property because the option hides but does not remove the value.
Color (Properties)	The color of the item name if it meets the conditions.
Background (Properties)	The background color of the item name if it meets the condition.

Note: Graphic configurations do not apply to:

- Virtual documents
- Content in search results unless the format, type, or property that controls the graphical configuration is included in the search result columns.

Writing a conditional phrase

Here are some basic rules to compose a conditional phrase:

- Conditions must be formatted with a left attribute, an operator, and a right attribute or value. For example, `status == 'Draft'` would check if the status attribute of an item is `Draft`. Values must be wrapped in single or double quotation marks.
- A condition phrase can consist of multiple conditions using either logical "OR" `||` or "AND" `&&` operators.
- You do not need to define the attribute if your conditional phrase uses a property of `dm_sysobject`. To use a property of another attribute, you must define it before the property. For example, to use an employee number property, you could use `employee_records.employee_number`, where `employee_records` is a subtype of `dm_document` and `employee_number` is property defined within `employee_records`.
- Define an aspect type using the same format: aspect name and then attribute. For example, `subject == miplabelinfo.first_mip_label` would check if `first_mip_label` of `miplabelinfo` matches the subject.
- To add a list of repeating attributes, use a `|` pipe symbol. For example, `emp_records.skillset == 'Java|Python|Machine Learning'` would check for files with the *`skillset`* attribute that includes Java, Phyton or Machine Learning.
- When using numbers for values, do not use quotation marks.

List of operators

- `==` : Equals
- `!=` : Does not equal
- `>` : Greater than
- `<` : Less than
- `>=` : Greater than or equals
- `<=` : Lesser than or equals
- `>=` : Greater than or qquals
- `contains` : attribute includes this attribute or aspect
- `not contains` : attribute does not include this attribute or aspect
- `begins with` : attribute starts with this attribute or aspect
- `ends with` : attribute ends with this attribute or aspect

Date formatting

- All dates are in the following format: `MM/DD/YYYY`. For example, `05/28/2028`.
- You can use `$TODAY` to represent the current date.

- You can calculate the difference between two dates using the following format: *<inputDate>* `date diff` *<compareToDate>*. Where:
  - `inputDate` can only be an attribute name (for example, `emp_records.doj` or `emp_records.last_promoted_date`) and cannot be a direct value (for example, `'03/04/2024'` or `'$TODAY'`).
  - `compareToDate` can be either an attribute or a direct value.

  For example:
  - `emp_records.doj date diff '$TODAY' >=1830 && emp_records.doj date diff '$TODAY' <=3660`
  - `emp_records.doj date diff emp_records.last_promoted_date <= 1826`

Null or Empty keywords

You can create a conditional phrase that use the null or empty keyword. For example:

- `(dm_document.r_aspect_name == null || dm_document.r_aspect_name == empty) && r_modify_date == '$TODAY'` : Includes all items modified today with no aspect.
- `(emp_records.skillset != null || emp_records.skillset != empty) && r_object_type == ‘emp_records'` : Includes all documents of the `emp_records` type with a defined skillset.

Upper and Lower keywords

By default, conditions are case sensitive. You can define attributes to be case sensitive using the `Upper` and `Lower` keywords. For example:

- `UPPER(title) contains ‘MANAGER’`
- `LOWER(emp_records.domain) contains ‘hr’`
- `UPPER(title) begins with UPPER(domain)`
- `LOWER(emp_records.skillset) contains ‘employee'`

13.35 Configuring Display Filters

You can create and configure display filters that end users can select in the client. Display filters affect the content shown in the Document list widget. For example, you can create a quality assurance filter that only shows content labeled with the keyword QA. You can create a quality assurance Draft filter that only shows content labeled with the keyword QA with the status Draft.

Note: The filter configuration is limited to the basic Document list and Collections widgets and is not fully implemented for other extended Documentation list widgets such as the Favorites, Versions, or Locations widgets.

In Smart View, depending on the applied filter configuration, facets will be displayed, as follows:

- If the applied filter configuration includes a DQL criteria and the document list size exceeds the *maxResultSize*, facets will not be displayed.
- If the applied filter configuration includes a DQL criteria and the document list size is within the *maxResultSize*, facets will be displayed.
- if the filter criteria is empty, facets will be displayed, regardless of the document list size.

Note: *maxResultSize* for the doclist widget can be configured either in client configuration (under the **Maximum Results Filtering** section in the widget configuration) or in the `D2FS.properties` file.

1. Navigate to **Go to** > **Filter** from the menu bar.
2. Click **New** to create a display filter.

   If you want to create a child display filter that inherits the properties of an existing display filter, select a display filter and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this display filter applies. For example, adding the `QA` application would cause the display filter to only apply to matching quality assurance cases.
Label <*language*>	Type a label.
Default	Select to make the filter the default setting.
All versions	Select if you want to apply the filter to all versions of content. Clear to apply the filter to current versions.
Show folders	Display folders in the filtered list. Unless enabled, only files display in the filtered list.
DQL filter	Type the DQL query defining the filter.
Type	Select the content type to which the DQL query is applied.

4. Click **Save**.

Notes

- If no filter is configured, the following filters display:
  - All versions
  - Current version
- When a user accesses a document or collections list, the following priority controls which filter is applied:
  - The last filter applied by the user.
  - The default filter.
  - First filter listed in the configuration matrix.
- If a document list has a defined DQL, then it overrules the filter configuration.
- The **Show All Version** option does not appear on document and collection lists because that option is controlled by the filter configuration.

Related Topics

"Configuring a Client Theme" on page 255

"Configuring a Workspace for Classic View" on page 246

"Configuring an External Widget" on page 286

"Configuring an Internal Widget" on page 276

"Configuring Client Menus" on page 293

"Configuring Client Column Preferences" on page 329

"Configuring Custom Icons for Rendition Formats and Content Types (Classic)" on page 319

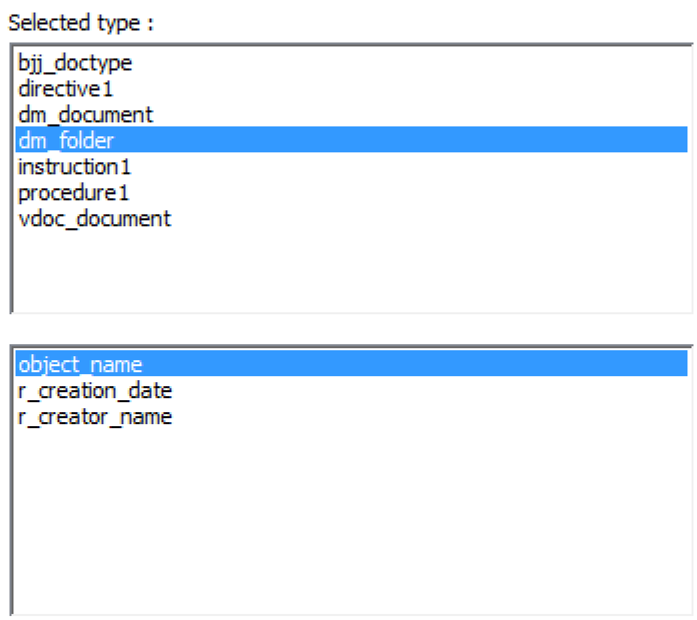
"Configuring Icons and Text Color for Content Properties in List Widgets (Classic)" on page 321

13.36 Configuring Client Column Preferences

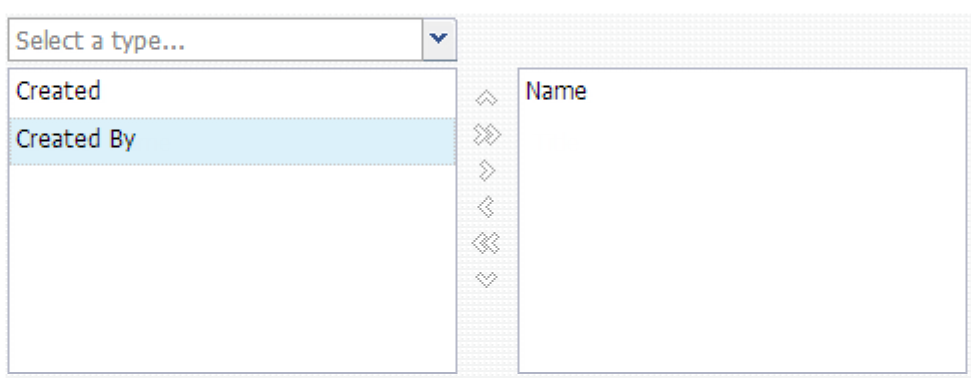
Default columns in the repository for tables display all properties and types. You can configure column preferences to allow users to select columns that filter the table based on property and type.

Note: In Smart View, users can customize columns by accessing **Columns Setting** under the table's **Settings** menu.

For example, the following image is a `dm_folder` content type with its configured properties:



The following image is the resulting column configuration interface as well as the property `Title`, which shows the folder path:



If you do not define a content type for a column, `dm_document` and `dm_folder` are the defaults.

To configure a column preference:

1. Navigate to **Interface** > **Display configuration** from the menu bar.
2. Use the list controls to modify the list of columns that filter based on content type.
3. To add a property filter:
   a. Select a type from **Properties of selected type**.
   b. Use the list controls to modify the list of property filters.
   c. For each property, you can select a dictionary and select an alias or language to configure the display of values within the column.

      For example, you can configure a **Status** column to use the English language setting of a dictionary. Depending on dictionary configuration, values in a different language within the column can be translated into English. You can use dictionaries and aliases to universalize column value structure.
4. Click **Save**.

Related Topics

"Configuring a Client Theme" on page 255

"Configuring a Workspace for Classic View" on page 246

"Configuring an External Widget" on page 286

"Configuring an Internal Widget" on page 276

"Configuring Client Menus" on page 293

"Configuring Display Filters" on page 326

"Configuring Custom Icons for Rendition Formats and Content Types (Classic)" on page 319

"Configuring Icons and Text Color for Content Properties in List Widgets (Classic)" on page 321

13.37 Default Column Preferences for the Client

To update the column types for a list of preferences object, with or without default recreation, run `D2JobPreferencesUpdate` job with the following parameters:

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>default <code>true|false</code></td><td>Set to <code>true</code> to create or destroy the default preferences object and recreate based on the <code>D2cPreferences.properties</code> resource file.

For example: <code>-default true</code>

The default value is <code>false</code>.</td></tr>
<tr><td>user_names <<<i>user_name</i>>, <<i>user_name2</i>>, ...></td><td>Specify a list of users on which the preferences object should have the column updated against the <i>Default</i> instance.

For example: <code>-user_names dmadmin,user1</code></td></tr>
<tr><td>group_names <<<i>group_name</i>>, <<i>group_name2</i>>, ...></td><td>Specify a list of group that is resolved as a user list; on each user, the preferences object should have the column updated against the <i>Default</i> instance

For example: <code>-group_names admingroup, docu</code>

Note: The administrator can either set user_names or group_names, or a combination of both the parameters, with each value as a unique value or a list of comma-separated values.</td></tr>
<tr><td>col_types <<<i>coltype</i>>, <<i>coltype2</i>>, ...></td><td>Specify a list of column types to update, if empty or not present, all column types will be restored to <i>Default</i>.

For example: <code>-col_types default, detail_location</code></td></tr>
</table>

Note: If user has already customized the columns, the widget on the client side considers the current user's customization, even if the administrator configures the default columns per widget type. The `D2JobPreferencesUpdate` job reapplies the column preferences to the users after administrator's consideration.

13.38 Configuring Access to Administration Widgets in the Client

1. Navigate to **Tools** > **Options** from the menu bar.

2. Fill out the form as described in the following table:

   Note: The user administration widget does not allow users to add groups from the Group membership tab unless the logged-in user is an Admin or Super User, even if the logged-in user is part of administration groups.

Field	Description
Access group for Client	Select the user group to have access to the client user interfaces.
Access group for Client-Config	Select the user group to have access to client configuration.

   Note: If a field is left blank, all groups have access.

3. Click **Save**.

Related Topics

"Configuring a Workspace for Classic View" on page 246

"Configuring Access to client user interface and the client configuration" on page 207

13.39 Refreshing and Reloading

You can refresh and reload certain aspects of OpenText Documentum CM using the **Tools** menu in client configuration, as described in the following table:

Action	Description
Tools > Refresh cache	Run this task to rebuild the cache, which is an operation that is normally run when the server is restarted. This action will refresh the client configuration caches, as well as the `DmMethods` and business process manager (BPM) caches across all Java message service (JMS) instances, and send refresh requests to each of the clients in the **Tools > Options... > Clients URL** list.

Action	Description
Tools > Reload Client options	Run this task to reload D2-Config.properties. You can use this task to load the updated D2-Config.properties settings without restarting your application server.

Note: When performing either **Refresh Cache** or **Reload Client options**, the server also performs the same operations for the client URLs and automatically invokes the same operation for `DmMethods` & BPM applications in all JMS instances.

Related Topics

"Configuring a Workspace for Classic View" on page 246

13.39.1 FontConfig Cache Refresh in Java Method Server

After you add or delete a font from client configuration, to ensure that font entries are updated in the `FopFontConfiguration` file and are available for generation in PDF rendition, you must run `D2RefreshCacheMethod` in Documentum Administrator to update the fonts in the temp location of the Java Method Server.

Method name	Parameters
D2RefreshCacheMethod	• docbase_name: *<Name of the docbase>* • cache_names: *`font_config`*

13.39.2 UserGroup Configuration Cache Refresh in Java Method Server

When you add or remove a user from a group, you must run `D2RefreshCacheMethod` in Documentum Administrator to update the landing page configuration.

Method name	Parameters
D2RefreshCacheMethod	• docbase_name: <*Name of the docbase*> • cache_names: `document_set_switch,user_group,d2options` The following describes these cache options: – *document_set_switch*: This cache contains the list of the configurations and their mapped contexts such as landing page and search configurations. – *user_group*: This cache contains info about users and their associated groups. – *d2options*: This cache contains URL information for the caches that need to be refreshed. These are configured in Client Configuration options. **Note:** Cache names must be listed as above, as comma separated values.

13.40 Smart View client-side cache service

Smart View supports a client-side data caching service to offer faster navigation for users. The cache service runs on a user's browser and assists with repetitive data requirements by reducing the number of HTTP data request sent to the web-server when a user quickly navigates between pages.

Prerequisites

- User browsers must support Progressive Web App (PWA) APIs, such as Service Worker, Indexed DB, Cache Storage, Client-Controller-Messaging and Fetch APIs.
- The connection between a browser and the Smart View app-server must employ an SSL encryption configured with an X.509 certificate that can be validated publicly. More simply, the connection requires a HTTPS protocol enabled with a valid certificate.

13.40.1 Configuring a client-side cache service

This configuration controls the dynamic behavior of the cache service. The configuration is a piece of JSON object separated into a file named `sw-config.json`, which is loaded by the cache service after installed on the user's browser. When installing the cache service, the configuration is parsed, processed, and stored in Indexed DB storage. When the cache service updates, these steps are repeated. This section includes the JSON object's format details and a sample configuration.

Notes

- JSON is a data only format and does not support comments. However, any key names that are not listed below are ignored during processing, so can be used for storing additional data like comments.
- JSON format follows a strict syntax. You must ensure the content of `sw-config.json` has valid syntax or else the cache service will fail to process the configuration.
- `sw-config.json` is added as a configmap for cloud deployment. For more information, see the *OpenText Documentum CM Cloud Deployment Guide*.

13.40.1.1 Format

<JSON_ROOT>

Object. It can have the following properties:

Name	Type	Default	Description
enabled	Boolean	false	Controls whether or not the Cache Service is enabled. Useful for system administrators to take control of caching behaviors of all connected browsers.
logging	Boolean	false	Controls verbosity of AJAX interception and response preparation of the cache service. When set to *`true`*, the cache service prints useful log statements in the web console.

Name	Type	Default	Description
permanentCacheEndpoints	Pattern[]	[]	Defines a set of Client REST API endpoint matchers. When an endpoint is matched against this set, the response from web-server is cached across user sessions on the same browser so subsequent requests to the same endpoint are handled by the cache.
stepForwardCacheEndpoints	Pattern[]	[]	Defines a set of Client REST API endpoint matchers. Upon matching a request against this set, the response is generated from cache and simultaneously a web-server request is made to update the cached content when the same endpoint is invoked again. For example, when an endpoint is invoked for the first time, the response is collected from web-server and put to cache and returned. When the same endpoint is invoked again, the response is given from the initial data and a request is made to update the cached data so that it can be used again.

Name	Type	Default	Description
cacheCategories	Object	{}	Defines an arbitrary set of endpoint matchers. Caching behavior for endpoints matched against this set follows a time-based expiration policy. The name of the keys in this object can be anything, however, the values must be a CacheCategory type. See the following table for more information.

CacheCategory

Object. Associates an expiration time to a set of endpoint matchers. It can have the following properties.

Name	Type	Default	Description
ttlInMS	Number (positive integer)	30000	Cache expiration time in milliseconds. Applies to the set of matchers defined by *`endpoints`*.

Name	Type	Default	Description
endpoints	Pattern[]	[]	Defines a set of Client REST API endpoint matchers. A request matched against this set receives a response from cached data (if it exists), up to a period of time defined by *`ttlInMS`* from when it was first cached. A request made after the amount of time in *`ttlInMS`* causes fresh data retrieval from the web-server and an update to the cache for subsequent usage for another period of time defined by *`ttlInMS`*.

Pattern

String. Defines a pattern to match a Client REST API endpoint. Can be set to a simple text string or a string containing JavaScript Regex constructs or a combination of both. For example, `"d2-inbox"`, `"[a-f0-9]{16,}"`, `"users/[^/]+"` are all valid patterns.

Notes

- All patterns are only matched against the path portion of a Client REST API endpoint URL. All patterns are only compared against “ends-with” ($) operator. So a match is found only if a Client REST API endpoint URL's path ends with one of the configured patterns. It also means that an endpoint declared with the intention to find a match against certain query parameters of a Client REST API endpoint URL will not work.
- All patterns that start with a double exclamation mark characters (`!!`) are ignored and could be used as a comment. For example, “`!!d2-inbox`” is ignored by the cache service.

13.40.1.2 Sample

```
{
  "enabled": true,
  "logging": false,
  "permanentCacheEndpoints": [
    "!!sample-comment that starts with double exclamation(!!) followed by any text and is ignored by the system.",

    "services",
    "!!product-info"
  ],
  "stepForwardCacheEndpoints": [
    "notes/count"
  ],
  "cacheCategories": {
    "short": {
      "ttlInMS": 60000,
      "_comment": "Data for endpoints in this category is cached for 60,000 milliseconds i.e. a minute.",
      "endpoints": [
        "d2-preferences",
        "[a-f0-9]{16,}",
        "!![a-f0-9]{16,} pattern matches against all Client REST API endpoint URLs that ends-with an object-id"
      ]
    },
    "medium": {
      "ttlInMS": 180000,
      "endpoints": [
        "users/[^/]+"
      ]
    },
    "long": {
      "ttlInMS": 480000,
      "endpoints": [
        "repositories/[^/]+"
      ]
    },
    "super_long": {
      "ttlInMS": 780000,
      "endpoints: []
    },
    "comment": "We can have as many cache categories as we want"
  },
  "whatever": "This is also ignored and practically serves as a comment"
}
```

13.40.2 Verifying that the cache server is running

1. Open Smart View in a new browser tab.
2. In the same browser tab running Smart View, open the developer tools (usually F12 on the keyboard) and navigate to the **Console** tab.
3. Refresh the browser tab and wait until the landing page refreshes.

   When refreshing, do not use the **Hard reload** option available in Chrome or other browsers because it temporarily disables some PWA components and might cause a false negative.
4. Watch for a warning message that starts with “[SW] Not running due to”. If a similar log statement does not appear, then the cache service is running correctly.

13.40.3 Disabling or bypassing the cache server

There are two options to disable the cache server and ensure that fresh data is used:

- As an administrator, edit the `sw-config.json` for a Smart View deployment and ensure the *`enabled`* property is set to *false* in the configuration. Restart the app server (or container/pod) to disable the cache service for all users connected to this instance of Smart View.
- A Smart View user can add `skip_data_cache=true` in the query parameters when accessing the Smart View URL to bypass the cache service. For example: `https://our.documentum.d2.com/D2-Smartview/ui/?skip_data_cache=true`.

13.41 Configuring Smart View Import/Export

You can configure the Smart View Import/Export feature that allows users to zip and download multiple files together. You can control the size and type of files that the user can download.

Options configured under **PDF Configuration** > **Export configuration** apply to downloaded files. So if you configure an export configuration and a user selects a .docx file to download, the user will instead download a PDF file with the configured properties. For more information on export PDF configurations, see “Configuring Export Configuration” on page 366.

1. Navigate to **Go to** > **Smartview Content Import/Export** from the menu bar.
2. Click **New** to create a configuration.
3. Fill out the form as described in the following table:

Field	Description
Name	Type the name of the content import/ export configuration.
Description	Type a description.
Applications	Add or remove the applications to which the configuration applies.

Field	Description
Maximum total file size in MB	Define in MB, the total allowable size of the selected documents (files) before zipping. For example, if the value is 25, and you select 3 files of 10MB each, this would exceed the threshold and throw an exception to the user. If empty, there is no limit. The Maximum total file size is calculated using the native file size of all selected documents. As such, other applicable configuration elements in context for the user, could apply - like PDF configurations, etc. This could alter the actual final file size.
Maximum number of files	Define the allowable number of files that can be added in one export operation. If empty, there is no limit on the number of files allowed in a .zip download.
Supported formats	Define the allowable dm_format objects can be included in .zip download. If you do not define the allowable formats, all formats are allowed. If the user downloads a folder, only supported files are added to the download.

4. Click **Save**.

13.42 Configuring Business Admin Actions

The Business Admin Actions page controls which additional actions users with the `dmc_business_manager_role_dynamic` role can perform. You can view and edit existing configurations, or create a new one by clicking **Go to** > **Business Manager Actions**.

To enable the Business Admin Actions page, you must register Smart View as a privileged client. For more information, see *OpenText Documentum Content Management - Client Installation Guide (EDCCL-IGD)*.

Here are the configuration options:

Field	Description
Name	Type the name of the business admin actions configurations.
Description	Type a description.

Field	Description
Applications	Add or remove the applications to which the configuration applies.
Actions	Select the action you want to allow the user to perform. Here are the supported actions: • Cancel checkout • Pause workflow • Resume workflow • Abort workflow • Update supervisor **Note:** The Smart View client must be approved as a privileged client to action to work. To approve it, run the `d2privilegedclient` utility. See "d2privilegedclient" on page 621.

DQL Predicate	The portion of the DQL statement after the `FROM` clause. Document-related actions require `dm_sysobject` or its subtypes. Workflow related actions require `dm_workflow`as the object type (type_name).
	Here is the necessary syntax for this field:
	`type_name WHERE condition`
	For example:
	`dm_document WHERE a_status != 'Approved'`

Comment options	Controls comment requirements. • Show comments: When a user performs this action a dialog appears where the user can add a comment. • Comments mandatory: When a user performs this action a dialog appears where the user is required to add a comment.
Add / Remove action	• Click **+** to add a new action below this action row. • Click **x** to delete an action.

These options allow users with the appropriate role to perform actions on content that they do not own. These actions are audited.

Chapter 14

Troubleshooting

14.1 Debugging configuration and content issues using Dump

Users can debug their configuration and content in a few different ways in the clients.

A summary of the dump information appears in a panel and is separated (if applicable) into **OpenText Documentum CM object** and **OpenText Documentum CM configuration** sub-tabs. Click **Copy** to capture the text on your clipboard for pasting into a text file.

In Classic:

- Use the keyboard shortcut `Ctrl + Alt + D` to create a dump of selected content. Right-click and select **Switch** to see the client configuration matrix applied.

  Note: In a content dump (or `getString` on any attributes that are specific to time) the timezone format used is specified in `dfc.time_zone`. You might notice the date in the dump widget is not the same as the actual date in the client user interface.

- Creating and making available in the workspace a widget based on the `DumpWidget` type.

In the Smart View client:

Note: To turn on the dump function, set the `D2FS.properties` file settings `enabledump` and `dumpGroupName`. For more information, see the *OpenText Documentum Content Management Installation Guide* and the file comments.

- Use the keyboard shortcut `Ctrl + Alt + D` to create a dump of selected content, or UI context (such as the main Landing page).
- Position your pointer over content and click **Dump** in the inline action menu.

  Note: The **Dump** Smart View menu item does not appear by default in the inline action menu. For information about adding **Dump**, see "Configuring Smart View client Menus" on page 304.

14.2 Troubleshooting Logs

OpenText Documentum CM uses the following logs for troubleshooting purposes:

- `D2-JMS.log`: Logs the information related to methods that run on the Methods server (JMS). Examples of such methods are *<D2LifeCycleChangeStateMethod>* and *<D2WFUpdateWorkflowAliasesMethod>*.

  An example of an error: If lifecycle changes do not work for documents, refer the D2-JMS.log file for detailed information.

  Note: If you find that the `D2-JMS.log` is getting too large, which might result in performance degradation and failure of JMS methods, make the following edit to the logback.xml file to institute a size-and-time based policy. The logback file can be found at: *`<Install Path of Documentum CM Server>`*`\Tomcatversion\webapps\DmMethods\WEB-INF\classes`

  Change:

  ```
  <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
  ```

  To:

  ```
  <rollingPolicy class="ch.qos.logback.core.rolling.SizeAndTimeBasedRollingPolicy">
  ```

  The following parameters can then be used in the logback file to control size and time (in this example each file would be at most 5MB, 5 days worth of history would be kept, but the total size burden would not exceed 1GB):

  ```
  .log.zip</fileNamePattern>
  <MaxHistory>5</MaxHistory>
  <maxFileSize>5MB</maxFileSize>
  <totalSizeCap>1GB</totalSizeCap>
  </rollingPolicy>
  ```

- `D2.log`: This log is available at the application server level. Logs in this file provide information about methods that run on the client side of the application server. Examples of such methods are *<taskservice>* and *<browserservice>* implementations.

  An example of an error: If the download or checkout of a particular file results in an error, refer `D2.log` for detailed information.

  Note: On RHEL / Jboss 6.4 with DFC jar version 6.7 SP2, the following error is present in the logs:

  ```
  Error Caused by: java.lang.NullPointerException: null at   java.
  util.Hashtable.put(Hashtable.java:514) [na:1.7.0_75] at   com.
  documentum.fc.common.impl.preferences.PreferencesManager
  .readPersistentProperties(PreferencesManager.java:343)
   [dfc.jar:na]
  ```

- `D2-Config.log`: This log is the least used of the three logs. It logs information related to client configuration, such as the xml structure, after a configuration is saved.

An example of an error: If there are any issues related to configurations, such as ‘Not able to save a properties page’ or ‘Not able to create a context’, view the client configuration logs for details.

- `Log4j 2.x`: This log that tracks system activity for troubleshooting purposes. It requires some configuration, described here.

Name

You must name `Log4j 2.x` configuration files as follows:

– `log4j2.xml`

– `log4j2.properties`

`log4j2.xml` example configuration

```
<Configuration status="WARN">
    <Appenders>
        <Console name="ConsoleAppender" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{HH:mm:ss} %-5p %c{1} - %m%n"/>
        </Console>
        <RollingFile name="FileAppender" fileName="logs/app.log"
                     filePattern="logs/app-%d{MM-dd-yyyy}-%i.log.gz">
            <PatternLayout pattern="%d{HH:mm:ss} %-5p %c{1} - %m%n"/>
            <SizeBasedTriggeringPolicy size="10MB"/>
        </RollingFile>
    </Appenders>
    <Loggers>
        <Root level="ERROR">
            <AppenderRef ref="ConsoleAppender"/>
            <AppenderRef ref="FileAppender"/>
        </Root>
        <Logger name="com.yourcompany" level="DEBUG" additivity="false">
            <AppenderRef ref="ConsoleAppender"/>
            <AppenderRef ref="FileAppender"/>
        </Logger>
    </Loggers>
</Configuration>
```

`log4j2.properties` example configuration

```
status = error
name = PropertiesConfig
property.filename = logs/app.log
appenders = console, rolling
appender.console.type = Console
appender.console.name = ConsoleAppender
appender.console.layout.type = PatternLayout
appender.console.layout.pattern = %d{HH:mm:ss} %-5p %c{1} - %m%n
appender.rolling.type = RollingFile
appender.rolling.name = FileAppender
appender.rolling.fileName = ${filename}
appender.rolling.filePattern = logs/app-%d{MM-dd-yyyy}-%i.log.gz
appender.rolling.layout.type = PatternLayout
appender.rolling.layout.pattern = %d{HH:mm:ss} %-5p %c{1} - %m%n
appender.rolling.policies.size.type = SizeBasedTriggeringPolicy
appender.rolling.policies.size.size = 10MB
loggers = root
logger.root.level = error
logger.root.appenderRefs = console, rolling
```

Best practices

– Avoid multiple configuration files. There should only be one `Log4j 2.x` configuration file in the *`classpath`*.

– Set *additivity="false"* to prevent duplicate logs.

Troubleshooting

– Logs do not generate: Verify the file name and location.

– Duplicate logs generate: Check for multiple configurations or if *additivity* is set to `true`.

These log files capture traces for different components of the product. For a particular user action, the different product parts interact together. Therefore, all log files record the related details of the action.

14.3 Enabling D2-Client.log in the Debug Mode

1. Navigate to the logback.xml file in the following location:`<Install path of Web Application Server>\webapps\D2\WEB-INF\classes`
2. Change the level from `info` and `warn` to `debug`.
3. Change the value under the <root> tag from `info` to `debug`.
4. Clear the old logs (after stopping the application server).
5. Start the application server service.

14.4 Enabling D2-Config.log in the Debug Mode

1. Navigate to the logback.xml file in the following location:`<Install path of Web Application Server>\webapps\D2-Config\WEB-INF\classes`
2. Change the level from `info` and `warn` to `debug`.
3. Change the value under the <root> tag from `info` to `debug`.
4. Clear the old logs (after stopping the application server).
5. Start the application server service.

14.5 Enabling D2-JMS.log in the Debug Mode

1. Navigate to the logback.xml file in the following location: *<Install Path of Documentum CM Server>*`\Tomcatversion\webapps\DmMethods\WEB-INF\classes`
2. Change the level from `info` to `debug`.
3. Change the value under the <root> tag from `info` to `debug`.
4. Clear the old logs.
5. Restart the JMS service.

14.6 Setting a refresh delay for file list widgets

In a folder with more than 1000 documents, when a user saves or checks in a document that they have moved or auto-linked to another folder, the file list widget might refresh but still display the previous file location. The document is successfully linked to the new location, but the file list widget displays the old path.

To resolve this issue, you can introduce a delay to the file list widget refresh rate.

To introduce a refresh rate delay to file list widget:

1. Open `settings.properties` and find the **x3_portal** section.

2. Remove the comment tags around `autoLinkAndRefreshLocateDelay=1000`.

14.7 Configuring locate and refresh events for specific widgets

By default, when a user checks in and cancels a checkout, widgets do not refresh with the selected item's location. To set a widget to refresh using the document location, perform the following configuration change:

1. In client configuration, click **Go to** > **Menu** and select the menu you want to modify.

2. Under Action, select the action you want to modify.

3. In the Message field, enter `targetType==`*`<widget name>`*. Replace *<widget name>* with the name of the widget you want to modify.

4. Click **Save**.

14.8 DM_API_E_EXIST login error

If a user deletes a preference file, a **DM_API_E_EXIST** error might occur at login, preventing login. To resolve, you must remove the *`x3_preferences`* and the *`d2c_preferences`* properties from the `d2–cache` file:

1. Open the following file: `tomcat\webapps\D2\WEB-INF\lib\C6-Common-<xx.x.0.jar>\com\emc\common\java\cache\d2-cache.xml`.

2. Search for *`object_ids-cache`* and under `<cacheExtensionFactory>`, remove the *`x3_preferences`* and the *`d2c_preferences`* properties.

3. Restart the application server.

Note: You only need to complete this procedure once to resolve the issue.

Chapter 15

Configuration Troubleshooting and Tips

15.1 Accessing client configuration without Opening a New Browser Window

If you do not want to open client configuration in a new browser window, add the `newWindow` parameter to the client configuration URL and set it to `false`. For example:

```
http://<server>:<port>/D2-Config/?newWindow=false
```

15.2 Email Notification Error when Assigning a Workflow Task to the User

Problem

An email notification error occurs when a workflow task is assigned to the user.

Cause

The way `D2EventSenderMailMethod` works has to adapt to the argument signature of the mailing method in the Documentum CM Server layer which is changed due to a security fix.

As part of the security fix, the argument structure of the default mail method of the Documentum CM Server (*dm_event_sender*) was changed wherein, an additional argument *web_server_port* was added. Before the fix, the *web_server_port* information was fused with another argument (web_server) using a pipe (|).

Resolution

The *dm_event_sender* method needs arguments without pipe (|), hence we need to split the *web_server* argument by removing the pipe and adding an additional argument of *web_server_port*.

Add an environment variable *<addWebServerPort>* and set it to `TRUE` in Documentum CM Server environment to force the splitting of the *web_server* argument.

Note: Above workaround is applicable only for the users who use latest version of Documentum CM Server, if Documentum CM Server is not up-to date, then above configuration not required.

15.3 Unable to Receive the Task after Using Update Performer Action

Problem

The next user cannot receive the task after using **Update Performer** action

Cause

The way `D2EventSenderMailMethod` works has to adapt to the argument signature of the mailing method in the Documentum CM Server layer which is changed due to a security fix.

As part of the security fix, the argument structure of the default mail method of the Documentum CM Server (*dm_event_sender*) was changed wherein, an additional argument *web_server_port* was added. Before the fix, the *web_server_port* information was fused with another argument (web_server) using a pipe (|).

Resolution

The *dm_event_sender* method needs arguments without pipe (|), hence we need to split the *web_server* argument by removing the pipe and adding an additional argument of *web_server_port*.

Add an environment variable <*addWebServerPort*> and set it to `TRUE` in Documentum CM Server environment to force the splitting of the *web_server* argument.

Note: Above workaround is applicable only for the users who use latest version of Documentum CM Server, if Documentum CM Server is not up-to date, then above configuration not required.

Chapter 16

Configuring PDF functionality

16.1 Overview of PDF Configuration

You can extend PDF functionality by adding:

- Extra pages: you can add one or several pages at the beginning, end, or after a given page. These pages can contain content properties or variables.
- PDF layers: you can add headers, footers, or images to all or part of documents.
- Watermarks: you can add image or text watermarks to all or part of documents above or below text and as transparent images.
- Dynamic pages: you can add dynamically-constructed pages, such as signature pages, gradually over the course of the lifecycle of the PDF document. Create dynamic pages in XML format so that tables can be managed based on their size, which are linked according to configurations when the document is viewed, printed, or exported.
- If a PDF configuration exists, it is applied on a PDF rendition of a document. If a PDF rendition is unavailable, Content Transformation Services transforms non-PDF documents to PDF format before applying PDF configurations (view/rendition/print/export).
- You must enable and configure support for sensitivity labels to process documents protected by them, otherwise they cannot be processed by PDF configurations. Support for sensitivity labels is only available on cloud deployments.

Changes are applied to the PDF document according to four configurations: view, print, export, and static conversion. Use static conversion to configure an action that cannot be changed and is carried out once during processing. Static conversions overwrite the current rendition or create a new rendition. Toggle PDF configurations to contexts for end users the same way you configure and enable configuration components. If you enable a PDF View configuration, the system always uses the PDF view instead of the default view.

You can use PDF configurations to secure and apply controls to PDF documents when they are distributed out of the repository. The main functions are:

- Ensuring only secure versions are provided when viewing, exporting, and printing.
- Controlling printing using print counters and adding the ability to print with the note designating the recipient. The system audits the controlled prints using the information provided by the end user when submitting the content for printing.

- Controlling PDF security settings, such as setting passwords for opening the file, applying encryption, and removal of save and print functions.

Notes

- When you attempt to apply an overlay to any PDF page, a sizing check is performed. If your overlay is 10% larger than the width and height of a page, it will not be applied to that page.
- If users have RELATE permission or lower, PDF configurations are invoked when they use the right-click menu item **Export Rendition** to create a rendition of a document. If you have configured this right-click menu to use native content, the document will be produced with all applicable watermarks and overlays. Users with greater than RELATE permissions will get the native or base document.
- Intelligent Viewing supports View and Rendition PDF configurations. Print and Export are not currently supported.
  - View
  - Rendition
- When you attempt to apply an overlay to any PDF page, a sizing check is performed. If your overlay is 10% larger than the width and height of a page, it will not be applied to that page.
- When you import a PDF with sensitivity labels using BOCS in asynchronous mode, PDF configurations do not apply during view/export operations until the content file reaches the CS from BOCS.
- C2 is the legacy name for the PDF Configuration feature. It still appears in a few locations throughout client configuration and in this guide accordingly.

Related Topics

"Formatting Table of Contents using XSL in PDF configurations" on page 380

"Clearing the dmi_queue_item After Creating a PDF Rendition Configuration" on page 381

"Formatting Table of Contents using XSL in PDF configurations" on page 380

"Configuring View Configuration" on page 353

"Configuring Print Configuration" on page 360

"Configuring Export Configuration" on page 366

"Configuring a Static Configuration" on page 373

"Configuring Administrator Access to PDF Configuration" on page 381

16.2 Configuring View Configuration

1. Navigate to **PDF Configuration** > **View configuration** from the menu bar.
2. Click **New** to create a view configuration.

   If you want to create a child view configuration that inherits the properties of an existing view configuration, select a view configuration and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.
3. Fill out the form as described in the following table:

Field	Description
Name	Select a view configuration.
Description	Type a description.
Applications	Add or remove the applications to which this view configuration applies. For example, adding the `QA` application would cause the view configuration to only apply to matching quality assurance cases.
Default configuration	Select to make this configuration the default view configuration. If more than one default configuration is applicable for the end user in a context within the configuration matrix, the top-most configuration in the PDF Configuration view applies.
Compatibility	Select the oldest compatible version of the PDF viewing software.
Fast Web Compatibility	Select to generate linearized PDFs. The generation of fast web-enabled PDFs should only be enabled when strictly required. Producing linearized PDFs requires the documents to be sent to the CTS system for processing, increasing output time. **Note:** This setting is not operational if password protection is enabled with PDF configurations.

Field	Description
PDF/A Compatibility	Enable this option to generate PDF/A compliant PDFs with PDF configurations. This setting can be enabled alongside Fast Web Compatibility. **Notes** • To enable PDF/A output, you must enable the **Allow modify content** option on the **Protection** tab. • This setting is not available if password protection is enabled with PDF configurations.
Skip PDF Configuration For Signed PDF (Classic only)	Enable this option to skip PDF processing of digitally signed PDF documents. This prevents their digital signatures from being invalidated by altering the document after it was signed. When enabled, PDF processing of a signed PDF is skipped and the native PDF simply downloads without applying the PDF configuration.
Full compression	If you are using Acrobat 6.0 or later, select the option to compress the view configuration.
Label <*language*>	Type a label.
Input format	Select a rendition to use if no PDF rendition configuration is applied to the content.

4. To configure the list of properties of the content, select the **Properties** tab:

   a. Select a property from **Properties** or from the list of added properties.

   b. If you want to apply a dictionary to the property, select a dictionary from **Dictionaries** and optionally an alias or language.

   c. Click **Insert** to add the property.

5. To configure the security parameters of the content, select the **Protection** tab:

   a. Fill out the fields as described in the following table:

Field	Description
Encryption level	Select to use either 40 or 128 bit encryption levels.
User password	Set a password for opening the PDF.

Field	Description
Password to modify preference	Set a password for modifying the PDF.

b. Select **Allow printing**, **Allow modify content**, **Allow copy**, or **Allow modify annotations** to allow the user to perform the respective actions.

6. To configure the default viewing settings, select the **Initial View** tab:

a. Fill out the fields as described in the following table:

Field	Description
Page Layout	Select the page layout.
Display	Select the display mode.
Displayed page on running	Select the page number to show when the content is opened.

b. Configure the window by selecting or clearing **Resize from init page**, **Center on the screen**, and **Open with fullscreen mode**.

c. Configure the user interface by selecting or clearing **Hide Tool bar**, **Hide menu bar**, and **Hide Windows UI**.

7. To merge files into the current content:

a. To insert a table of contents, click **Insert table of content** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Import an XSL for formatting the table of contents XML generated by OpenText Documentum CM. Formatting C2 Table of Contents using XSL contains further information about the values to map.
Bookmark label	Type a label.
Adopt same size as first page	Select to ensure when the document is published that all pages will output as the same size as the first page.
Table of content's max depth	Select the number of levels used by the table of contents.
Separator type	Select the type of separator used between the title of a section and the page number.

b. To insert a blank page, click **Insert blank page** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
Landscape orientation	Select to flip the page to be horizontally oriented.
Only if it's a even page	Select if the page is a blank page that users only view when it falls on an even page.
Adopt same size as first page	Select to set added pages to the same size as the first page in the document. When the document is printed, all pages will print as expected on the same sized paper.

c. To insert a PDF file, click **Insert PDF** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**. **Note:** This PDF can be created with dynamic field targets so that it can be populated with data from OpenText Documentum CM. See “List of PDF Fields” on page 511 for more information on dynamic field usage.
Bookmark label	Type a label.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.
Adopt same size as first page	Select to set added pages to the same size as the first page in the document. When the document is printed, all pages will print as expected on the same sized paper.

d. To insert an XSL file, click **Insert XSL** and fill out the form as described in the following table:

Field	Description
Name	Type a name.

Field	Description
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Properties	Click to open the **PDF Configurations Xsl file** dialog box. You can add, remove, and reorder DQL queries for modifying the properties of the XSL file.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.
Adopt same size as first page	Select to set added pages to the same size as the first page in the document. When the document is printed, all pages will print as expected on the same sized paper.

e. Use the list controls to reorder merged files.

8. To add stamping to the content:

   a. To add an image watermark, click **Insert image watermark**. Select an image from the list box or click **New** to open a dialog box for adding an image. Type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

      Click **Properties** to view and edit the properties of the watermark.

   b. To add a text watermark, click **Insert text watermark**. Select a watermark from the list box or click **New** to open the **PDF Configuration watermark properties** dialog box. Fill out the form as described in the following table:

      Note: PDF configurations support text watermark TrueType font (.ttf) only. OpenType font (.otf) is not supported.

Field	Description
Name	Type a name.

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>Text</td><td>Type the text for the watermark. You can use the following variables:
<code>'$value()'</code> to use properties and optionally dictionaries.
For example, use <code>'$value(a_status)'</code> to show the status of content. Use <code>$value(r_modified_date.dd/MM/yyyy)</code> to show the modified date of content in dd/MM/yyyy format
<code>$dqlvalue()</code> to use a DQL query.
For example, to show the status of the specified content, use:
<code>$dqlvalue("select a_status from dm_document where r_object_id='$value(r_object_id)'")</code>
Pressing Enter within the text box will add a blank vertical space.</td></tr>
<tr><td>Font name</td><td>Select a font from the list box.</td></tr>
<tr><td>Font size</td><td>Select or type a font size.</td></tr>
<tr><td>Font color</td><td>Select a font color.</td></tr>
<tr><td>Horizontal align</td><td>Select the horizontal alignment.</td></tr>
<tr><td>Vertical align</td><td>Select the vertical alignment.</td></tr>
<tr><td>Opacity</td><td>Select the opacity.</td></tr>
<tr><td>Rotation</td><td>Select the rotation.</td></tr>
</table>

You can click **Properties** to view and edit the properties of the watermark.

c. To add a PDF layer, click **Insert PDF layer**. Select a file from the list box or click **Browse** to open a dialog box for adding a file. Type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

   You can use a PDF document to add dynamic image watermarking. The PDF document you add as a PDF layer must contain a text field that uses the `$image()` variable to refer to an image.

   For example, if you store images named after content status in the `/System/Images/` folder, type the following line in the text field:

   ```
   $image(/System/Images/$value(a_status).jpg)
   ```

d. You can click **Preview** to preview the look of the watermark.

   Watermarks that use a PDF layer and a variable do not show up in the preview.

e. Fill out the form as described in the following table:

Field	Description
Applicated on	Select the scope of pages you want to apply the watermarks.
From	Select the first bookmark.
to page	Select the last bookmark.
Apply on top of content	Select to place the watermark over the text. By default, watermarks are placed below the text.

9. To configure the document content:

   a. To insert a blank page, click **Insert blank page** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
Landscape orientation	Select to flip the page to be horizontally oriented.
Only if it's a even page	Select if the page is a blank page that users only view when it falls on an even page.

   b. To insert a PDF file, click **Insert PDF** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

   c. To insert an XSL file, click **Insert XSL** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

Field	Description
Bookmark label	Type a label.
Properties	Click to open the **PDF Configuration Xsl file** dialog box. You can add, remove, and reorder DQL queries for modifying the properties of the XSL file.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

d. To delete a page, click **Delete a document content page**, then type the page number of the page to delete.

e. Use the list controls to reorder document content.

10. Click **Save**.

11. Navigate to **Tools > Refresh Cache** to ensure that your changes take effect on the client.

Related Topics

“Overview of PDF Configuration” on page 351

“Formatting Table of Contents using XSL in PDF configurations” on page 380

“Configuring Print Configuration” on page 360

“Configuring Export Configuration” on page 366

“Configuring a Static Configuration” on page 373

“Configuring Administrator Access to PDF Configuration” on page 381

16.3 Configuring Print Configuration

1. Navigate to **PDF Configuration** > **Print configuration** from the menu bar.

2. Click **New** to create a print configuration.

   If you want to create a child print configuration that inherits the properties of an existing print configuration, select a print configuration and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Select a print configuration.
Description	Type a description.

Field	Description
Default configuration	Select to make this configuration the default print configuration. If you enable more than one default configuration in the configuration matrix, OpenText Documentum CM prioritizes top-to-bottom among the configurations labeled as default.
Compatibility	Select the oldest compatible version of the PDF viewing software.
Fast Web Compatibility	Select to generate linearized PDFs. **Note:** The generation of fast web-enabled PDFs should only be enabled when strictly required. Producing linearized PDFs requires the documents to be sent to the CTS system for processing, increasing output time.
PDF/A Compatibility	Enable this option to generate PDF/A compliant PDFs with PDF configurations. This setting can be enabled alongside Fast Web Compatibility. **Notes** • To enable PDF/A output, you must enable the **Allow modify content** option on the **Protection** tab. • This setting is not available if password protection is enabled with PDF configurations.
Skip PDF Configuration For Signed PDF (Classic)	Enable this option to skip PDF processing of digitally signed PDF documents. This prevents their digital signatures from being invalidated by altering the document after it was signed. When enabled, PDF processing of a signed PDF is skipped and the native PDF simply downloads without applying the PDF configuration.
Label *<language>*	Type a label.
Input format (will be used only if there is no PDF rendition configuration applied on document)	Select a file format.

4. Configure the form for end users to fill out as described in the following table:

Field	Description
Field number <*number*>	Select to enable and configure the field in the controlled printing dialog box.
Label <*language*>	Type a label.
Assistance type	Select the input mode for entering the reason for printing. • Free text: the user types the reason. • DQL: type a query to provide the user with a dynamic list of values. • Dictionary: select a dictionary for a static list of values.

End users are required to fill out these fields when they print through a PDF configuration. The information typed into the field is stored and shown with the audit for the controlled print.

The **Field n°1** field corresponds to the list of recipients. End users can type a list of users in this field during controlled print. The **Recipient** field in the recall print dialog functions as a list box of users entered.

5. To merge files into the current content:

   a. To insert a table of contents, click **Insert table of content** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Import an XSL for formatting the table of contents XML generated by OpenText Documentum CM. Formatting C2 Table of Contents using XSL contains further information about the values to map.
Bookmark label	Type a label.
Table of content's max depth	Select the number of levels used by the table of contents.
Separator type	Select the type of separator used between the title of a section and the page number.

   b. To insert a blank page, click **Insert blank page** and fill out the form as described in the following table:

Field	Description
Name	Type a name.

Field	Description
Landscape orientation	Select to flip the page to be horizontally oriented.
Only if it's a even page	Select if the page is a blank page that users only view when it falls on an even page.

c. To insert a PDF file, click **Insert PDF** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

d. To insert an XSL file, click **Insert XSL** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Properties	Click to open the **PDF Configuration Xsl file** dialog box. You can add, remove, and reorder DQL queries for modifying the properties of the XSL file.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

e. Use the list controls to reorder merged files.

6. To add stamping to the content:

   a. To add an image watermark, click **Insert image watermark**. Select an image from the list box or click **New** to open a dialog box for adding an image. Type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

   You can click **Properties** to view and edit the properties of the watermark.

b. To add a text watermark, click **Insert text watermark**. Select a watermark from the list box or click **New** to open the **PDF Configuration watermark properties** dialog box. Fill out the form as described in the following table:

Field	Description
Name	Type a name.
Text	Type the text for the watermark. You can use the following variables: • `'$value()'` to use properties and optionally dictionaries. For example, use `'$value(a_status)'` to show the status of content. Use `$value(r_modified_date.dd/MM/yyyy)` to show the modified date of content in dd/MM/yyyy format. • `$dqlvalue()` to use a DQL query. For example, to show the status of the specified content, use: `$dqlvalue("select a_status from dm_document where r_object_id='$value(r_object_id)'")`
Font name	Select a font from the list box.
Font size	Select or type a font size.
Font color	Select a font color.
Horizontal align	Select the horizontal alignment.
Vertical align	Select the vertical alignment.
Opacity	Select the opacity.
Rotation	Select the rotation.

Click **Properties** to view and edit the properties of the watermark.

c. To add a PDF layer, click **Insert PDF layer**. Select a file from the list box or click **Browse** to open a dialog box for adding a file. Type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

You can use a PDF document to add dynamic image watermarking. The PDF document you add as a PDF layer must contain a text field that uses the `$image()` variable to refer to an image.

For example, if you store images named after content status in the `/System/Images/` folder, type the following line in the text field:

```
$image(/System/Images/$value(a_status).jpg)
```

d. You can click **Preview** to preview the look of the watermark.

Watermarks that use a PDF layer and a variable do not show up in the preview.

e. Fill out the form as described in the following table:

Field	Description
Applicated on	Select the scope of pages you want to apply the watermarks.
From	Select the first bookmark.
to page	Select the last bookmark.
Apply on top of content	Select to place the watermark over the text. By default, watermarks are placed below the text.

7. To configure the document content:

a. To insert a blank page, click **Insert blank page** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
Landscape orientation	Select to flip the page to be horizontally oriented.
Only if it's a even page	Select if the page is a blank page that users only view when it falls on an even page.

b. To insert a PDF file, click **Insert PDF** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

c. To insert an XSL file, click **Insert XSL** and fill out the form as described in the following table:

Field	Description
Name	Type a name.

Field	Description
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Properties	Click to open the **PDF Configuration Xsl file** dialog box. You can add, remove, and reorder DQL queries for modifying the properties of the XSL file.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

d. To delete a page, click **Delete a document content page**, then type the page number of the page to delete.

e. Use the list controls to reorder document content.

8. Click **Save**.

9. Navigate to **Tools > Refresh Cache** to ensure that your changes take effect on the client.

Related Topics

“Overview of PDF Configuration” on page 351

“Formatting Table of Contents using XSL in PDF configurations” on page 380

“Configuring View Configuration” on page 353

“Configuring Export Configuration” on page 366

“Configuring a Static Configuration” on page 373

“Configuring Administrator Access to PDF Configuration” on page 381

16.4 Configuring Export Configuration

Note: PDF configurations apply to documents when they are downloaded either individually or in a bulk `.zip` file. To configure the zip and download option, see “Configuring Smart View Import/Export” on page 340.

1. Navigate to **PDF Configuration** > **Export configuration** from the menu bar.

2. Click **New** to create an export configuration.

   If you want to create a child export configuration that inherits the properties of an existing export configuration, select an export configuration and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Select a export configuration.
Description	Type a description.
Applications	Add or remove the applications to which this export configuration applies. For example, adding the `QA` application would cause the export configuration to only apply to matching quality assurance cases.
Default configuration	Select to make this configuration the default export configuration. If you enable more than one default configuration in the configuration matrix, OpenText Documentum CM prioritizes top-to-bottom among the configurations labeled as default.
Compatibility	Select the oldest compatible version of the PDF viewing software.
Fast Web Compatibility	Select to generate linearized PDFs. **Note:** The generation of fast web-enabled PDFs should only be enabled when strictly required. Producing linearized PDFs requires the documents to be sent to the CTS system for processing, increasing output time.
PDF/A Compatibility	Enable this option to generate PDF/A compliant PDFs with PDF configurations. This setting can be enabled alongside Fast Web Compatibility. **Notes** • To enable PDF/A output, you must enable the **Allow modify content** option on the **Protection** tab. • This setting is not available if password protection is enabled with PDF configurations.

Field	Description
Skip PDF Configuration for Signed PDF (Classic client only)	Enable this option to skip PDF processing of digitally signed PDF documents. This prevents their digital signatures from being invalidated by altering the document after it was signed. When enabled, PDF processing of a signed PDF is skipped and the native PDF simply downloads without applying the PDF configuration.
Full compression	This option appears if you are using Acrobat 6.0 or later. Select to compress the export configuration.
Label <*language*>	Type a label that appears in the interface.
Input format	Select the file format used when exporting content.

4. To configure the list of properties of the content, select the **Properties** tab:

   a. Select a property from **Properties** or from the list of added properties.

   b. If you want to apply a dictionary to the property, select a dictionary from **Dictionaries** and optionally an alias or language.

   c. Click **Insert** to add the property.

5. To configure the security parameters of the content, select the **Protection** tab:

   a. Fill out the fields as described in the following table:

Field	Description
Encryption level	Select to use either 40 or 128 bit encryption levels.
User password	Set a password for opening the PDF.
Password to modify preference	Set a password for modifying the PDF.

   b. Select **Allow printing**, **Allow modify content**, **Allow copy**, or **Allow modify annotations** to allow the user to perform the respective actions.

6. To configure the default viewing settings, select the **Initial View** tab:

   a. Fill out the fields as described in the following table:

Field	Description
Page Layout	Select the page layout.
Display	Select the display mode.
Displayed page on running	Select the page number that appears when the content is opened.

b. Configure the window by selecting or clearing **Resize from init page**, **Center on the screen**, and **Open with fullscreen mode**.

c. Configure the user interface by selecting or clearing **Hide Tool bar**, **Hide menu bar**, and **Hide Windows UI**.

7. To merge files into the current content:

   a. To insert a table of contents, click **Insert table of content** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Import an XSL for formatting the table of contents XML generated by OpenText Documentum CM. Formatting C2 Table of Contents using XSL contains further information about the values to map.
Bookmark label	Type a label.
Table of content's max depth	Select the number of levels used by the table of contents.
Separator type	Select the type of separator used between the title of a section and the page number.

   b. To insert a blank page, click **Insert blank page** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
Landscape orientation	Select to flip the page to be horizontally oriented.
Only if it's a even page	Select if the page is a blank page that users only view when it falls on an even page.

   c. To insert a PDF file, click **Insert PDF** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

Field	Description
Bookmark label	Type a label.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

d. To insert an XSL file, click **Insert XSL** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Properties	Click to open the **PDF Configuration Xsl file** dialog box. You can add, remove, and reorder DQL queries for modifying the properties of the XSL file.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

e. Use the list controls to reorder merged files.

8. To add stamping to the content:

   a. To add an image watermark, click **Insert image watermark**. Select an image from the list box or click **New** to open a dialog box for adding an image. Type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

      Click **Properties** to view and edit the properties of the watermark.

   b. To add a text watermark, click **Insert text watermark**. Select a watermark from the list box or click **New** to open the **PDF Configuration watermark properties** dialog box. Fill out the form as described in the following table:

Field	Description
Name	Type a name.

Field	Description
Text	Type the text for the watermark. You can use the following variables: • `'$value()'` to use properties and optionally dictionaries. For example, use `'$value(a_status)'` to show the status of content. Use `$value(r_modified_date.dd/MM/yyyy)` to show the modified date of content in dd/MM/yyyy format. • `$dqlvalue()` to use a DQL query. For example, to show the status of the specified content, use: `$dqlvalue("select a_status from dm_document where r_object_id='$value(r_object_id)'")`
Font name	Select a font from the list box.
Font size	Select or type a font size.
Font color	Select a font color.
Horizontal align	Select the horizontal alignment.
Vertical align	Select the vertical alignment.
Opacity	Select the opacity.
Rotation	Select the rotation.

You can click **Properties** to view and edit the properties of the watermark.

c. To add a PDF layer, click **Insert PDF layer**. Select a file from the list box or click **Browse** to open a dialog box for adding a file. Type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

   You can use a PDF document to add dynamic image watermarking. The PDF document you add as a PDF layer must contain a text field that uses the `$image()` variable to refer to an image.

   For example, if you store images named after content status in the `/System/Images/` folder, type the following line in the text field:

   ```
   $image(/System/Images/$value(a_status).jpg)
   ```

d. You can click **Preview** to preview the look of the watermark.

   Watermarks that use a PDF layer and a variable do not show up in the preview.

e. Fill out the form as described in the following table:

Field	Description
Applicated on	Select the scope of pages you want to apply the watermarks.
From	Select the first bookmark.
to page	Select the last bookmark.
Apply on top of content	Select to place the watermark over the text. By default, watermarks are placed below the text.

9. To configure the document content:

   a. To insert a blank page, click **Insert blank page** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
Landscape orientation	Select to flip the page to be horizontally oriented.
Only if it's a even page	Select if the page is a blank page that users only view when it falls on an even page.

   b. To insert a PDF file, click **Insert PDF** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

   c. To insert an XSL file, click **Insert XSL** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

Field	Description
Bookmark label	Type a label.
Properties	Click to open the **PDF Configuration Xsl file** dialog box. You can add, remove, and reorder DQL queries for modifying the properties of the XSL file.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

d. To delete a page, click **Delete a document content page**, then type the page number of the page to delete.

e. Use the list controls to reorder document content.

10. Click **Save**.

11. Navigate to **Tools > Refresh Cache** to ensure that your changes take effect on the client.

Related Topics

"Overview of PDF Configuration" on page 351

"Formatting Table of Contents using XSL in PDF configurations" on page 380

"Configuring View Configuration" on page 353

"Configuring Print Configuration" on page 360

"Configuring a Static Configuration" on page 373

"Configuring Administrator Access to PDF Configuration" on page 381

16.5 Configuring a Static Configuration

1. Navigate to **PDF Configuration** > **Rendition configuration** from the menu bar.

2. Click **New** to create a static configuration.

   If you want to create a child static configuration that inherits the properties of an existing static configuration, select a static configuration and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Select a print configuration.
Description	Type a description.

Field	Description
Compatibility	Select the oldest compatible version of the PDF viewing software.
Fast Web Compatibility	Select to generate linearized PDFs. **Note:** The generation of fast web-enabled PDFs should only be enabled when strictly required. Producing linearized PDFs requires the documents to be sent to the CTS system for processing, increasing output time.
PDF/A Compatibility	Enable this option to generate PDF/A compliant PDFs with PDF configurations. This setting can be enabled alongside Fast Web Compatibility. **Notes** • To enable PDF/A output, you must enable the **Allow modify content** option on the **Protection** tab. • This setting is not available if password protection is enabled with PDF configurations.
Skip PDF Configuration For Signed PDF (Classic client only)	Enable this option to skip PDF processing of digitally signed PDF documents. This prevents their digital signatures from being invalidated by altering the document after it was signed. When enabled, the native PDF is added as a PDF rendition to the source document without applying the PDF configuration.

4. Select an **Input rendition format** and an **Output rendition format** from the list boxes. For example, you can select `Acrobat PDF` and `PDF Rendition`, respectively, to create a configuration that converts Acrobat PDF to a specific PDF configuration.

5. To configure the list of properties of the content, select the **Properties** tab:

   a. Select a property from **Properties** or from the list of added properties.

   b. If you want to apply a dictionary to the property, select a dictionary from **Dictionaries** and optionally an alias or language.

   c. Click **Insert** to add the property.

6. To configure the security parameters of the content, select the **Protection** tab:

a. Fill out the fields as described in the following table:

Field	Description
Encryption level	Select to use either 40 or 128 bit encryption levels.
User password	Set a password for opening the PDF.
Password to modify preference	Set a password for modifying the PDF.

b. Select **Allow printing**, **Allow modify content**, **Allow copy**, or **Allow modify annotations** to allow the user to perform the respective actions.

7. To configure the default viewing settings, select the **Initial View** tab:

a. Fill out the fields as described in the following table:

Field	Description
Page Layout	Select the page layout.
Display	Select the display mode.
Displayed page on running	Select the page number that appears when the content is opened.

b. Configure the window by selecting or clearing **Resize from init page**, **Center on the screen**, and **Open with fullscreen mode**.

c. Configure the user interface by selecting or clearing **Hide Tool bar**, **Hide menu bar**, and **Hide Windows UI**.

8. To merge files into the current content:

a. To insert a table of contents, click **Insert table of content** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Import an XSL for formatting the table of contents XML generated by OpenText Documentum CM. Formatting C2 Table of Contents using XSL contains further information about the values to map.
Bookmark label	Type a label.
Table of content's max depth	Select the number of levels used by the table of contents.
Separator type	Select the type of separator used between the title of a section and the page number.

b. To insert a blank page, click **Insert blank page** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
Landscape orientation	Select to flip the page to be horizontally oriented.
Only if it's a even page	Select if the page is a blank page that users only view when it falls on an even page.

c. To insert a PDF file, click **Insert PDF** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

d. To insert an XSL file, click **Insert XSL** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Properties	Click to open the **PDF Configuration Xsl file** dialog box. You can add, remove, and reorder DQL queries for modifying the properties of the XSL file.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

e. Use the list controls to reorder merged files.

9. To add stamping to the content:

a. To add an image watermark, click **Insert image watermark**. Select an image from the list box or click **New** to open a dialog box for adding an image.

Type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

You can click **Properties** to view and edit the properties of the watermark.

b. To add a text watermark, click **Insert text watermark**. Select a watermark from the list box or click **New** to open the **PDF Configuration watermark properties** dialog box. Fill out the form as described in the following table:

Field	Description
Name	Type a name.
Text	Type the text for the watermark. You can use the following variables: • `'$value()'` to use properties and optionally dictionaries. For example, use `'$value(a_status)'` to show the status of content. Use `$value(r_modified_date.dd/MM/yyyy)` to show the modified date of content in dd/MM/yyyy format. • `$dqlvalue()` to use a DQL query. For example, to show the status of the specified content, use: `$dqlvalue("select a_status from dm_document where r_object_id='$value(r_object_id)'")`
Font name	Select a font from the list box.
Font size	Select or type a font size.
Font color	Select a font color.
Horizontal align	Select the horizontal alignment.
Vertical align	Select the vertical alignment.
Opacity	Select the opacity.
Rotation	Select the rotation.

Click **Properties** to view and edit the properties of the watermark.

c. To add a PDF layer, click **Insert PDF layer**. Select a file from the list box or click **Browse** to open a dialog box for adding a file. Type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.

You can use a PDF document to add dynamic image watermarking. The PDF document you add as a PDF layer must contain a text field that uses the `$image()` variable to refer to an image.

For example, if you store images named after content status in the `/System/Images/` folder, type the following line in the text field:

```
$image(/System/Images/$value(a_status).jpg)
```

d. You can click **Preview** to preview the look of the watermark.

   Watermarks that use a PDF layer and a variable do not show up in the preview.

e. Fill out the form as described in the following table:

Field	Description
Applicated on	Select the scope of pages you want to apply the watermarks.
From	Select the first bookmark.
to page	Select the last bookmark.
Apply on top of content	Select to place the watermark over the text. By default, watermarks are placed below the text.

10. To configure the document content:

a. To insert a blank page, click **Insert blank page** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
Landscape orientation	Select to flip the page to be horizontally oriented.
Only if it's a even page	Select if the page is a blank page that users only view when it falls on an even page.

b. To insert a PDF file, click **Insert PDF** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

c. To insert an XSL file, click **Insert XSL** and fill out the form as described in the following table:

Field	Description
Name	Type a name.
File	Select a file from the list box or click **Browse**, type a name for the file, click **Browse** to select a file, then select a **Format** from the list box, and click **OK**.
Bookmark label	Type a label.
Properties	Click to open the **PDF Configuration Xsl file** dialog box. You can add, remove, and reorder DQL queries for modifying the properties of the XSL file.
Include bookmarks contained in the pdf	Select to add the contained bookmarks to the list.

d. To delete a page, click **Delete a document content page**, then type the page number of the page to delete.

e. Use the list controls to reorder document content.

11. Click **Save**.

12. Navigate to **Tools > Refresh Cache** to ensure that your changes take effect on the client.

Related Topics

“Overview of PDF Configuration” on page 351

“Formatting Table of Contents using XSL in PDF configurations” on page 380

“Configuring View Configuration” on page 353

“Configuring Print Configuration” on page 360

“Configuring Export Configuration” on page 366

“Configuring Administrator Access to PDF Configuration” on page 381

16.6 Formatting Table of Contents using XSL in PDF configurations

OpenText Documentum CM automatically generates an XML containing the elements of the table of contents. Create and import an XSL for formatting the table of contents when viewing, printing, or exporting content with a PDF configuration.

The XML uses parameters within an `object` in the `root` as described in the following table:

Parameter	Description
`depth`	The depth is a numeric value indicating the number of indentations preceding the element in the table of contents. For example, `<depth>0</depth>` corresponds to 0 indentations, while `<depth>1</depth>` corresponds to a single indentation.
`action`	The action indicates the type of link being used by the element in the table of contents. Use `GOTO` to have the table of contents send the reader to the element when the link is clicked.
`title`	The title is the label shown for the element in the table of contents.
`page`	The page is the page number.

Related Topics

“Overview of PDF Configuration” on page 351

“Clearing the dmi_queue_item After Creating a PDF Rendition Configuration” on page 381

“Overview of PDF Configuration” on page 351

“Configuring View Configuration” on page 353

“Configuring Print Configuration” on page 360

“Configuring Export Configuration” on page 366

“Configuring a Static Configuration” on page 373

“Configuring Administrator Access to PDF Configuration” on page 381

16.7 Configuring Administrator Access to PDF Configuration

You can configure the group of users given access to changing PDF configurations using client configuration.

1. Navigate to **Tools** > **Options** from the menu bar.
2. Click the **PDF Configuration Options** tab.
3. Select or type a group to set as the PDF Configuration Administrator group.
4. Click **Save**.

Related Topics

"Overview of PDF Configuration" on page 351

"Formatting Table of Contents using XSL in PDF configurations" on page 380

"Configuring View Configuration" on page 353

"Configuring Print Configuration" on page 360

"Configuring Export Configuration" on page 366

"Configuring a Static Configuration" on page 373

16.8 Clearing the dmi_queue_item After Creating a PDF Rendition Configuration

The PDF Configuration plug-in creates a `c2_rendition` event in the `dmi_queue_item` queue to generate the PDF rendition. After completing the request and generating the PDF, the system marks the item as dequeued but does not clear the `dmi_queue_item` events.

To remove the queue items you must either:

- Use the queue management tool (dm_QueueMgt job) to remove the queue item.

  The queue management tool is installed out-of-the-box in an inactive state. The *Documentum CM Server Administration and Configuration Guide* contains more information about the queue management tool.

- Delete the queue items using the `DELETE...OBJECT` DQL query. For example, to remove all queue objects representing objects that were dequeued before January 1, 2013:

```
DELETE "dmi_queue_item" OBJECTS WHERE "dequeued_date" < DATE('01/01/2013') AND
"delete_flag" = true
```

Related Topics

"Overview of PDF Configuration" on page 351

"Formatting Table of Contents using XSL in PDF configurations" on page 380

16.9 Preventing PDF Configuration processing based on document page count

You can prevent PDF Configuration processing of a document based on the number of pages it contains by modifying the *c2ProcessingMaxPageThreshold* property in the D2FS properties file.

By default *c2ProcessingMaxPageThreshold* is set to `-1`, which permits processing a document of any size. It appears in the D2FS properties file as follows:

```
c2ProcessingMaxPageThreshold=-1
```

To create an maximum page number for PDF Configuration processing, change the value to the highest page count you want to allow. For example, if you want to limit processing to documents with 50 or fewer pages, set *c2ProcessingMaxPageThreshold* to `50`. It would appear in the D2FS properties file as follows:

```
c2ProcessingMaxPageThreshold=50
```

If a user attempts to process a document with more pages than the maximum limit, an warning message appears.

After modifying the *c2ProcessingMaxPageThreshold* value, you must restart the server for the configuration to take effect.

16.10 Skipping PDF processing for signed PDFs (Classic client only)

Using PDF processing on signed PDF documents can invalidate signatures because it alters the document after the signature was added. You can choose to skip PDF processing for signed PDFs by selecting the **Skip PDF Configuration For Signed PDF** option in PDF configurations. If you enable this option, the following occurs:

- PDF processing of a signed PDF is skipped and the native PDF simply downloads without applying the PDF configuration.
- The native PDF is added as a configured PDF rendition to the source document without applying the PDF configuration.

Chapter 17

Configuring Transfer Configurations

17.1 Configuring Transfer Configurations

Transfer Configuration allows you to manage transferring between OpenText Documentum CM and Microsoft Office documents as well as between .eml and .msg email messages. The following table describes file format and software version compatibility:

	Office 2007-2016 and Office 365 (Desktop)	Open Office 3.2
.doc	Yes, with macro support	Yes
.docx	Yes	Yes
.docm	Yes, with macro support	Yes
.xls	Yes	No
.xlsx	Yes, but cell values must not be empty	No
.ppt	Yes	No
.pptx	Yes	Yes

1. Navigate to **Transfer Configurations** > **Transfer configuration** from the menu bar.

2. Click **New** to create a transfer configuration.

   If you want to create a child transfer configuration that inherits the properties of an existing transfer configuration, select a transfer configuration and click **Create from**. Understanding Parent and Child Configurations contains more information on child configurations.

3. Fill out the form as described in the following table:

Field	Description
Name	Type a name to appear in the configuration matrix.
Description	Type a description.
Applications	Add or remove the applications to which this transfer configuration applies. For example, adding the `QA` application would cause the transfer configuration to only apply to matching quality assurance cases.

Field	Description
Format of "date" property type	Type a date format using `dd` for date, `MM` for month, and `yy` for the last two digits of the year.
Trigger events	Use the list controls to add and remove the events that trigger a properties transfer.
Do not generate rendition request	Select to disable automatic rendition generation in the configuration. **Note:** Existing renditions for a document included in this configuration are deleted when you apply this setting.

4. Configure properties to transfer:

   Note: Transfer Configurations support custom properties only, and does not recognize default, out-of-the-box Microsoft Office properties.

   a. Use the list controls to add or remove properties.

      Content imported in a client with an associated configuration inherits the selected properties.

   b. Fill out the form as described in the following table:

Field	Description
Client property	Type the repository name of the property.
Office property	Type the Microsoft Office name of the property.
Relation direction	Select the direction of the property transfer. Do not use a bidirectional relationship with repeating properties.
Dictionary	Select a dictionary to use a dictionary.
Alias/Locale	Select an alias or locale to specify the dictionary being used.

5. Configure DQL queries to use in the Microsoft Office document:

   a. Use the list controls to add or remove DQL queries.

   b. Fill out the form as described in the following table:

Field	Description
DQL query identifier	Type a name.

Field	Description
DQL query	Type a DQL query.

Note: For example, to fetch `user_address` using a DQL query and assign it to a doc attribute, the doc property should be named `test.user_address` and the Transfer Configurations DQL query identifier `test`. If `user_address` is the exact column name of the `user_address` table, the same name should be provided in the doc template.

If you are importing email (`.eml` or `.msg`), `mail_body` is not recommended as part of the Transfer ConfigurationsTransfer Configuration because the mail body could be very large and the database table or column might not be able to hold all of the data. The mail body is automatically truncated to equal the size of the `mail_body` attribute. If it is not possible to fetch the size of the `mail_body` attribute, the default value of the mail body size is set at 2000.

6. Click **Save**.
7. Navigate to **Tools > Refresh Cache** to ensure that your changes take effect on the client.

Related Topics

"Configuring File Formats Recognized by transfer configurations" on page 385

"Configuring Microsoft Office with transfer configurations" on page 386

17.2 Configuring File Formats Recognized by transfer configurations

1. Navigate to **Tools** > **Options** from the menu bar.
2. Click the **Transfer Configurations Options** tab.
3. Use the list controls to add and remove file formats recognized by Transfer Configurations.
4. Click **Save**.

Related Topics

"Configuring Transfer Configurations" on page 383

"Configuring Microsoft Office with transfer configurations" on page 386

17.3 Configuring Microsoft Office with transfer configurations

Perform the following steps on all end user machines. Refer to the documentation for the Microsoft Office products for detailed instructions on creating and enabling a macro, setting trusted locations, and adding properties.

Note: A limitation exists in Microsoft Word with the TEXT property in that it cannot exceed 255 characters. This property is defined in Microsoft Word as a char data type with the limitation of 255 characters. This limitation does not exist in OpenText Documentum CM, however when you map a property with a transfer configuration, whose contents exceeds 255 characters, to a property in Microsoft Word, you will experience a data loss due to the Microsoft Word limitation.

Due to a Microsoft Word limitation, when a OpenText Documentum CM property (which is transferred by the transfer configuration plugin to the Microsoft Word Custom property) contains carriage returns (for example, new line or \n character) the Word document is not able to display the property correctly in its contents. As a result, do not to add carriage returns in OpenText Documentum CM properties that are mapped to Word's Custom property.

1. Create a macro and enter the following sets of code:

   Microsoft Word only:

```
Sub AutoOpen()
Dim rngStory As Word.Range
Dim lngValidate As Long
Dim oShp As Shape
lngValidate = ActiveDocument.Sections(1).Headers(1).Range.StoryType
For Each rngStory In ActiveDocument.StoryRanges
'Iterate through all linked stories
Do
On Error Resume Next
rngStory.Fields.Update
Select Case rngStory.StoryType
Case 6, 7, 8, 9, 10, 11
If rngStory.ShapeRange.Count > 0 Then
For Each oShp In rngStory.ShapeRange
If oShp.TextFrame.HasText Then
oShp.TextFrame.TextRange.Fields.Update
End If
Next
End If
Case Else 'Do Nothing
End Select
On Error GoTo 0
'Get next linked story (if any)
Set rngStory = rngStory.NextStoryRange
Loop Until rngStory Is Nothing
Next

ActiveDocument.Saved = True
End Sub
```

   Microsoft Excel only:

```
Private Sub Workbook_Open()

```

```
    MsgBox "This Macro will execute when the workbook is opened"

End Sub
```

2. In the Microsoft Office configurations:

   a. Enable the macro.

   b. Add the repository view and checkout folders used by end users to the **Trusted Locations** list.

3. Create the properties in Microsoft Word, Excel, and Powerpoint to allow transferring property information.

4. To update the values of inserted properties in Microsoft Word, right-click the properties and select **Update fields**.

   If you cannot update the fields, make sure **Update fields** is enabled in the Microsoft Word configurations.

Related Topics

"Configuring Transfer Configurations" on page 383

"Configuring File Formats Recognized by transfer configurations" on page 385

Chapter 18

Configuring Brava!

18.1 About Brava!

Brava! is the advanced viewer for OpenText Documentum CM that is included in your installation. A separately purchased license is required to configure and use Brava! for Classic and the Smart View. The Brava CSR viewer is a separate, light version of the viewer that is available for use in both Classic and Smart View without a license fee.

This section explains how to configure the full version of Brava!.

Brava! offers simple-to-use annotation and collaboration, managing markup files within the repository and user interface. Users can view, annotate, and redact native files including PDFs, Microsoft Office documents, image files, 2D CAD drawings, and even video clips. Printing and annotation rights are honored based on User Groups and Roles. Source documents are never modified, keeping repository files intact. With either the DocMerge feature, or on-demand publishing through Brava!, documents can be easily saved directly into the repository, within any repository location, or checked into the Home Cabinet. With its zero-footprint HTML5 client, Brava! eliminates the need to install native applications on every desktop.

For more information about comparison of features and format support across the viewers, see “BravaCSR Viewer, Brava! Enterprise Viewer, and Intelligent Viewing feature reference“ on page 605

18.1.1 Smart View client implementation notes

The Smart View client implementation of the full Brava viewer is similar to the Classic implementation with the following differences:

- The ActiveX viewer is not supported. The HTML viewer is the primary viewer. All of the HTML viewer Brava parameters documented in this guide are applicable to the Smart View client. The following HTML viewer features are not supported:
  - Email link functionality
- A separate Brava Changemark widget is not available. Changemarks are integrated directly into the viewer.

18.2 Configuring Brava! Menus

Note: The following configuration information is applicable to Classic only. For more information about setting up Brava! menu items for the Smart View client, see “Brava! Compare menu items” on page 315.

To facilitate tighter integration of the Brava! Viewer workflows, a custom menu item configuration is available that enables integrators to overload the default **View** menu item to launch documents within the Brava! View widget. In addition, the Brava! Compare and Brava! DocMerge functionality is tightly integrated with a custom menu item. These menus initialize custom events to interact directly with other Brava! widgets in the user interface.

Additional custom menu items or events can be created to initiate events within Brava! widgets using the following custom event **Action** `D2_EVENT_CUSTOM`, combined with the **Message** parameters: `brava==view`, `brava==compare`, or `brava==docmerge`.

Alternately, you can set up these menu items automatically by importing `D2-Menu-Config-Brava.zip` from the`D2\igc\docbase` directory.

See “Configuring Client Menus” on page 293 for more general information about menu configurations.

18.3 About the BravaEnterpriseViewerWidget

Enables the three standard Brava! Enterprise widgets (Brava! Viewer widget, Brava! Compare widget, and Brava! Changemarks widget), which can be set up separately using the radio buttons in the widget configuration.

Note: Only the Brava! Viewer widget is supported in the Smart View client.

See “Configuring an Internal Widget” on page 276 and “List of Widgets” on page 457 for more information about BravaEnterpriseViewerWidget and general information on widget setup.

18.4 Configuring the Brava! Viewer Widget

The Brava! Viewer widget that more tightly integrates all viewing capabilities of the Brava! system. The contents of this widget are highly dependent upon configuration of `brava_parameters.properties` entries. The widget will host viewing sessions for users utilizing the Brava! HTML, Brava! ActiveX, Brava! Html Graphical Compare, Brava! Text Compare, and Brava! Html Video viewers, depending upon document selection and configuration of the Brava! system.

18.5 Configuring the Brava! Compare Widget

A Brava! Compare widget utilizes the bidirectional protocol (OAH) to manage viewing side by side documents in the Brava! Html Graphical Compare and Brava! ActiveX viewers. Which viewer is launched when documents are selected for comparison is determined by a combination of configuration parameters from the `brava_parameters.properties` file, as well as the user's browser. The widget receives events to load documents into a compare session which is then launched inside the Brava! Viewer widget. To utilize the Brava! Compare widget, the Brava! Viewer widget must also be loaded into the users workspace.

For more information about setting up Brava! compare menu items for the Smart View client, see “Brava! Compare menu items” on page 315.

18.6 Configuring the Brava! Changemarks Widget

Note: This widget is not supported in the Smart View client.

Brava! Changemarks have received considerable improvements and beginning with the 16.4 release, a new Brava! Changemark widget is available that allows users to more effectively utilize Changemark markup entities to collaborate over their content.

18.7 Using Intelligent URLs With Brava!

The Brava! integration Email Document Link feature utilizes Intelligent URLs to share a reference to content via the Brava! Viewer. Email Document Link URIs contain all the information and commands necessary to locate and select the requested document in the client application.

Note: Email links are not supported in the Smart View client.

The Email Link button launches a “mailto” hyperlink that uses the client's default mail client to prepare an email with a hyperlink to the document. The Email body contains the following text:

```
Please use the link below to view <filename> in the Brava! Viewer widget. If the
Brava! Viewer widget does not already exist in your D2 workspace you will need to add
it.
 http://<D2Server>?
locateId=<docid>&docbase=<repository>&commandEvent=D2_EVENT_CUSTOM&commandParam=brava==
view!!oam_id==<docid>
```

The values supplied for the fields *<filename>*, *<D2Server>*, *<docid>* and *<repository>* are provided by the Brava! integration and will differ based off the selected document, repository and system configuration.

Browsing to the URL provided in the email body will launch the client and prompt the user for their login credentials. Once authenticated, the system locates the document associated with the URL and issue an Event to display it in the Brava! Viewer widget.

Note: The user must have the Brava! Viewer widget added to their Workspace and must have permission to view the document for the Email Document Link functionality to work.

To enable the launch of Intelligent URLs to view documents in Brava! Viewer, Administrators must enable the D2_EVENT_CUSTOM action in client configuration.

1. As an administrator, log in to client configuration and access the **Client Options** dialog via **Tools > Options**.
2. Move the D2_EVENT_CUSTOM action from the **Forbidden actions** list to the **Allowed actions** list.

18.7.1 Configuring Brava! Email Document Links

This configuration step is required for Brava! Email Document Link functionality to work.

Note: Email links are not supported in the Smart View client.

The Brava! Viewer widget must be configured with the Focus Event value of D2_EVENT_LOCATED_OBJECT applied. This configuration setting causes the Brava! Viewer widget to be exposed within the workspace when it receives an Intelligent URL command to locate a document.

18.8 Adding custom Stamps (raster images)

You can add your own directory of custom raster images to be used with the integration. These images are available to the client machines when using the Insert Raster Image markup entity in Brava! markup mode.

For Brava! ActiveX Client:

1. Create or gather your desired raster images (PNG or JPG) and place them in a zip file.
2. Rename the file StampImages.bin.
3. Place the bin file in the <application server>\webapps\D2\igc directory.
4. The bin file will automatically be downloaded to the client machines and unzipped to the StampImage directory defined by the *MarkupRasterDirectory* parameter. By default, Windows 7 puts this folder in C:\<user_name>\IGC\<version>\StampImages.

For Brava! HTML Client:

1. Create or gather your desired raster images (PNG or JPG) and place them in the Brava! Server machine raster stamp directory as noted below:
2. The location of the raster stamps used by the HTML Client is located on the Brava! Server within the `raster` folder found within the Brava! Enterprise Server installation directory: `C:\Program Files\OpenText\Brava! Enterprise\Brava! Server\raster`.
3. With the HTML Client, users cannot browse (from the client machine or network drive) to select from raster stamps outside of the Brava! Server raster stamp location as they can with the ActiveX Client.

18.9 Adding Custom Symbols

Note: This functionality is not supported in the Smart View client.

To enable the Brava! HTML Client **Symbols** tool, Brava! must be configured to use a symbols webservice.

You can implement a webservice of your own that responds with the Symbol Template JSON Catalog Format that is defined in the BravaSDK. The BravaSDK symbol sample pages can be used to define your own rootUrl to service .xsp and .png/.jpg symbols and place the resulting URI to your symbol webservice as a value for the *`html.markupsFeature.SYMBOL_CONFIG_URI`* configuration parameter. If this parameter value is left blank (the default), the **Symbol** button is not available in Brava! HTML Client.

Note: The `BravaSDK` folder is installed when running the Brava! Enterprise installer with the **BravaSDK** option selected in the available components list. If this option was not selected during install, you can re-run the installer with just this option selected to obtain the necessary files.

To use the BravaSDK symbol sample pages:

1. Copy the `BravaSDK/HTML/samples/symbols` directory content into the Application Server hosting OpenText Documentum CM at `D2/igc/resources/symbols/`.
2. In the *`brava_parameters.properties`* file, update:`html.markupsFeature.SYMBOL_CONFIG_URI=/D2/igcresource?rpath=/igc/resources/symbols/sample-symbols.json`
3. In the `D2/igc/resources/symbols/sample-symbols.json` file, update the`rootUrl` value to `/D2/igcresource?rpath=/igc/resources/symbols`

18.10 Configuring the Save Markups With All Versions Option

This additional installation step must be performed to allow the **Save Markups with all Version** feature to recognize the proper security permissions and work properly.

- Run the following DQL line from the DQL editor:

  ```
  update dm_relation_type objects set security_type='PARENT' where
  relation_name='DM_ANNOTATE';
  ```

Notes

- You must be the docbase owner to run the above command.
- The parameter *`html.markupsFeature.EnableVersion`* must be set to TRUE in the `brava_parameters.properties` file to make the **Save with all Versions** option available in the HTML Viewers.
- The parameter *`EnableVersion`* must be set to TRUE in the `brava_parameters.properties` file to make the **Save with all Versions** option available in the ActiveX Viewer.

18.11 Configuring Onscreen Watermarks

The Brava! onscreen watermark feature provides the ability to configure banners, print watermark, and screen-watermark to pull in metadata values from the repository based on document types. This feature allows concatenation of free text along with a token (`%dbstring`) to obtain the metadata value from the docbase repository. This is done through configuring an xml page where users can specify multiple document types for onscreen watermark print and publish options.

Through configuring the `Brava_onscreen_watermark.xml` page, a user can specify multiple document types for onscreen watermark print and publish options when using the Brava! viewers (ActiveX, HTML, and Graphical Compare).

The `Brava_onscreen_watermark.xml` configuration file is located within the `igc` folder of the deployed application; for example: `\webapps\D2\igc`

The configurable elements and attributes contained in the `Brava_onscreen_watermark.xml` file are as follows.

Note: Attributes do not apply to the Screen Watermark `<screenwatermark>` or Screen Banner `<screenbanner>` elements.

`doctype`

Value: *<string>*

Description: Specifies doctypes where the banner and watermark will be applied.

`id`

Value: <*string*>

Description: Specifies the id for each doctype. The id must be unique and doctype name is an ideal id. It is a required field. Please be advised, there will be no cascading effect based on doctype. In other words, watermark and banners will not be applied to the subtypes of the specified doctype.

Example: `id=<doctype id>`

`name`

Value: <*string*>

Description: This required tag specifies the name for each doctype. The name must be unique.

Example: `name=<doctype name>`

`enabled`

Value: <*boolean*>

Description: This required tag specifies whether to enable the enforcement of on-screen watermark or banners based on the specified doctype.

Example: `enabled=<true/false>`

`watermark`

Value: <*string*>

Description: This optional value specifies the watermark to be rendered on print and publish. The user can specify clear text or pull metadata from the repository by using `%dbstring(attribute)`. Please be advised, the `%dbstring` is case sensitive and must be all lower caps.

Example: `watermark=Version Label %dbstring(r_version_label)`

`screenwatermark`

Value: <*string*>

Description: The `screenwatermark` setting provides the ability to set a on screen watermark that is independent of the watermark defined for the print settings. An on screen watermark will be visible to users when viewing the document.

Example: `screenwatermark=Internal use only`

`screenbanner`

Value: <*string*>

Description: The optional `screenbanner` setting provides the ability to set a banner on screen that is independent of the banners defined for the print or publishing settings. An on screen banner will be visible to users when viewing the document.

Example: `screenbanner=The date is %Date` `topleft`

Value: <*string*>

Description: This optional tag specifies the text that will display on the left corner of the top of the document.

Example: `topleft=Document Owner %dbstring(owner_name)`

`topcenter`

Value: <*string*>

Description: This optional tag specifies the text that will display on the center of the top of the document.

Example: `topcenter=Last modified %dbstring(r_modify_date)`

`topright`

Value: <*string*>

Description: This optional tag specifies the text that will display on the right corner of the top of document.

Example: `topright=Last modified %dbstring(r_modify_date)`

`bottomcenter`

Value: <*string*>

Description: This optional tag specifies the text that will display on the center of the bottom of the document.

Example: `bottomcenter=Object Name %dbstring(object_name)`

`bottomright`

Value: <*string*>

Description: This optional tag specifies the text that will display on the right corner of the bottom of the document.

Example: `bottomright=Last modified %dbstring(r_modify_date)`

`bottomleft`

Value: <*string*>

Description: This optional tag specifies the text that will display on the left corner of the bottom of the document.

Example: `bottomleft=Last modified %dbstring(r_modify_date)`

`righttop`

Value: <*string*>

Description: This optional tag specifies the text that will display on the top of the right side of the document.

Example: `righttop=Object Name %dbstring(object_name)` `rightcenter`

Value: *<string>*

Description: This optional tag specifies the text that will display on the center of the right side of the document.

Example: `rightcenter=Object Name %dbstring(object_name)`

`rightbottom`

Value: *<string>*

Description: This optional tag specifies the text that will display on the bottom of the right side of the document.

Example: `rightbottom=Object Name %dbstring(object_name)`

`lefttop`

Value: *<string>*

Description: This optional tag specifies the text that will display on the top of the left side of the document.

Example: `lefttop=Object Name %dbstring(object_name)`

`leftcenter`

Value: *<string>*

Description: This optional tag specifies the text that will display on the center of the left side of the document.

Example: `leftcenter=Object Name %dbstring(object_name)`

`leftbottom`

Value: *<string>*

Description: This optional tag specifies the text that will display on the bottom of the left side of the document.

Example: `leftbottom=Object Name %dbstring(object_name)`

Example 18-1: xml excerpt

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ISOBannerandWatermarkSettings>
    <doctype id="dm_document" name="dm_document" enabled="true">
        <watermark>Appproved</watermark>
        <screenwatermark>Appproved</screenwatermark>
        <screenbanner></screenbanner>
        <topleft></topleft>
        <topcenter>Object Name: %dbstring(object_name) ObjectId
%dbstring(r_object_id)</topcenter>
        <topright>Internal Use Only</topright>
        <bottomleft></bottomleft>
        <bottomcenter></bottomcenter>
        <bottomright></bottomright>
        <righttop></righttop>
        <rightcenter></rightcenter>
```

```
        <rightbottom></rightbottom>
        <lefttop></lefttop>
        <leftcenter></leftcenter>
        <leftbottom></leftbottom>
    </doctype>
    <doctype id="qa_document" name="qa_document" enabled="true">
        <watermark>%dbstring(a_status)</watermark>
        <screenwatermark>%dbstring(a_status)</screenwatermark>
        <screenbanner></screenbanner>
        <topleft></topleft>
        <topcenter>Object Name: %dbstring(object_name) ObjectId
%dbstring(r_object_id)</topcenter>
        <topright>Internal Use Only</topright>
        <bottomleft></bottomleft>
        <bottomcenter></bottomcenter>
        <bottomright></bottomright>
        <righttop></righttop>
        <rightcenter></rightcenter>
        <rightbottom></rightbottom>
        <lefttop></lefttop>
        <leftcenter></leftcenter>
        <leftbottom></leftbottom>
    </doctype>
</ISOBannerandWatermarkSettings>
```

18.11.1 Custom Attributes

Watermark and Banner elements can have one or more of the following attributes set. These attributes can be used in combination with the elements in the previous list (excluding `screenbanner` and `screenwatermark`):

Note: The attributes listed here are only used with the Brava! HTML viewer print and publish output and do not apply to Watermarks and Banners in Brava! ActiveX viewer. ActiveX viewer publish output is controlled by the ActiveX specific parameters: `BannerFontSize`, `WatermarkBannerFontName`, and `WatermarkBannerFontStyle`.

`font`

Value: *<string>*

Description: Specifies font face used.

Example: `font="Lucidia"`

`color`

Value: *<HEX color triplet>*

Description: Specifies the text color used.

Example: `color="00FF00"`

`size`

Value: *<integer>*

Description: Specifies the text size used.

Example: `size="12"`

`style`

Value: *<pipe delimited styles>*

Description: Specifies the text style used.

Example: `style="bold|italic|underline"`

`editable`

Value: *<boolean>*

Description: Specifies whether the watermark or banner entry is editable in the client.

Example: `editable="false"`

Example 18-2: xml excerpt with attributes

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<ISOBannerandWatermarkSettings>
    <doctype id="dm_document" name="dm_document" enabled="true">
        <watermark color="0000FF">PrintWatermark</watermark>
        <screenwatermark></screenwatermark>
        <screenbanner></screenbanner>
        <topleft font="Lucida">Top Left Lucida</topleft>
        <topcenter color="00FF00">Top Center Green</topcenter>
        <topright size="24">Tom Right Size 24</topright>
        <bottomleft style="bold">Bottom Left Bold</bottomleft>
        <bottomcenter style="bold|italic|underline">BC bold italic underline</
bottomcenter>
        <bottomright editable="true">Bottom Right Editable</bottomright>
        <righttop></righttop>
        <rightcenter></rightcenter>
        <rightbottom></rightbottom>
        <lefttop></lefttop>
        <leftcenter font="New York" color="00FF00" style="bold|underline"
size="24">LEFT CENTER - New York Green Bold Underline 24</leftcenter>
        <leftbottom></leftbottom>
        <leftbottom></leftbottom>
    </doctype>
</ISOBannerandWatermarkSettings>
```

18.12 Configuring the Watermark Folder Feature

The Watermark folder feature allows you to turn on watermarks based on the folder where the content resides.

In the *`brava_parameters.properties`* file:

- Set *`WatermarkFolders`* to TRUE
- Set a value for *`WatermarkFolderName`*

This allows you to add a folder name watermark to the files contained in the specified directory. These watermarks are applied during rendering and do not affect the original file.

Example 18-3:

If you have three folders named `Documents`, `Invoice`, and `Statement`, and only want to apply watermarks to documents that are added to `Invoice` and `Statement`:

`Set WatermarkFolders=True`

`Set WatermarkFolderName=Invoice,Statement`

Any file contained in `Invoice` or `Statement` folders now has watermarks applied (the folder name).

18.13 Configuring Brava! Logging

The OpenText Documentum CM integration uses Logback for logging events to be written to the app server. Separate log files are maintained by the Classic client and the Smart View client. By default, the appender related to logging is not present in `\WEB-INF\classes\logback.xml`. The following appender therefore needs to be added inside the <*config …*> tag in the `logback.xml` file for appending IGC specific events to `Igc_D2.log`. The location of the `Igc_D2.log` file is shown below. The directory entry can be configured as per business requirements.

Notes

- In the `brava_parameters.properties` file, the parameter *`ClientLogging`* must be set to the appropriate level (0-5) for logs to be written in the `Igc_D2.log` file.
- For the Smart View client, change the naming of the file to `Igc_D2Smartview.log` in all instances.

```
<appender class="ch.qos.logback.core.rolling.RollingFileAppender"
name="IgcD2Appenders">
        <file>C:\logs\Igc_D2.log</file>
        <append>true</append>
        <filter class="ch.qos.logback.classic.filter.ThresholdFilter">
            <level>debug</level>
        </filter>
        <rollingPolicy class="ch.qos.logback.core.rolling.TimeBasedRollingPolicy">
            <fileNamePattern>C:\logs\Igc_D2-%d{yyyy-MMdd}.log.zip</fileNamePattern>
            <MaxHistory>5</MaxHistory>
        </rollingPolicy>
        <layout class="ch.qos.logback.classic.PatternLayout">
            <pattern>%-27(%date{yyyy-MM-dd HH:mm:ss} [%-5level]) : %message%n</pattern>
        </layout>
    </appender>


    <logger additivity="false" name="com.igc">
        <level value="all"/>
        <appender-ref ref="IgcD2Appenders"/>
    </logger>
```

In addition, Brava! occasionally utilizes the application servers Standard Output or Standard Error interfaces to write exception information related to unexpected runtime errors. When troubleshooting errors in the Brava! integration, it might be necessary to enabled the output of these streams to disk. Different Application Servers support different mechanisms of logging; STDOUT and STDERR. Please refer to your application server documentation for instructions specific to your environments implementation.

18.13.1 Turning On Brava! Logging

Client, Server, and Job Processor logging can be set in following three locations: `brava_parameters.properties`, `Server.properties`, and `JobProcessor.config`.

To set up logging:

1. In `brava_parameters.properties` (Brava! Client control logging), set the parameters *`ClientLogging=0`* and *`ClientLogFileLength=100000`* (adding a few extra 0's ensures that everything gets written to the log file). The file is written to the folder specified in the `brava_parameters.properties` file, `ClientLogFile=parameter`.
2. In `Server.properties` (Brava! Server logging), set *`log.level=all`*. The file is written to the `C:\Program Files\OpenText\Brava! Enterprise\Brava! Server\log`.
3. To turn on the Job Processor logging, go to `<Windows Drive>:\Program Files\OpenText\Brava! Enterprise\JobProcessor\JobProcessor.config` and set the log parameter to **ON** in the `JobProcessor.config` file. Save the `JobProcessor.config` file and restart the Job Processor service. To view the logs, go to `http://localhost:7070/log`.

18.14 Determining the Default Compare Client

When two documents are selected to launch in compare mode, those documents are added to a Brava! Compare viewing session in the Brava! Viewer widget (configured for either the HTML or ActiveX viewer). The compare view mode that is launched is highly dependent upon the configuration set in the `brava_parameters.properties` file, *`DefaultViewer`* parameter.

Note: View the Online help for the Brava! ActiveX, Brava! HTML Compare, or Brava! Text Compare client for assistance with using the Brava! Compare mode.

An HTML Text Compare client can also be configured to launch, by default, for specified formats. This viewer can be configure through the `brava_parameters.properties` file via the *`html.textCompare.formats`* parameter.

For example, if you want all pdf, doc, docx, rtf, and txt files to always launch in Text Compare mode, set `html.textCompare.formats=pdf,doc,docx,rtf,txt`. Both of the selected document formats must be defined in this list to launch with the HTML

Text Compare viewer. Otherwise, the compare launches using the graphical compare viewer, as defined by the *`DefaultViewer`* parameter.

18.15 Bi-directional Stamp feature overview

When inserting dynamic stamp tokens into markup text strings, an integration can use the custom `%dbupdatestring` and `%dbstring` tokens in the stamp template to allow communication with the integration database and to dynamically update the stamp's field label values when the stamp is inserted into a document markup.

Stamp templates can be created using multiple `%dbstring` and `%dbupdatestring` tokens for fields that need to be populated from OpenText Documentum CM. Fields marked with `%dbupdatestring` are displayed in a **bi-directional** dialog when the end-user initially places the resulting stamp on a document.

For ActiveX Viewer only:

Upon dialog initialization, the field values are populated with the latest values from the docbase. This dialog allows users to update various fields in the docbase. The fields can be either edit fields or combo boxes. The dialog has two buttons: **Update To Server** and **Cancel**.

If the user modifies a field value and then presses **Cancel**, they'll be asked if they want to update their changes to the server. When the dialog closes, the stamp values are updated accordingly.

When the **Update To Server** button is selected, any values the user has updated are saved immediately to the server. If the stamp entity that launched the **bi-directional** dialog is later deleted, or if the markup that the stamp was added to isn't saved, that doesn't affect the document attribute values that were previously saved to the server via this dialog.

Bi-directional stamp templates support **Value Assistance Fields**, which enable administrators to define a set of valid values for a bi-directional stamp template field. The Value Assistance Fields are configured through a Value Assistance attribute configuration of the content type being rendered in the Brava! Viewer. (Please see the OpenText Documentum CM Administrator documentation for information on creating a custom content type that is associated with the Value Assistance list that you define for user selection).

Once configured, pick lists can be enabled for `%dbupdatestring` stamps in both the ActiveX and HTML Viewers (`%dbupdatestring(`*`<picklist>`*). This feature enables users to select from a list of specified values from their repository. Below shows an example of the pic klist enabled in the ActiveX viewer dialog.

Token notes:

- The values to the `%dbstring` and `%dbupdatestring` fields are case-insensitive in terms of client processing. For example, `%dbupdatestring(Project)` is considered to be the same as `%dbupdatestring(project)`.

- If the server value for a particular token is set to a null or blank value, then the stamp displays a blank line as opposed to the term `<no value>`.
- If the `%dbupdatestring` is prefaced by `_readonly:` then the field is disabled in the dialog (i.e. `%dbupdatestring(_readonly:document_id)`.
- When a stamp is initially placed, values for the `%dbupdatestring` fields are retrieved from the server. The server's values (displayed in the **bi-directional** stamp dialog) can be more current than what is currently displayed in the stamp. If the user selects **Cancel** on the **bi-directional** stamp dialog, and if the parameter *`RefreshStampOnCancel`* is set to `true`, the values on the stamp are still updated with the latest values retrieved from the server to reflect the current status. There must be at least one token that's not a `%dbupdatestring` for the integration to detect initial stamp placement.
- If the stamp contains a `%dbupdatestring` starting with `_rowHeight:`, the height of each row is set to the value after this prefix (i.e. `%dbupdatestring(_rowHeight:45)`. If this token isn't set, the parameter *`DbUpdateStringRowHeight`* sets the row height. If set to a value of less than 30 or is not set at all, its value is set to 30.
- If the stamp contains a `%dbupdatestring` starting with `_labelWidth:`, the width of the label column is set to the value after this prefix (i.e. `%dbupdatestring(_labelWidth:65)`. If this token isn't set, the parameter *`DbUpdateStringLabelWidth`* sets the width of the label column. If not set, then its width is calculated based on the longest token string.
- If the stamp contains a `%dbupdatestring` starting with `_editWidth:`, the width of the edit control column is set to the value after this prefix (i.e. `%dbupdatestring(_editWidth:170)`. If this token isn't set, the parameter *`DbUpdateStringEditControlWidth`* sets the width of the edit control column. If not set, then its width is calculated based on the longest value retrieved for the `%dbupdatestring` fields. The minimum length for the field is 85, and the maximum is 220.
- If the stamp contains a `%dbupdatestring` starting with `_function:`, then the value after the colon is passed as an http parameter to all server calls. For example, for `%dbupdatestring(_function:marketing)` would add the parameter `&dbupdatefunction=marketing` to the query string for HTTPGET server calls. For POSTcalls, this is added to the header.
- If the server does not recognize a field as one it can update, the field does not need to be returned in the response to the client. If a field is not returned in the client response, and the parameter *`DbupdatestringNotRetrieved`* is set to `Hide` then the client does not display that field in the **DbUpdateString** dialog. If this parameter is not set, the default is to show the field in the dialog with a blank value.

The HTML Viewers and ActiveX Viewer support interaction with document properties through the `%dbstring` and `%dbupdatestring` tokens. As illustrated in the following matrix, the viewer types offer slightly different availability for the functionalities listed:

Feature	ActiveX	Html
Stamp Template %dbstring(token)	Yes	Yes
Stamp Template %dbupdatestring(token) docbase content update	Yes	Yes
Stamp Template %dbupdatestring(token) docbase content display	Yes	No*
Stamp Template %dbupdatestring(*<picklist>*) docbase Value Assistance Fields	Yes	Yes*
DbupdatestringNotRetrieved	Yes	No
%dbupdatestring(_readonly:document_id)	Yes	No
%dbupdatestring(_rowHeight:45)	Yes	No
%dbupdatestring(_labelWidth:65)	Yes	No
%dbupdatestring(_editWidth:170)	Yes	No
%dbupdatestring(_function:marketing)	Yes	No

Note: *The Html Viewer enables users to apply Stamp Templates with %dbupdatestring(token) including **Value Assistance Field** properties. When the dialog within the viewer is launched in the HTML Viewer, the current value of the selected document property is not displayed. This is a known issue.

Chapter 19

Configuring the OpenText Extended ECM plugin

The OpenText Extended ECM plugin (`D2-xECM-Plugin-Install.jar`) is available for installation for use in the Smart View client. Once it is installed, the following items are added to client configuration:

- An **Extended ECM** > **Options** menu, where workspace mapping can be configured.
- A **Create Business Workspace** menu item in **Menu Smart View** > **Actions Toolbar** > **+ Menu** This item is a placeholder, and during runtime will be replaced with whatever is configured in **Extended ECM** > **Options**. Normally, this menu item does not need to be manipulated by the administrator.

In **Extended ECM** > **Options**, add Extended ECM server URL information and Template Mappings, which includes the **Apply to folder path** settings, which control where the **+** menu items can appear in the Smart View client.

See the Extended ECM documentation for more information about settings and workspaces.

Chapter 20

Configuring the OpenText InfoArchive plugin

The InfoArchive plugin allows administrators to add InfoArchive tiles to the Smart View client landing page, giving users quick access to information stored in InfoArchive. The tiles are pre-configured query forms for specific InfoArchive applications.

Note: OpenText Directory Services (OTDS) SSO is strongly recommended to ensure a seamless user authentication experience with InfoArchive. OTDS SSO for InfoArchive is mandatory for integration. If InfoArchive employs its default authentication scheme, integration will fail.

If OTDS is not enabled, but is set up for InfoArchive, you must create a new OAuth Client corresponding to OpenText Documentum CM, and Redirect URLs under that client must include: `http(s)://<d2smartview_host:port>/<d2smartview_app>/ui/oauth2-login-cb`

Perform the following steps to set up InfoArchive integration:

1. During installation, select the InfoArchive plugin.
2. After successful installation, in client configuration, navigate to **Tools > InfoArchive**.
3. Complete the server connection details for both your **InfoArchive Server** and **OTDS Server**.

   Click the **Test Connection** button to ensure the connection works for both the InfoArchive and OTDS servers. Once the **Test Connection** button is pressed, the system tries to establish a connection between Gateway and the InfoArchive or OTDS server, depending on which server you are testing. The system will inform you if the connection could be made or not.

   If the connection is not successfully established, the error message indicates the reason why the connection failed. Make the necessary changes indicated in the error message and click the **Test Connection** button again.
4. Click **Save**.
5. In client configuration, navigate to **Go to > Property Page**, then click **New**.
6. Create a custom Property Page that allows your users to search for information in your InfoArchive application.

   a. Click **New**, and add a **Name** for your Property Page.
   b. In the **Document type** field, select `dm_sysobject`.

      Note: When you are adding each of the search controls to the Property Page, ensure that **Do Not Link to Property** is selected. The

Control id entry must match the search criteria that is declared in InfoArchive.

If you are adding controls to your Property Page that require data sources (for example, combo control, list control) and leave the **Datasource** field blank, the UI fetches the data from the InfoArchive server. Only **Specified Manually** and **Derived from XQuery** values are supported.

c. Click **Save**.

7. In client configuration, navigate to **Widget View > Widget**, then click **New**.

8. In the **Widget Type** field, select `InfoArchiveWidget`.

9. In the **Parameters** section, complete the following fields:

   a. In the **Server configuration** field, select the server you defined in **Tools > InfoArchive**.

   b. In the **Application** field, enter the name of your InfoArchive application.

   c. In the **Search Form** field, enter the name of the InfoArchive application’s Search Form.

   d. In the **Property Page** field, select the InfoArchive Property Page you created earlier.

   e. In **Parameters mapping**, define the **Operators** for the search fields you created in the Property Page.

   Note: In Table vs SIP archive types, the Operator is mandatory only for SIP types. The “between” operator is mandatory for both types, and you should choose it for range type search criteria. In property page range fields, there must be two controls named <param_name>_from and <param_name>_to (case sensitive).

10. Save your widget, then add it to the Configuration Matrix.

11. In client configuration, navigate to **Widget View > Smart View Landing Page** and add the InfoArchive tiles to your Smart View landing page xml file (one tile per InfoArchive widget you prepared). Add a tile container with individual tiles, as shown in this sample:

```
<tile-container size="full">
    <tile name=”Sample IA Tile 1” type=”InfoArchiveWidget”
        <theme color=”shade4” type=”teal1”/>
    </tile>
<tile name=”Sample IA Tile 2” type=”InfoArchiveWidget”
        <theme color=”shade4” type=”teal2”/>
</tile>
</tile-container>
```

12. When users log into the Smart View client, they can click the InfoArchive tiles you have set up, which will open a query form that allows them to search for applicable InfoArchive content.

Chapter 21

Configuring OpenText Intelligent Viewing

Intelligent Viewing provides a cloud-designed, feature-rich collaboration viewer and transformation engine. OpenText Directory Services (OTDS) is required for authentication purposes.

For more information about comparison of features and format support across the viewers, see “BravaCSR Viewer, Brava! Enterprise Viewer, and Intelligent Viewing feature reference“ on page 605

Note: OpenText Intelligent Viewing should not be used in concert with BravaCSR viewer or Brava! Enterprise Viewer as markups and annotations are not cross-compatible.

1. Intelligent Viewing is available as a Microsoft Windows install, or a cloud-deployable container. Follow the instructions in the *Windows Install Guide: OpenText Intelligent Viewing* or, for cloud deployment the *OpenText Documentum Content Management Cloud Deployment Guide*, then, in client configuration, perform the following setup actions:
2. Navigate to **Tools > OpenText Viewing**.
3. Complete the **URL Details** section by providing the server URLs for the following Intelligent Viewing services. In a cloud deployment. these URLs can be found by opening a PowerShell command window and entering the following command:

```
helm get all otiv -n <namespace>
```

   - **Highlight Service URL**

     Note: Post-Windows install, this information can be found in <Install directory>\Intelligent Viewing\<service name>\.env

   - **Markup Service URL**

     Note: Post-Windows install, this information can be found in <Install directory>\Intelligent Viewing\markupservice\.env

   - **Publication Service URL**

     Note: Post-Windows install, this information can be found in <Install directory>\Intelligent Viewing\SearchService\.env

   - **Viewer Service URL**

     Note: Post-Windows install, this information can be found in <Install directory>\Intelligent Viewing\ViewerService\.env

4. Complete the **OTDS Details** section by providing the following OTDS information:

   - **OTDS Service URL**

     Note: You can find information about the next two settings in the `.env` and `env.conf` files for each respective service:

   - **OTDS Client ID**
   - **OTDS Client Secret**

5. Complete the **Redaction Scripts** and **Redaction Groups** sections. These sections define redaction rules and which groups have redaction applied to documents they interact with.

   - **Redaction Scripts**: A list of scripts that define what content to redact in documents. To edit the list:

     – Click **...** to browse your system for a script to upload.

     – Click **Download** to download the selected script to review or edit it.

     – Click **Delete** to delete the selected script from the list of scripts.

   - For more information on creating redaction scripts, see https://bravadocs.bp-paas.otxlab.net/docs/core-viewing-usage-redactions/ and *OpenText Blazon Enterprise - Redaction Module Advanced User Guide (CLBRVW-SBZ)*.

   - **Redaction Applied to Groups**: This section includes a list of all the groups in your configuration and allows you to select which groups to apply the redaction rules to. Use the arrows to move groups between the list of available groups (no redaction applied) and the groups which have redaction applied. Users of these groups have redaction applied to the documents they interact with, based on the configured redaction scripts.

6. Complete the **Job Parameters** section by providing the following values appropriate to your environment:

   - **Publication Cache Deletion Limit Percentage**: The storage percentage to be cleaned up after max storage threshold for publications cache has reached on server. Default is 25 percent.

   - **Publication Cache Size (in MB)**: The max storage of the volumes configured for publications artifacts. The volumes paths are configured using `MKONDO_BLOB_ROOTS` and `MKONDO_BLOB_ROOT_STORAGE_LIMITS` fields in the Intelligent Viewing services installation script

   - **Publication Cache Size Threshold Percentage**: The max storage threshold after which deletion job should cleanup publications cache on Intelligent Viewing server. Default is 90 percent.

7. Click **Save**.

8. Navigate to **Widget View > Widget** and click **New**.

9. Type a **Name** for the Intelligent Viewing widget.

10. In the **Widget type** field, select `OpenTextIntelligentViewingWidget`.

11. Configure the different versions of the viewer, as necessary. Each configuration is contained on its own tab: **Viewer**, **Text compare**, and **Graphical compare**.

    Available on all tabs:

    - Select **Disable Annotations** if you want to restrict your users from annotating content in the viewer.
    - You can download and customize the configuration file:
      - Download the sample configuration file from the following location: `\webapps\D2-Config\utils\d2opentextintelligentviewing\example_opentext_intelligent_viewing_config.json`
      - Edit the `.json` file as required. See "Customizing the Open Text Intelligent Viewing Configuration file" on page 412 for more information.
      - In the **OpenText Intelligent Viewing Configuration file** field, click **...** and import your edited `.json` configuration file.

        Note: Refresh the client configuration cache every time an updated `.json` file is uploaded.

    Viewer tab

    - Select **Override PDF configuration** to display the native document and not the PDF version created based on PDF configuration. If not selected, settings defined in PDF configurations are applied when a user views a document in the viewer.

      Note: If you do not use PDF configurations for Intelligent Viewing, to improve the client's performance, you can still enable this setting to allow Intelligent Viewing to use the cached version of the document.

    - For **Watermark Text** and **Banner Text**, type in the desired text string or enter a standard token that resolve in the displayed output. Watermarks appear over the document, and banner text appears along the bottom of the document.
    - You can use multiple tokens in one banner or watermark string (for example, `%User: %Date %Time` might resolve as "tjones: 06/08/2022 1:27 PM Eastern Daylight Time"). See section 5.3.1 "Tokens" in *OpenText Documentum Content Management - Intelligent Viewing User Help (CLIVCS-H-UDM)*.

Custom tokens (`%dbstring<field value>`) are also supported that allow communication with the integration database and dynamically update to insert the requested field value.

Text compare
Select **Enable text compare** to enable comparison of text content in the viewer through the Text Compare menu.

When enabled, enter the text compare supported formats separated by a comma (e.g., txt,pdf,doc,docx).

Graphical compare
Select **Enable graphical compare** to enable comparison of graphical content in the viewer through the Graphical Compare menu.

When enabled, enter the graphical compare supported formats, separated by a comma (e.g., tif,jpeg,dwg).

12. Click **Save**.

21.1 Customizing the Open Text Intelligent Viewing Configuration file

A `.json` file is used to customize the look of the Intelligent Viewing interface in the Smart View client by dynamically adding or removing panes, panels, and components. You can download a sample file from: `\webapps\D2-Config\utils\d2opentextintelligentviewing\example_opentext_intelligent_viewing_config.json`

- To disable toolbar items from the viewer, remove components as required from the `.json` file. For example:

```
{
            "component":"SaveButton"
         },
         {
            "component":"SelectButton"
         },
         {
            "component":"PanButton"
         },
         {
            "component":"ZoomToRectangleButton"
         },
         {
            "component":"RotateButton"
         },
         {
            "component":"ZoomInButton"
         },
         {
            "component":"ZoomOutButton"
         },
         {
            "component":"ZoomExtentsButton"
         },
         {
            "component":"ZoomWidthButton"
         },
```

```
         {
            "component":"ExportButton",
            "format":"pdf"
         },
         {
            "component":"ExportButton",
            "format":"tiff"
         },
         {
            "component":"ExportButton",
            "format":"print"
         },
```

- To remove options from the left-hand pane of the viewer, delete the following from the `.json` file, for example:

```
"leftTabContainer":{
      "sidebarName":"leftTabContainer",
      "primary":"primary",
      "tabs":[
         {
            "component":"ThumbnailPane",
            "title":"tab.thumbnails"
         },
         {
            "component":"BookmarksPane",
            "title":"tab.bookmarks",
            "layoutKey":"bookmarks"
         },
         {
            "component":"MarkupPane",
            "title":"Markups",
            "layoutKey":"markupTools"
         },
         {
            "component":"MarkupDetails",
            "title":"Detail"
         },
         {
            "component":"StampRasterPane",
            "title":"tab.stampRaster",
            "layoutKey":"markupTools"
         },
         {
            "component":"TakeoffPane",
            "title":"Takeoffs",
            "layoutKey":"measureTools"
         },
         {
            "component":"NavigationPane",
            "title":"tab.navigation",
            "layoutKey":"markupFilters"
         },
         {
            "component":"SearchResultsPane",
            "title":"tab.searchResults"
         },
         {
            "component":"LayersPane",
            "title":"tab.layers"
         }
      ]
   },
```

21.2 Configuring multi-file support

Intelligent Viewing can display multiple items as if they were a single document, also known as a binder. In this view, users navigate through the multiple documents as if they are the pages of the same document.

By default, the maximum number of items that can be added to a binder is 20. You can change this maximum by modifying *maxNumberOfFilesForMultiFileViewIV* in the *<Tomcat_Directory>*\<SmartViewFolder>\WEB-INF\classes\D2FS.properties file.

21.3 Intelligent Viewing help link

Intelligent Viewing services version 22.2 and later includes a link to Intelligent Viewing help information. If you are using 22.1 or earlier, you must disable the link. If you do not disable the link, an error appears when a user opens the viewer.

To remove the help link, open the relevant config.json file and remove the following text:

```
{
 "component": "HelpButton",
 "helpUrl": "",
 "windowFeatures": "width=800, height=600"
},
```

21.4 Migrating Brava! markup to Intelligent Viewing

Intelligent Viewing supports migrating document markup done in Brava! viewer to Intelligent Viewing. Once enabled, when a user opens a document in Intelligent Viewing that had markup added in Brava! viewer, the markup comments are migrated to Intelligent Viewing. Migration is not immediate and so the markup comments do not appear when the user first opens the document, but once the markup comments are migrated, the user is prompted to reload the document with markup enabled.

To enable Brava! markup migration, you must enable *enableMarkupMigrationToIV* in the D2FS properties file. You must also update OTDS configuration by adding the *read_any_markups* and *write_any_markups* scopes to the Smart View client ID and IV-publisher client ID.

1. A redaction markup title update will only happen if the IV oAuth client user is assigned to a regular license, otherwise, the migration will fail.

   For example, if we have oAuth client iv-d2sv:
   a. In OTDS, access the **Users & Groups** area.
   b. Check for iv-d2sv.
   c. Assign a regular license to iv-d2sv.

2. When you open a document with migrated markups in the Brava! viewer, you cannot add additional markups - the markup option is disabled.

3. When you open a content item in OpenText Intelligent Viewing, any annotations in the item (existing or new) are migrated to OpenText Intelligent Viewing. If you open the item in BravaCSR or OpenText Brava! Enterprise at a later time, the annotations feature will be turned off and you will not be able to edit or create annotations.
4. If you compare documents in Brava! viewer and one document has annotations disabled, annotations are disabled for both documents while in that view.
5. A document is considered migrated when a user opens it in Intelligent Viewing while it contains the *`markupMigrationEnabled`* flag.

21.5 Adding custom stamps (raster images)

You can add custom images (rasters) to the Intelligent Viewing integration that users can add to documents as markups. Images must be a `.png` or `.jpg` file format. To do add these images, update the Intelligent Viewing Configuration file (`opentext_intelligent_viewing_config.json`) with the following image information:

```
"imageRaster" :[
{ "name": "<name>", "uri": "image uri with same host", "mimeType": "imageType" }
{ "name": "<name>", "uri": "image uri with same host", "mimeType": "imageType" }
]
```

Chapter 22

Configuring Digital Signature Plugins

OpenText Documentum CM integrates with digital signature providers to allow members of your organization to digitally sign documents. This feature is supported in Smart View Workflows.

This chapter includes instructions specific to the different digital signature provider, as well as general digital signature configuration instructions. To set up a digital signature provider, follow the instructions in their relevant section. These sections include links to any required general configurations steps.

Here are the configuration steps for each supported digital signature providers:

- OpenText Core Signature: “Configuring OpenText Core Signature” on page 417
- DocuSign: “Configuring DocuSign” on page 419

Notes

- Digital signature solutions, such as OpenText Core Signature, can also be referred to as electronic signature solutions.

  In this and other product guides, “digital signature” refers to wet-ink signature technology mixed with cryptographic technology. References to “electronic signature” refer to when a user confirms an action by providing login credentials. The user can provide identity provider (IDP) credentials or basic credentials depending on the how the user logged into Smart View.
- Smart View, Client Configuration, and the Java method server (`DmMethods`) require access to the internet, either directly or through a proxy, to communicate with digital signature APIs.

22.1 Configuring OpenText Core Signature

The following configuration steps are required to use OpenText Core Signature with Smart View:

1. **Obtain configuration information from OpenText Admin Center:** OpenText Admin Center is an administration tool included with a subscription to OpenText Core Signature. From it, you must obtain details to configure OpenText Core Signature in client configuration, including the Tenant ID, Client ID, and Client Secret. For more information, see “Obtain configuration information from OpenText Admin Center” on page 418.
2. **Create a digital signature workflow:** Users access OpenText Core Signature features through workflows in Smart View. You must create specific workflows to use the OpenText Core Signature functionality. For more information, see “Creating or updating a digital signature workflow” on page 423.

3. **Configure OpenText Core Signature as a digital signature provider:** You must configure OpenText Core Signature as a digital signature provider in client configuration. Here you must add the configuration details that you collected from the OpenText Admin Center. For more information, see "Configure Core Signature as a digital signature provider" on page 419.
4. **Configure signature options for the digital signature workflow element:** In client configuration, you must configure the digital signature options for the workflow element used for your digital signature workflow. These options are under the **Signature** section of the workflow options. For more information, see "Configure signature options for the digital signature workflow element" on page 424.
5. **Activate *`D2JobCoreSignStatusUpdate`*:** You must manually activate a OpenText Core Signature job that is added as part of the installation. For more information, see "Activate digital signature jobs" on page 424.

22.1.1 Obtain configuration information from OpenText Admin Center

OpenText Admin Center allows tenant administrators to support applications, subscriptions, users, and user access. In OT2 OpenText Admin Center, you can configure settings at the tenant and the subscription level.

This section provides details on how to collect information from the OpenText Admin Center that is required to set up OpenText Core Signature in client configuration. For information on how to setup your site, subscription, and clients for Core Signature API access, see the "OpenText Admin Center" documentation.

Collect or configure the following information from OpenText Admin Center:

- **Tenant ID:** The Tenant ID is used as the **Site ID** in client configuration. Obtain the Tenant ID on the **Details** tab.
- **Client ID:** A unique ID for OAuth Clients. Obtain the Client ID under **API service credentials** of the tenant subscription.
- **Client Secret:** A secret code only used between the OAuth Client and the Authorization Server. The code is generated in OpenText Admin Center when you create the Client ID.

  Important
  The Client Secret only appears once when you create the Client ID so ensure that you copy it and save it somewhere secure.
- **Redirect URLs:** When you create a Client ID in OpenText Admin Center, you need to add the redirect URLs for OpenText Documentum CM. This is the URL you use for Smart View. The Redirect URLs in the API service credentials must use HTTPS URLs. So make sure Smart View runs in HTTPS mode.

  For example: `https://myserver:8443/D2-SmartView`
- Enable the **signature:tenant-viewer** role on the **API service credentials** tab.

This option ensures that a signature request is cancelled if the corresponding workflow is aborted.

22.1.2 Configure Core Signature as a digital signature provider

Provide Admin Center details:

1. In client configuration, open **Tools** > **Signature Providers** > **Core Signature**.
2. Enter the details collected in "Obtain configuration information from OpenText Admin Center" on page 418

   - **Core Signature Base URL:** The Core Signature API URL.

     For example: `https://sign.core.opentext.com/api/v1/`
   - **Site ID:** This is the Tenant ID obtained from Admin Center.

     For example: `24ac65ed-b931-4994-b55d-31c8a3d26f1g`
   - **Client ID:** This is the Client Secret obtained from Admin Center.

     For example: `3c2f09b2-401e-467e-83bd-e3ecc96d07fe`
   - **Authorization Endpoint URL**

     For example: `https://otdsauth.ot2.opentext.com/otdstenant/24ac65ed-b931-4994-b55d-31c8a3d26f1g/login`
   - **Token Endpoint URL:** The OAuth endpoint on the authorization server where the OAuth Client exchanges the authorization grant, client_id, and client secret for an access token.

     For example: `https://otdsauth.ot2.opentext.com/otdstenant/24ac65ed-b931-4994-b55d-31c8a3d26f1g/login`

22.1.3 Sample application

A sample application is available in `d2-doclibrary-sampleapp.zip`. It contains an example workflow with the required OpenText Core Signature tasks.

22.2 Configuring DocuSign

The following configuration steps are required to use DocuSign with Smart View:

1. **Install DocuSign:** The client installer provides an option to include specific plug-ins during installation. To install the DocuSign plug-in, select the **DocuSign** check box in the installer interface. This action installs the required DAR and JAR files and applies the necessary configuration changes. For more information, see section 20 "Digital signature components" in *OpenText Documentum Content Management - Client Installation Guide (EDCCL-IGD)*.
2. **Access the DocuSign portal and gather configuration requirements:** To begin integrating DocuSign with OpenText Documentum CM, you must first create a DocuSign account and access the DocuSign portal. From it, you must obtain

details to configure DocuSign in client configuration. For more information, see "Access the DocuSign portal and gather configuration requirements" on page 420.

3. **Grant DocuSign asynchronous access**: To enable asynchronous operations such as the signature tracker job, you must grant DocuSign appropriate access. This process is only required once, and allows the system to generate tokens on behalf of the user. For more information, see "Grant DocuSign asynchronous access" on page 421.

4. **Configure DocuSign as a digital signature provider:** You must configure DocuSign as a digital signature provider in client configuration. Here you must add the configuration details that you collected from the DocuSign portal. For more information, see "Configure DocuSign as a digital signature provider" on page 422.

5. **Configure a digital signature workflow:** Users access digital signature features through workflows in Smart View. You must create new workflows or update existing workflows to use DocuSign functionality. For more information, see "Creating or updating a digital signature workflow" on page 423.

6. **Configure signature options for the digital signature workflow element:** In client configuration, you must configure the digital signature options for the workflow element used for your digital signature workflow. These options are under the **Signature** section of the workflow options. For more information, see "Configure signature options for the digital signature workflow element" on page 424.

7. **Activate *`D2JobDocuSignStatusUpdate`*:** In OpenText Admin Center, you must manually activate a DocuSign job that is added as part of the installation. For more information, see "Activate digital signature jobs" on page 424.

22.2.1 Access the DocuSign portal and gather configuration requirements

To integrate DocuSign with OpenText Documentum CM, you must create a DocuSign account and gather configuration information.

1. Go to https://www.docusign.com and create an account. This account allows you to test your integration using DocuSign's API and tools.

2. After account creation, go to the DocuSign portal to:

   - Generate your Integration Key (Client ID)
   - Set up Redirect URIs
   - Create and store your Secret Key and Private Key
   - Retrieve your API Account ID and User ID

3. Once your integration is tested and ready for production, complete the DocuSign Go-Live process. This process requires submitting your application for review to validate API usage.

For more information about DocuSign Account Setup, see DocuSign documentation.

22.2.2 Grant DocuSign asynchronous access

To enable asynchronous operations such as the signature tracker job, you must grant DocuSign appropriate access. This process is only required once, and allows the system to generate tokens on behalf of the user.

1. 1. Construct the consent URI using the following format:

   ```
   <DOCUSIGN_ACCOUNT_URI>/oauth/auth?response_type=code&scope=signature
   %20impersonation&client_id=<CLIENT_ID>&redirect_uri=<REDIRECT_URI>
   ```

   Use the following values when constructing the consent URL:

   - *<DOCUSIGN_ACCOUNT_URI>*:
     – **Demo**: `https://account-d.docusign.com`
     – **Staging**: `https://account-s.docusign.com`
     – **Production**: `https://account.docusign.com`
   - *<CLIENT_ID>*: Integration key used in client configuration.
   - *<REDIRECT_URI>*: Client (Smart View) URI configured under Redirect URIs.

   For example:

   ```
   https://account-d.docusign.com/oauth/auth?response_type=code&scope
   =signature%20impersonation&client_id=12345678-a538-472e-9e81-
   d027e7123456&redirect_uri=https%3A%2F%2Fwww.acme.com%2FSmartview
   ```

2. 2. Paste the URI in a browser and click **Allow Access** when prompted.

   Note: If you are not logged in to DocuSign, you are prompted to log in.

3. Ensure the *<REDIRECT_URI>* parameter is URL-encoded.

   For example:

   `redirect_uri=https%3A%2F%2Fwww.acme.com%2FSmartview` instead of `redirect_uri=https://www.acme.com/Smartview`

22.2.3 Configure DocuSign as a digital signature provider

1. In client configuration, open **Tools** > **Signature Providers** > **DocuSign**.
2. Enter the details collected in “Access the DocuSign portal and gather configuration requirements” on page 420

   - **DocuSign Base URL**: Enter the appropriate DocuSign application URL. For example:
     - **Demo**: `https://apps-d.docusign.com/`
     - **Staging**: `https://apps-s.docusign.com/`
     - **Production**: `https://apps.docusign.com/`
   - **Account ID**: Enter the API account ID from DocuSign.
   - **User ID**: Enter the DocuSign user ID used for tracking signatures.
   - **Client ID**: Enter the Integration key generated in DocuSign.
   - **Client Secret**: Enter the Secret key generated in DocuSign.
   - **Private Key**: Enter the DocuSign RSA private key used for asynchronous authorization with DocuSign REST APIs.
   - **Authorization Endpoint URL**: Enter the appropriate DocuSign authorization endpoint URL. For example:
     - **Demo**: `https://account-d.docusign.com/`
     - **Staging**: `https://account-s.docusign.com/`
     - **Production**: `https://account.docusign.com/`
   - **Token Endpoint URL**: Enter the appropriate DocuSign token endpoint URL. For example:
     - **Demo**: `https://account-d.docusign.com/oauth/token`
     - **Staging**: `https://account-s.docusign.com/oauth/token`
     - **Production**: `https://account.docusign.com/oauth/token`
3. Click **Test Connection** to verify that the application server can reach the DocuSign URLs.
4. Click **Save** to apply the configuration.

22.3 General digital signature configuration steps

This section includes configuration information applicable to all digital signature plug-ins.

22.3.1 Creating or updating a digital signature workflow

To send documents for digital signatures, a workflow must have with specific activities necessary for a digital signature workflow. Use Workflow Designer to create a digital signature workflow. The workflow can include multiple tasks, but two tasks are required: a signature request preparation task and a tracker task to hold the workflow until all signatures are collected.

- **Signature request preparation:** This task does not require any special configurations when you create it in Workflow Designer. However, when you configure the workflow element in client configuration, you must configure specific digital signature settings on the **Signature** tab.
- **Tracker task:** This task holds the workflow and does not allow it to advance until the digital signature process is complete. This task requires specific details to support digital signatures:
  - On the task’s **Performer** tab, set **Performers** to a predefined specific group **d2_digital_signature_tracker** and **Processed By** to **Individual users and group members at the same time**.
  - If you have configured a reject flow for the tracker task, ensure on the **Transition** tab of the reject flow, you select **Let performer select next activities**. If this is not selected, the workflow proceeds even when the signature request is declined or cancelled.

For more information about creating a workflow with Workflow designer, see “Using Workflow Designer” on page 148.

Note: This guide and the user interface sometimes uses the terms “task”” and “activity” interchangeably. An activity is a task that a user must perform on the workflow to pass the document to the next activity in the workflow. The client configuration and client applications typically use the word “task”, while Documentum CM Server and Workflow Designer typically use the word “activity”.

22.3.2 Configure signature options for the digital signature workflow element

You must configure the digital signature options of the workflow element used by your digital signature workflow. These are configured under the Signature section of the workflow element configuration.

For more information, see this workflow configuration step.

22.3.3 Activate digital signature jobs

Based on which digital signature plugin you install, a corresponding job is added as part of the installation. Here are the possible jobs:

- OpenText Core Signature: *`D2JobCoreSignStatusUpdate`*
- DocuSign: *`D2JobDocuSignStatusUpdate`*

These jobs are a critical component of the integration of digital signatures with workflows. By default they are inactive and you must manually activate it.

The job polls the digital signature service on the status of pending signature requests. Depending on the status of the request, the workflow proceeds or waits. Ensure that you have approved the DFC client **DmMethods** (under JMS) as a privileged client.

For more information about jobs in Documentum Administrator, see section 9.1 “Jobs” in *OpenText Documentum Content Management - Administrator Online Help (EDCAC-H-UGD)*.

Chapter 23

Configuring OpenText Content Aviator

OpenText™ Content Aviator is an intelligent assistant powered by artificial intelligence (AI) that allows you to ask natural language questions about or to summarize the contents of a folder.

If your administrator has turned on Content Aviator in your environment and designated folders in your environment as *Aviator assignments*, you can open an interactive chat pane and initiate natural language conversations to get information about those folders and their contents.

If you ask your question in a supported language, Content Aviator will attempt to respond using the same language.

Note that your data is protected and will not be used to train large language models (LLMs).

Content supported by Content Aviator

When a folder or virtual document is designated as an Aviator assignment, users can ask questions about its contents. Users can also ask Content Aviator to summarize content.

Content Aviator supports content when the following conditions are met:

- Content is in a folder or virtual document designated as an Aviator assignment.
- The user has permission to access the content.
- Content has a native format with searchable text (full-text indexable). Renditions are not supported.
- An administrator has assigned a Content Aviator license to the user.

When Content Aviator is available

Content Aviator is available in Smart View when each of the following is completed:

1. An administrator has turned Content Aviator on in Admin Console and designated one or more folders or virtual documents as Aviator assignments. See section 2 “Configure Content Aviator” in *OpenText Documentum Content Management - Content Aviator Administration Help (EDCADC-H-AAV)*.
2. An administrator has configured client configuration to activate Content Aviator chat in Smart View. See section 23.1.1 “Ensuring that Content Aviator chat is available to users” in *OpenText Documentum Content Management - Client Configuration Help (EDCCL-H-AGD)*.

3. The user has the required permissions to access an Aviator assignment folder or virtual document and its contents:
   - Browse permission on the folder or virtual document.
   - Read permission on the contents in the folder or virtual document.
4. An administrator has assigned a Content Aviator license to the user.

23.1 Configuring Content Aviator in client configuration

An administrator must configure, turn on, and set up the environment for Content Aviator by following these procedures:

1. Configure Content Aviator in Admin Console. See section 2 “Configure Content Aviator” in *OpenText Documentum Content Management - Content Aviator Administration Help (EDCADC-H-AAV)*
2. “Ensuring that Content Aviator chat is available to users” on page 426

23.1.1 Ensuring that Content Aviator chat is available to users

To allow users to access Content Aviator chat, you need to enable the chat action:

To make Content Aviator chat available to users:

1. Sign into client configuration as a system administrator.
2. From the menu bar, navigate to **Go to** > **Menu Smart View**.
3. On the **Menu Smart View** page, set the following:
   - **Menu properties > Label**: Aviator
   - **Menu properties > Action > Action**: Show dialog with manager “AviatorChatManager”
   - **Menu properties > Action > Parameters > Manager**: “AviatorChatManager”
   - **Conditions**: Selection is ready for Aviator Chat
   - **Contextual menus > Action Toolbars**: Aviator
4. The condition is validated for a user, and that user can access Content Aviator chat provided the following is *true* for that user:
   1. Content Aviator is turned on. See section 2 “Configure Content Aviator” in *OpenText Documentum Content Management - Content Aviator Administration Help (EDCADC-H-AAV)*.
   2. A folder has been assigned as an Aviator assignment. See section 2.2.1 “Designate folders” in *OpenText Documentum Content Management - Content Aviator Administration Help (EDCADC-H-AAV)*.
   3. The user has the necessary permission:

- `BROWSE` permission to Aviator assignments.
- `READ` permission on a document within Aviator assignments.

5. See "Configuring Smart View client Menus" on page 304.

Chapter 24

Microsoft Sensitivity Labels

Sensitivity labels from Microsoft Purview Information Protection allow organizations to classify and protect Microsoft 365 documents across devices, apps, and services. Sensitivity labels control access to content and can be applied with and without protection. Examples of sensitivity labels are *Public*, *Confidential*, and *Highly Confidential*.

Support for sensitivity labels allows OpenText Documentum Content Management (CM) to store and process documents with sensitivity labels while preserving the protection defined by those labels.

Support for sensitivity labels includes the following:

- **Protected storage:** Enables users to store labeled content with label information stored and visible.
- **Protected content processing:** Provides full-text indexing and thumbnail generation for content specified with sensitivity labels.
- **Protected transformation:** Enables PDF renditions of labeled content to be protected with the same sensitivity label as in the original document.

The following document types are supported:

- Microsoft Word (.doc, .docx, .docm)
- Microsoft PowerPoint (.ppt, .pptx)
- Microsoft Excel (.xls, .xlsx)
- Email (.msg)
- Adobe® Acrobat® (.pdf)

24.1 Configuring Sensitivity Labels

Sensitivity labels are enabled and configured in Admin Console. See *OpenText Documentum Content Management - Microsoft Integrations Administration Help (EDCADC-H-AIN)*.

After sensitivity labels are enabled and configured, you can also configure Smart View to display them in a few locations, like in doclists or on properties pages. Here is a list of optional configuration steps:

> **Important**
> Currently, the only sensitivity label aspect property you should include on property pages or in doclists is the `first_mip_label` property of the

`miplabelinfo` aspect. Other aspects and their properties are not meant to be visible to Smart View users.

- You can list sensitivity labels on property pages. To do so, add a label and map it to the sensitivity label aspect property you want to display. For information on how to add a label to a property page, see "Configuring a Property Page" on page 48.
- You can add a sensitivity label column to doclist widgets. To do so, customize the default widget columns for a doclist widget, then select the sensitivity label aspect as the type, and then add the appropriate property to the list of displayed columns. For more information on how to customize the columns in a doclist widget, see "Configuring an Internal Widget" on page 276.
- You can allow a user to add a sensitivity label column to a doclist. To do so, add the sensitivity label aspect to the list of displayable properties in **Display configuration**. For more information on how to customize the allowable columns in a doclist, see "Configuring Client Column Preferences" on page 329.
- If you add a sensitivity label column to a doclist, you must add it to quick and query search indexing so that its data appears in searches. To add to quick search, see section 7.3.2 "Making types and attributes searchable" in *OpenText Documentum Content Management - Search Administration and Development Guide (EDCSRCH-AGD)*. For query searches, do one of the following:
  - Ensure that you enable **Add the user column preferences**. If you have added the sensitivity label aspect property to the list of displayable properties a user can add to a doclist, the query search indexes its data.
  - If you do not enable **Add the user column preferences**, you must add the fully qualified aspect property to the DQL query and enable **Add mandatory attribute in hidden mode** so users can perform actions on the items in returned by the query. Here is an example query with a sensitivity label aspect property included:

    ```
    Select r_object_id, object_name, miplabelinfo.first_mip_label from dm_document
    ```

Note: Support for sensitivity labels is available only on cloud deployments.

Chapter 25

Using Specs

25.1 Configuring Specs Module Descriptions

You can modify the descriptions of modules. Specs use the descriptions when generating a specification document of your configuration.

1. Navigate to **Tools** > **Modules description** from the menu bar.
2. Type a description for modules used.
3. Click **Save**.

Related Topics

"Generating a Specifications Document" on page 431

25.2 Generating a Specifications Document

1. Navigate to **Specifications** > **Generate configurations** from the menu bar.
2. Fill out the form as described in the following table:

Field	Description
Applications	Select the application for which the specification is generated.
Display elements without application	Select to display configuration components that are not assigned to applications.
Include dictionaries	Select to include dictionaries.
Include taxonomies	Select to include taxonomies.
Include registered tables	Select to include registered tables.

3. Click **OK**.

Related Topics

"Configuring Specs Module Descriptions" on page 431

Chapter 26

Using Retention Policy Services (RPS)

26.1 Understanding the RPS Plug-in

The RPS plug-in adds two modules to the configuration matrix to allow you to manage and use retention, markup, and hold policies in lifecycles:

- Retention policy configuration: Allows management of the preservation of repository content through a retention policy configuration.
- Retention markup: Allows usage of RPS markup policies. Policies can be designated as types: *Hold*, *Freeze*, *Review*, *Permanent* or *Vital* per the RPS Markup configuration defined in Records client.

To use retention policies, configure policies using Retention Policy Services. The *OpenText OpenText Documentum CM Records Client Administration and User Guide* contains further information.

Apply the retention, or markup policies you create to contexts in the configuration matrix and associate the context with a policy. For instance, to apply a particular RPS policy to a specific type of document when the status is *APPROVED*, create a context configuration that is scoped to the specific document type and attribute value, and associate the context configuration with the retention and/or markup policy in the matrix. If the same document type should have another retention policy associated with it when the document status is *OBSOLETE*, then create another context for that document type and attribute value, and associate the context with the retention and/or markup policy in the matrix. A single context configuration can be associated with one or more RPS policies in the matrix.

When you have the policy configurations and contexts created in the matrix, you can then use the policies in a lifecycle using the `D2ApplyRetentionMethod`, `D2ApplyMarkupMethod` methods. These methods do not take any parameters or arguments. When a method is invoked by the lifecycle action, the RPS policies that apply are associated with the matching document context configured in the matrix.

To apply retention or markup policies to folders or documents, create a job in OpenText Documentum CM Administrator using “D2ApplyRententionmethod” with the following argument:

```
-selectedIds id1¬id2¬id3¬id4¬idn
```

Where *`id1`*, *`id2`*, *`id3`*, *`id4`*, and *`idn`* are folder or document IDs.

For the job to run successfully, create a context configuration that includes the specific attribute value and object types (folders or documents), and associate the context configuration with the retention and markup policies in the matrix.

Related Topics

"Adding RPS Policies to the Configuration Matrix" on page 434

26.2 Adding RPS Policies to the Configuration Matrix

1. Navigate to:

   - **D2–RPS** > **Retention policy** from the menu bar for retention policies.
   - **D2–RPS** > **Retention markup** from the menu bar for markup policies.

2. Fill out the form as described in the following table:

Field	Description
Name	Select the policy that was configured in RPS.
Description	Type a description.
Application	Add or remove the applications to which this document link applies. For example, adding the `QA` application would cause the policy to only apply to matching quality assurance cases.

3. Click **Save**.

Related Topics

"Understanding the RPS Plug-in" on page 433

Chapter 27

Using the Recycle Bin

27.1 Configuring the Recycle Bin

The Recycle Bin plug-in adds a recycling-bin approach to managing document removal. If the Recycle Bin is:

- Not installed, content deleted by end users is permanently removed.
- Installed, content deleted by end users is moved to the electronic recycling bin, which allows end users and functional administrators to view, permanently remove, and restore content.

Note: When the Recycle Bin is enabled, a user is not able to delete a document when they do not have permission to access the parent folder. Users who do not have minimal privileges (BROWSE on parent folder structure, for example) cannot retrieve `dm_folder` objects to maintain the proper links.

Additionally, the user requires read permission on the document in order to move it to a different location (Recycle Bin). In the case of Extended delete permission, the user requires at least read permission on the document to delete if the Recycle Bin is enabled.

1. Navigate to **Recycle Bin** > **Options** from the menu bar.
2. Fill out the form as described in the following table:

Field	Description
Group allowed to administrate every recycle bin	Select the user group to have administrative privileges for the Recycle Bin widget.
Users can destroy documents located in their own recycle bin	Select to allow users to permanently remove content in their own recycle bin.
Send an email warning when the global size of the recycle bin is more than <*value*> MB	Enter a value in megabytes. When the combined size of all user recycle bins becomes larger than the input size, the system sends an email warning.
Send an email warning when the size of a user recycle bin is more than <*value*> MB	Enter a value in megabytes. When the size of any individual user recycle bin becomes larger than the input size, the system sends an email warning.
Send email warning to	Select the user group to send email warnings.

Field	Description
Max days in Recycle bin for user	Enter the maximum days a document can exist in the recycle bin. Once a file exceeds this value, the `D2BinJobDeleteExpiredObject` job deletes the file from bin.

3. Click **Save**.

Related Topics

"Incorrect Properties for Content in the Recycle Bin" on page 436

27.2 Incorrect Properties for Content in the Recycle Bin

Due to the way the repository handles content in the recycle bin, the following properties display incorrect values:

- `a_content_type`
- `i_chronicle_id`
- `i_contents_id`
- `i_antecendent_id`
- `i_direct_dsc`
- `i_latest_flag`
- `i_vstamp`
- `i_folder_id`
- `r_folder_path`
- `r_version_label` - The value for `r_version_label` shows the correct version but also incorrectly labels the content in the recycle bin as the `CURRENT` version.
- `r_link_cnt`
- `r_modifier`
- `r_modify_date`
- `r_page_cnt`
- `r_content_size`
- `r_full_content_size`
- `r_immutable_flag`

Related Topics

"Configuring the Recycle Bin" on page 435

Chapter 28

Optimizing options

28.1 Optimizing DQL Queries

You should structure queries, especially those with large numbers of attributes or table references, in the following manner:

- Group attributes by the object type table (_s, _r) that contains them, starting with the parent table in the object hierarchy.
- Order attributes in the same order (left-to-right) as in the database table definition.

If you do not structure it as explained, it can result in severe performance issues (minutes versus seconds) when the database contains millions of rows.

28.1.1 Using DollarValue Strings in DQL

Several DollarValue strings are available for use in multiple DQL contexts. See below for examples:

DollarValue String Items	Example Usage
$alias	$alias(<*aliassetname.aliasname*>)
$value	'$value(<*att.dico.locale*>)' '$value(<*att.format*>)' (for use with time attribute)
$count	$count(<*att*>)
$dqlvalue	$dqlvalue(<|"*att*|">)
$note	$note(<*att*>)
$repeatingvalue	$repeatingvalue(<*att*>) $repeatingvalue(<*att.format*>) (for use with time attribute)
$skin	$skin(<*att*>)
$Repeatingskin	$Repeatingskin[]
$foreachdoc	$foreachdoc{}

DollarValue String Items	Example Usage

`$foreachworkflownote`	`$foreachworkflownote{}`
	For example:
	`$foreachworkflownote{$note(date.dd/MMM/yy)`

28.1.2 Using attributes to access default date and time formats for external widgets

For external widgets, you can use the following attributes to access the default date and time formats from the client's preferences:

Attribute	Description
`$DEFAULT_DATE_FORMAT`	Used to retrieve the default date format from preferences.
`$DEFAULT_DATE_TIME_FORMAT`	Used to retrieve the default date and time format from preferences.

28.2 Optimizing DQL filtering on multiple attributes

You can configure DQL so that you can perform filter operations for multiple attributes together. For example, this supports the scenario where a user wants to fetch all the users from the `dm_user` table where `user_name` or `user_address(email)` contains a string: “test”.

Configure the multiple attribute DQL by following this pattern:

```
<Select query with any attributes or *> where <condition1> AND <space><Filter Attribute
name 1> <space>$value(filter_ex)  or <space><Filter Attribute name 2><space>
$value(filter_ex) and <condition2>  or<space> <Filter Attribute name 3> <space>
$value(filter_ex)
```

For single attribute filtering, the pattern is: filtering attribute name, then space, then `$value(filter_ex)<space><Filter Attribute name 1> <space>$value(filter_ex)`.

For multiple attribute filtering, both the conditions are need to be separated by an operator (`and / or`): `user_name $value(filter_ex) or user_address $value(filter_ex)`.

Note: Both DQL approaches are supported, but cannot be combined in a single DQL statement.

In the multiple attribute DQL filter type, you must disable the client side filtering to list all the results. The **No client filtering** check box is available only for combo fields. It is not available for list combo fields.

Sample DQL:

```
select user_name as name, user_name from dm_user where user_name
$value(filter_ex)
```

```
select user_name as name, user_name as label from dm_user where user_name
$value(filter_ex) or user_address $value(filter_ex)
```

```
select user_name as name, user_name from dm_user where r_is_group=0 and
 (user_name $value(filter_ex) or (user_address $value(filter_ex)))
```

28.3 Optimizing Content Import

You can configure OpenText Documentum CM to run actions, such as a lifecycle action, during content import. Enabling **Execute actions on start** increases the back-end processing to complete the transaction and the complexity of the actions further impacts the processing time. To prevent unnecessry performance overhead, enable **Execute actions on start** for a configuration only if the actions are required.

Related Topics

“Optimizing Property Pages” on page 441

“Optimizing Searches” on page 442

“Optimizing Classic View Workspaces” on page 443

28.4 Optimizing Property Pages

Select **Load asynchronously** when configuring property pages that contain combo or list fields that perform a DQL query. When the option is enabled, the DQL runs when the end user selects the specific property instead of when the property page is loaded. OpenText Documentum CM does not need to perform the DQL query when loading a property page because the query is not necessary when the end user is viewing instead of editing. Avoiding unnecessary DQL queries improves the response time for displaying the property page and reduces the load on the server and the database.

If you use an asynchronous DQL query:

- Use the `'$value(filter)'` or `'$value(filter_ex)'` wildcard to enable filtering. If you do not use the wildcard, the query returns all values when an end user selects a list box.

  If you load asynchronously, a DQL query for loading values must include `as name`, `as label`, and `'$value(filter)'`

  For example, `select user_name as name, user_name as label from dm_user where '$value(filter)' order by 1`

- Limit the returned result set by using the `enable (return_top <number of results>)` keyword. To use this keyword, set the `return_top_results_row_based` parameter in the Documentum CM Server `server.ini` file to `F`.

- Set the **Auto load first results** field to load all, a set number, or no results when a user clicks the field in the property page. This allows you to match the behavior of synchronous combo fields in your pages. The field recognizes the following values as triggers:
  - `-1`: The default setting for new property pages and upgrades, keeps current behavior intact where the user must click the combo field's down arrow itself.
  - `0`: Indicates all of the values in the list will autoload up to the `maxResultSetSize` or `maxListAssistanceResultSetSize` from `d2fs.properties`. If those settings are not set, then the system defaults are used.
  - `###` : indicates some value greater than 0. Regardless of the input value, the system always honors `maxResultSetSize` or `maxListAssistanceResultSetSize`.

The response time for loading a property page depends on the number of components on the property page. Take advantage of contexts to create and configure targeted property page definitions instead of a single comprehensive property page.

Related Topics

"Optimizing Content Import" on page 441

"Optimizing Searches" on page 442

"Optimizing Classic View Workspaces" on page 443

28.5 Optimizing Searches

Use the following guidelines to optimize performance of searches:

- Use a full-text search engine in a large-scale environment because by default the search operations use DQL queries, which may be inefficient on large data sets.
- You can create function-based indexes to improve search performance, but it is difficult to predict the attributes that OpenText Documentum CM uses for case insensitive searches. Each additional function-based index consumes additional storage and can cause space issues.
- Set the **Maximum results returned by search** to a smaller value in the Global search configuration page to reduce the impact of searches with a large number of results.
- Query form searches do not use the **Maximum results returned by search** value nor the `dfc.search.max_results` parameters in `dfc.properties`. Make sure to design the DQL query for query forms to be more selected or use the `enable (return_top <number of results>)` keyword to limit the returned result set and reduce performance overhead.

Note: You can use the d2c_query_dq object only in versions 3.x. In 4.x versions, you must use query forms to perform a search operation. In the 4.5

version, you can also use a DQL Intelligent URL to run a query and get multiple results in a doclist.

Related Topics

Related Topics

"Optimizing Content Import" on page 441

"Optimizing Property Pages" on page 441

"Optimizing Classic View Workspaces" on page 443

28.6 Optimizing Classic View Workspaces

The number of widgets displayed on a web page can affect response time, particularly in WAN conditions with low latency. Construct workspaces focused on specific tasks instead of a comprehensive workspace for all possible scenarios.

When configuring a workspace, note that widgets on a workspace do not share query results. For example, if a workspace contains both a Doclist widget and a Searches widget, running an Advanced Search with facets triggers the same full-text query twice: one to update the result list in the Doclist widget and one to update the facet tree in the Searches widget. In a high workload scenario, this uses more search engine resources.

Related Topics

"Optimizing Content Import" on page 441

"Optimizing Property Pages" on page 441

"Optimizing Searches" on page 442

28.7 Parallel Streaming Configuration

Parallel streaming enables faster downloads. The file is divided into smaller segments and transferred or downloaded in multiple concurrent threads.

Note: Network latency and bandwidth restrictions have an impact on parallel streaming.

Parallel streaming download is available only if the download URL targets an ACS or a BOCS server. If your organization has not deployed any BOCS servers, that is permissible. A download URL will target an ACS server only if:

1. `D2-BOCS=true` and `includeAcsServer=true` in `D2FS.properties`.
2. The file to be downloaded is not a PDF file or Microsoft Office file when the PDF or transfer configuration plugins respectively have been deployed.

Regarding *(1)*, note that BOCS must be deployed to ACS whenever `D2-BOCS=true` and `includeAcsServer=true` even though BOCS itself will not be involved when files are downloaded directly from ACS. This is because upload URLs will also target the BOCS/ACS server when D2-BOCS=true and includeAcsServer=true, so file upload will be broken if BOCS has not been deployed to ACS. Likewise, download URLs will target the BOCS/ACS server in this case when the PDF or transfer configuration plugins are involved, and if BOCS has not been deployed to ACS, then such download requests will fail.

Regarding *(2)*, if the PDF or transfer configuration plugins are not being used, PDF or Microsoft Office files can be downloaded directly from an ACS or BOCS server. If the PDF Configuration plugin is being used, then outbound content manipulation of the requested PDF file can occur. For example, the plugin can insert a watermark or an additional page into the downloaded PDF file based on configuration.

Likewise, if the transfer configuration plugin is being used, then outbound content manipulation of the requested Office file can occur. For example, the plugin can inject custom properties into the downloaded Office file based on configuration.

This is one reason why BOCS is needed: neither ACS nor BOCS have an extension framework which would allow the PDF or transfer configuration plugin code to run and manipulate the downloaded content according to relevant configurations. A download request is handled by BOCS/ACS by first fetching the file from the adjacent ACS server and then performing the desired content manipulation on this file before serving it up to the client. Likewise for BOCS/BOCS: the download request is handled by first fetching the file from the adjacent BOCS server (which in turn will fetch it from ACS if it is not already in the BOCS server cache) and then performing the desired content manipulation before serving it up to the client. It should be clear that when a file is downloaded through BOCS (for example, it is a PDF or Office document that may entail some content manipulation), parallel streaming download is not possible.

Parallel streaming is controlled using the parameters defined in the `X3Applet.properties` file. Administrator can edit the `X3Applet.properties` file located at `<webapp location>\D2\applet\X3Applet.properties`.

Following table describes the parameters of `X3Applet.properties`:

Parameter	Description	Values
download.useParallel	Enables or disables parallel streaming.	<*True*>: to enable parallel streaming <*False*>: to disable parallel streaming

Parameter	Description	Values
download.maxNumberParallelDownloadStreams	Indicates the maximum number of concurrent threads required to download the content. **Note:** Administrator can set **download.maxNumberParallelDownloadStreams** to the required maximum value, provided the settings match the ACS or BOCS server settings.	Integer
bocs.url.prefix	Indicates the prefix for the BOCS URL configured in the repository.	`/bocs/servlet/ACS`
acs.url.prefix	Indicates the prefix for the ACS URL configured in the repository.	`/ACS/servlet/ACS`
download.forceParallelSizeInMB	Indicates the size in megabyte that forces parallel streaming to begin the download. Any file size greater than this value, triggers parallel streaming.	Integer

The `X3Applet.properties` file gets downloaded to `{User Home}\ Documentum\D2\ X3Applet.properties` when the user accesses OpenText Documentum CM for the first time.

Note: If the user modified any of the settings in the Base App `/applet/ X3Applet.properties`, they will have to delete the `X3Applet.properties` file downloaded to `@<USER_HOME>/Documentum/D2`.

Example:

download.useParallel = truedownload.maxNumberParallelDownloadStreams = 15bocs.url.prefix=/bocs/servlet/ACSacs.url.prefix=/acs/servlet/ ACSdownload.forceParallelSizeInMB = 500

In this example, parallel streaming is enabled (download.useParallel = true). A file of size greater than 500 MB, initiates download using 15 concurrent threads.

28.8 Cleaning your Browser Cache

Clearing your browser cache is sometimes required to improve performance times.

Clear your browser cache:

- After a configuration change.
- After the deployment of an update.
- After other similar activities.

This ensures that the client receives the updated values from client configuration.

When using WSCTF, clearing your browser cache may cause slower performance because you need to reconnect to WSCTF. The WSCTF JavScript code uses browser storage to track the WSCTF port.

28.9 Improving virtual document load times (Classic)

By default, when a user displays a virtual document, all levels of the document are listed in the structure. Depending on the size of the virtual document, this can cause long load times. To improve virtual document load times, enable the *`fetchVDPartially`* D2FS property to limit how many levels of the virtual document display by default. When enabled, only the first two levels of the virtual document display. To view documents in the structure deeper than two levels, the user can expand the structure as necessary.

28.10 Temporary file management

The temporary file management process, named `D2FileCleanTracker`, allows you to control when and how temporary files are stored and removed from the app server. It can be configured in the `D2FS.properties`, `D2-Config.properties`, and `D2-BOCS.properties` files. Here are its parameters and their default values:

- *`disableCleanupProcess`*`=false`
- *`maxTempFiles`*`=2000`
- *`maxTempSizeMB`*`=1024`
- *`fileCleanSchedulerInterval`*`=15`
- *`printPendingRecords`*`=false`
- *`enableTimeBasedCleanup`*`=false`
- *`timeToLive`*`=5`
- *`bulkImportTimeToLive`*`=15`

For more detailed information on these parameters, see *OpenText Documentum Content Management - Client Installation Guide (EDCCL-IGD)*.

`D2FileCleanTracker` maintains two lists: **remove** and **hold**.

Files on the **remove** list are deleted based on how you configure the above parameters. The `D2FileCleanTracker` process run when any of the following occur:

- When the number of files on the **remove** list reaches the *`maxTempFiles`* limit.
- When the total size of files on the **remove** list reaches the *`maxTempSizeMB`* limit.
- If *`enableTimeBasedCleanup`* is enabled, when the number of minutes defined in *`fileCleanSchedulerInterval`* pass since the last time the process ran.

When the process runs, it removes all files from the **remove** list that are older than the number of minutes defined in *`timeToLive`*.

Files on the **hold** list are not deleted by the `D2FileCleanTracker` process until the server restarts.

28.11 Concurrent Login Management (Classic)

By default Classic users can run concurrent sessions from different browsers on the same machine, or the same browser on different machines. To restrict this functionality, in the `d2fs.properties` file, set the property *`allow_concurrent_login`* to false. When restricted, a user will receive a notification that concurrent logins are not allowed if they attempt to login from different locations concurrently.

To clean up entities related to logged in users go to `D2\utils\d2cleanuputil` and run `D2UserSessionCleanupUtil.cmd` / `D2UserSessionCleanupUtil.sh` to clean all the entries in `d2_session_store`.

Note: You can run this utility when the Tomcat/application server goes down.

Use `d2cleanusersessionsmethod` to clean up specific user name, server IP, application URL entries in `d2_session_store` with the following argument:

```
apply,c,NULL,DO_METHOD,METHOD,S,d2cleanusersessionsmethod,ARGUMENTS,S,'-
cleanup_type <cleanup type> <cleanup detail>'
```

The <*cleanup type*> and <*cleanup detail*> options are as follows:

- `CLEANUP_BY_USER_NAME` with `-user_name` and then the user_login_name of the dm_user object. For example:

  ```
  apply,c,NULL,DO_METHOD,METHOD,S,d2cleanusersessionsmethod,ARGUMENTS,S,
  '-cleanup_type CLEANUP_BY_USER_NAME -user_name Administrator'
  ```

- `CLEANUP_BY_SERVER_IP` with `-server_ip` and then the *`server_ip`* of the OpenText Documentum CM application server. For example:

  ```
  apply,c,NULL,DO_METHOD,METHOD,S,d2cleanusersessionsmethod,ARGUMENTS,S,
  '-cleanup_type CLEANUP_BY_SERVER_IP -server_ip 128.192.2.1'
  ```

- `CLEANUP_BY_APPLICATION_URL` with `-application_url` and then the application URL. For example:

```
apply,c,NULL,DO_METHOD,METHOD,S,d2cleanusersessionsmethod,ARGUMENTS,S,
'-cleanup_type CLEANUP_BY_APPLICATION_URL -application_url https://
csws242ssl.otxlab.net:8443/D2'
```

Chapter 29

Configuring Your First Application

29.1 Use Case: Configuring as Basic Content Library

Users from various user communities use the repository as a content library or repository of records. If you are new to the client or migrating from another client such as Webtop, we recommend that you implement the following configuration steps:

Note: Ensure that OpenText Documentum CM is installed and you have a user account to access client configuration.

1. Create a dictionary to map the key value for your document type to the equivalent language. Example: Use the key**doc** and map it to English as **Generic Document**. For more information, refer to Configuring a Dictionary.
2. Create a property page for your documents. If the document types have shared attributes, you can create one property page for multiple document types or create multiple property pages for each document type. For more information, refer to Configuring a Property Page.
3. Create a creation profile and perform the following:
   a. Choose your dictionary.
   b. Select the key value from the dictionary.
   c. Map the key value to a document type.
   d. Map the key value to your property page.

   For more information, refer to Configuring Creation Profiles.
4. Create the following widgets to interact with the content:
   a. Doclist
   b. Repository Browser
   c. Property Page
   d. Renditions
   e. Relations
   f. Versions
   g. Favorites
   h. Searches
   i. Facets

Note: To use the Facets widget, a Search server is required.

For more information, refer to List of Widgets.

5. To contain the widgets, create a workspace. for more information, refer to Configuring a D2 Client Workspace.
6. Create a context. In the context, you can define the permissions for a group of users to use the property page, widgets, and workspace configurations. For more information, refer to Configuring Contexts.
7. In the matrix, place a checkbox at the intersection of the group context and property page. For more information, refer to Organizing the Configuration Matrix.

To configure an advanced Content Library, perform the following additional steps:

1. Create a context specific to a document type rather than for a group of users. For example, a Procedure Document.
2. Add a auto link configuration to place the defined document type in a particular location.
3. Add a security configuration to define a set of permission for the document created.
4. Add a lifecycle configuration for documents to transition from `Draft` to `Released`.
5. Design a workflow in Workflow Manager or Process Designer and add a workflow configuration that utilizes it.

29.2 Quick Start Steps for your Smart View client solution

If you have an existing OpenText Documentum CM solution set up, but want to add the Smart View client, review this checklist for getting started quickly:

1. Review the capabilities of the Smart View client. See “Understanding the Smart View client” on page 237 and related topics for details.
2. Create new widgets for use on the Smart View landing page as needed. Replace location-based repository browser widgets with location-based doclist widgets. Create a new Recently Accessed widget if desired. See “Configuring an Internal Widget” on page 276.
3. Create a new Smart View landing page configuration file. Include existing doclist widgets and any new location-based doclist widgets. Existing Favorites, Checked-Out, and Saved Search widgets can be included along with any new Recently Accessed widgets. See “Designing a Smart View and Mobile Landing Page” on page 260 for more information.
4. Create a new Smart View theme configuration. See “Configuring a Smart View client Theme” on page 274.

5. Create a new Smart View menu configuration. See “Configuring Smart View client Menus” on page 304.
6. Set new configurations against the appropriate context in the matrix.

29.3 Extracting Email Attachments and Adding them as Linked Documents

If your users import email messages with attachments into their repositories, you can configure the import to extract and add the attachments as separate files that are related to the parent email and each other.

1. Configure the importation of email attachments and define the file types you want to allow into the system, as described in “Configuring Importing Email Attachments” on page 128.

2. Define a creation profile for linked documents, as shown in “Configuring Creation Profiles” on page 120.

3. Define a linked documents configuration, as described in “Configuring Linked Documents for Importing Emails” on page 127 . In the linked documents configuration, choose a **Link type**, for example, **Relation**. For the relation you can choose anything that is relevant to your use case, but do not use `dm_` as a relation name prefix. For example:

relation_name	child_parent_label	parent_child_label
`is_loreum_ipsum`	`loreum_ipsum_file`	`is_loreum_ipsum`
`corp_email_attachment`	`is_email_attachment`	`has_email_attachment`
`user_mail_attachment`	`mail_attachment`	`mail_message`

The following is sample DQL for relation creation, using one of the examples presented in the table:

```
create dm_relation_type object set child_parent_label = 'is_email_attachment',
set parent_child_label = 'has_email_attachment',
set description = 'connects email message to extracted attachments'
set parent_type = 'corp_email'
set child_type = 'corp_email_attachment'
set direction_kind = 0
set integrity_kind = 0
set relation_name = 'corp_email_attachment',
set security_type = 'SYSTEM'
```

Notes

- `parent_type` and `child_type` could be anything (including dm_document) that fits the use case.
- The `security_type` example assumes you don’t want anyone but sysadmin or superuser to control the creation of the relationships. In other words, users cannot create this on their own using Create Relations actions, but must let the system do it for them through the import email creation flow you have created.

For more information about Relations, see the *Documentum CM Server System Object Reference Guide*.

Note: Any new relation will only show up by `relation_name` when the applications restart.

4. Build an Email Context in **Goto > Context** where something in your system can be used to identify the object being created is an email, such as the type or some attribute.

5. Select the Email Context in the Linked document configuration.

6. Optionally, if you want to take properties from the email and populate into the attributes of the email object, you need a transfer configuration, as explained in “Configuring Transfer Configurations” on page 383. For example, where properties are whatever your object attributes are for capturing the information, the # in the **Office property** configuration are default values used to map the Outlook fields to your given attributes that are on your object type that represents the email message, for example, `corp_email`.

Chapter 30

Date and Time Configuration

This chapter provides information on how configure time and date settings as well as how they display for end users.

30.1 Time Zone Setting

You can control how the client user interfaces display the date and time, either using the end user's time zone or the server time zone. To change this setting, you must modify the *`enableClientTimezoneAwareness`* setting in the `D2FS.properties` file.

Here are the options:

- `true`: The client user interface displays the time and date according to the user's time zone.
- `false`: The client user interface displays the time and date according to the server's time zone.

Notes

- The default setting for Classic is `false` (server time) and the default for Smart View is `true` (user's local time).
- Time zone includes relevant daylight saving time.
- Date values stored in a string are treated as plain text and are not affected by this setting.
- Date and time information is stored in UTC format.

30.2 Understanding Date Formats

Date and time is displayed using the time zone of the client that performs a create/update/delete action on an item. Daylight savings time is also considered (relative to the current time of the client machine).

Note: In Documentum CM Server , all date and time information is stored in UTC format.

You can configure date and time formats by changing their pattern strings. The system interprets unquoted letters from A to Z and a to z as patterns representing the components of a date or time string. You can surround text in single quotes to avoid interpretation. The following table describes the defined letters:

Letter	Component	Presentation	Example
G	Era designator	Text	AD
y	Year	Year	1996 or 96
Y	Week year	Number	3
M	Month in year	Month	July or Jul or 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday or Tue
a	AM/PM marker	Text	PM
H	Hour in day (0–23)	Number	0
k	Hour in day (1–24)	Number	24
K	Hour in AM/PM (0–11)	Number	0
h	Hour in AM/PM (1–12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time or PST or GMT-08:00
Z	Time zone	RFC 822 time zone	–0800

All other characters from A to Z and a to z are reserved.

Make use of rules for presentations to determine formatting. The following table describes how different string contexts affect a presentation:

Presentation	Rule
Text	If the number of pattern letters is 4 or more, the full form is used. Otherwise a short form is used if available. For parsing, both forms are accepted independent of the number of pattern letters.

Presentation	Rule
Number	The number of pattern letters is the minimum number of digits required. If there are not enough numbers, the amount is zero-padded. For parsing, the number of pattern letters is ignored unless needed to separate adjacent fields.
Year	If the number of pattern letters is 2, the year is truncated to 2 digits. Otherwise the year is treated as a number. For parsing with the abbreviated year pattern (y or yy), the year is abbreviated relative to the century. The date is adjusted to be between 80 years before and 20 years after the creation of the SimpleDateFormat instance. For example, if a date using the format MM/dd/yy is created January 1, 1997, the string 0/11/12 is treated as January 11, 2012. The string 05/04/64 is treated as May 4, 1964. Note that a string must consist of exactly two digits as defined by Character.isDigit(char) to be parsed into the default century. Other numeric strings, such as a single-digit string, are parsed literally. For example, 01/02/3 is treated as January 2, 3 AD.
Month	If the number of pattern letters is 3 or more, the month is treated as text. Otherwise the month is treated as a number.
General time zone	If the time zone has a name, it is treated as text. If the time zone represents a GMT offset value, the following syntax is used: GMTOffsetTimeZone: GMT *<Sign>* *<Hours>*: *<Minutes>* Where <*Sign*> is + or –, <*Hours*> is between 0 and 23, and <*Minutes*> is between 0 and 59. For parsing take digits from the Basic Latin block of the Unicode standard. The locale is independent. RFC 822 time zones are accepted for parsing.

Presentation	Rule
RFC 822 time zone	For formatting use the following syntax: `RFC822TimeZone:` *`<Sign>`* *`<TwoDigitHours>`* *`<Minutes>`* Where <*Sign*> is + or –, <*TwoDigitHours*> is between 00 and 23, and <*Minutes*> is between 0 and 59.

Use the following table for examples of how OpenText Documentum CM interprets date and time pattern strings:

Date and Time Pattern	Result
yyyy.MM.dd G 'at' HH:mm:ss z	2001.07.04 AD at 12:08:56 PDT
EEE, MMM d, ''yy	Wed, Jul 4, '01
h:mm a	12:08 PM
hh 'o''clock' a, zzzz	12 o'clock PM, Pacific Daylight Time
K:mm a, z	0:08 PM, PDT
yyyyy.MMMMM.dd GGG hh:mm aaa	02001.July.04 AD 12:08 PM
EEE, d MMM yyyy HH:mm:ss Z	Wed, 4 Jul 2001 12:08:56 –0700
yyMMddHHmmssZ	010704120856–0700

Chapter 31

Widgets Reference

31.1 List of Widgets

Any change to widget configurations in client configuration requires **Tools > Refresh Cache** to be performed to purge cached data in the AppServer.

Use internal repository navigation widgets as described in the following table:

Name in client configuration	Name in the client	Description
BrowserWidget	Browser	Displays cabinets and folders. Enables uncluttered browsing of a repository tree. Required for use in a workspace when an IntelligentURL locateID is used to locate a document or folder object.
BusinessWorkspaceWidget	Workspace	Displays items of a Smart View workspace. **Smart View only.**
DoclistWidget	Document list	Displays content and virtual documents. Enables full browsing and content-seeking capabilities. Use this widget to access the content context menu. Displays results from Quick search and Predefined search widgets. In Smart View, filter configurations will not be applied if the Doclist widget contains DQL content.

Name in client configuration	Name in the client	Description
DocgalleryWidget	Thumbnail	Enables thumbnail previews for browsing content. You must set up a thumbnail server to show and customize thumbnails.
FavoritesWidget	Favorites	Displays a list of content marked as favorite.
CollectionWidget	Collections	Displays a list of file collections. Select **Display on header panel** to display the collections button on the client header. The **Participants** parameter returns a list of users and/or groups that you can share a collection with. All the objects returned by the query will be **dm_user** or **dm_group** type. All other types will be excluded from the response. For the **dm_user** type, include the **r_object_id**, and **user_name** attributes in the query. Optionally, you can also include the **user_address** attribute to return the user email. For the **dm_group** type, include the **r_object_id**, and **group_name** attributes in the query. If the text box is left empty all of the users and groups in the system will be returned.
CheckoutDocumentWidget	Checkout	Displays a list of checked-out content with user name and time of checkout.

Name in client configuration	Name in the client	Description
CenterStageWidget	CenterStage browser	Enables browsing of CenterStage spaces. Only displays the Collaboration folder.
CommentsWidget	Comments	Enables viewing, creating, deleting, and replying to comments about content.
RecycleBinWidget	Recycle Bin	Enables managing and restoring deleted content. If you have administrator privileges, you can access the recycle bins of other users.
RecentlyAccessedWidget	Recently Accessed	Recently Accessed is only supported in Smart View client. Displays a list of the last 25 files a user has created, viewed, edited or checked out (as configured by the administrator). **Note:** For users to see items in this widget, they must have the `Audit view` privilege and you must select the **Audit config** dropdown in the widget's configuration.
SharedDocumentWidget	My External Shares	SharedDocumentWidget is only supported in Smart View client. Displays documents that the user has shared externally using a service such as OpenText Core Share.

Name in client configuration	Name in the client	Description
OpenURLWidget	Open URL	OpenURLWidget is only supported in Smart View client. Allows linking to external web sites or internal content and locations. Only static URLs without dynamic elements are supported. Relative paths are supported.

Use viewer widgets as described in the following table:

Name in client configuration	Name in the client	Description
BravaCSRViewerWidget	Brava Viewer	Displays the selected PDF or supported image format in the Brava CSR viewer. Select **Enable Chunking** to allow the Brava CSR viewer in the Smart View client to download and display large PDFs in smaller segments or chunks, so the user does not have to wait for the whole PDF to download before viewing can begin. Chunking support must also be turned on and configured in the `D2FS.properties` file. See the *OpenText Documentum Content Management - Client Installation Guide (EDCCL-IGD)* for more information. PDF Configuration-controlled views are not supported by the viewer when chunking is enabled. **Note:** If the user opens a document in the Brava CSR Viewer with a PDF View configuration, and the user chooses to **Print** or **Export** (Download) the document, the resulting output will be based on any PDF Print or Export configuration applied. If no PDF Print or Export applies, then the user will get the native PDF rendition without any overlay. If the PDF View configuration displayed in the Brava CSR Viewer has a different configuration than PDF Print or PDF Export, the output from **Print** or **Export** might not match what

Name in client configuration	Name in the client	Description
		the user sees in the viewer. User annotation tools in the viewer are enabled by default. If you want to restrict annotations for all users, select **Disable Annotations** in the widget configuration. Additionally, you can enter a regular expression to filter which documents the restriction applies to, based on based on file properties. With annotations disabled, users can use filter conditions inside the widget if filter conditions are enabled. Filter conditions include the following functions: • equals • notEquals • contains • notContains • isGreaterThan • isGreaterOrEqualsThan • isLessThan • isLessOrEqualsThan The following are examples of possible filter conditions: • (getValue(a_application_type).contains('hh')) • ((getValue(a_application_type).contains('hh'))&&(getValue(a_content_type).equals('dd'))) • ((getValue(a_application_type).contains('hh'))||(getValue(a_content_type).equals('dd')))

Name in client configuration	Name in the client	Description
BravaEnterpriseViewerWidget	Brava Enterprise Viewer Widget	Enables the three standard Brava! Enterprise widgets, which can be set up separately using the following radio buttons in the widget configuration: • **Brava! Viewer**: Standard Brava! Viewer functionality for displaying and annotating content in your repository. This widget helps in loading the HTML/Activex/Compare/Docmerge viewers. Select the **View selected document on Brava!** checkbox so user can view a document in the Brava! Viewer when navigating through documents in a DocList widget in Classic (this setting is not supported in the Smart View client). When this checkbox is not selected, the user can only view the selected document in the Brava! Viewer using the contextual menu option **View**, which appears on right click on selected document. • **Brava! Compare**: widget that allows you to compare documents in your repository and notes the differences. This widget helps to load two documents in comparison mode in Brava! Viewer widget and swap the documents being compared in the left side/right side. This viewer is not supported in the Smart View client.

Name in client configuration	Name in the client	Description
		• **Brava! Changemark**: widget for creating and tracking embedded notes and conversations. This widget helps display the Changemarks conversations (original comments/replies) on the document. It also contains quick links to launch the document with the selected changemark on the Brava! Viewer. This viewer is not supported in the Smart View client. The Brava! Server URL replaces the deprecated `ServerHostName` in Brava custom parameters. The format should be: `$PROTOCOL$://$APP_SERVER$:$APP_PORT$`
OpenTextIntelligentViewingWidget	Open Text Intelligent Viewing	OpenText Intelligent Viewing is a new, modern viewing service alternative that can be implemented instead of the Brava CSR viewer and the Brava Enterprise viewer. This product includes standard viewer features and document transformation services. OpenText Intelligent Viewing servers can be deployed as containers or set up using legacy software installation processes. **Note:** This viewer is available for the Smart View client only.

Use search widgets as described in the following table:

Name in client configuration	Name in the client	Description
QuickSearchWidget	Quick search	Enables full-text searches.

Name in client configuration	Name in the client	Description
SearchFormWidget	Search query form	Enables using pre-configured query form searches. Administrators configure an instance of the search query form widget for each query form option. Select the widget instance that matches the query form search you want to use.
SearchWidget	Searches	Enables viewing, editing, and running past and saved searches.
FacetsWidget	Facets	Displays facet categories based on the search results along with the facet values for each category. The facet selection will also be displayed in the doclist breadbox. The breadbox displays the facets selection in the order you added or modified the facets. Add the focus event `D2_EVENT_SEARCH_RUN` to automatically focus on the facet widget while doing a search operation. **Note:** If the user has multiple Classic View workspaces open, the facets widget for only the workspace they are currently viewing is refreshed after a search is run. In a workspace with multiple views, facets may refresh across all views in the workspace.

Use internal content properties widgets as described in the following table:

Name in client configuration	Name in the client	Description
PropertiesWidget	Properties	Displays properties of the selected content.

Name in client configuration	Name in the client	Description
DetailsLocationsWidget	Locations	Displays a list of folder locations in which the selected content is found. Classic client only.
LocationsWidget	Locations	Displays a list of folder locations in which the selected content is found. Smart View client only.
DetailsVersionsWidget	Versions	Displays a list of the versions of the selected content.
DetailsRenditionsWidget	Renditions	Displays a list of renditions of the selected content.
DetailsRelationsWidget	Relations	Displays a list of what is linked to the selected content. This widget manages content distribution by allowing you to start distributions, edit the list of recipients, and generate reports.
DistributionWidget	Distribution	Displays a list of distributions sent to the user. This widget allows you to accept, reject, and stop distribution tasks.
DetailsAuditWidget	Audit	Displays a list of audited actions for the selected content.
ThumbnailWidget	Preview	Displays the selected content as a slideshow. You must set up an ADTS rendition server to render previews and storyboards.
VirtualDocumentWidget	Virtual document	Displays virtual document structure.
DetailsSnapshotsWidget	Snapshots	Displays a list of virtual document snapshots for a virtual document selected in a Doclist or Virtual document widget.
DetailsRetentionsWidget	Retentions	Displays retention policies applied to the selected content.
DetailsMarkupsWidget	Markups	Displays markup policies for the selected content.

Use internal workflow widgets as described in the following table:

Name in client configuration	Name in the client	Description
TaskFoldersWidget	Tasks browser	Displays a list of tasks sorted by category that have been assigned to you. Use this widget to refresh the Workflow task list widget.
WorkflowOverviewWidget	Workflow history	Displays a list of past and current events to show workflow progress. In the Smart View client, this widget is used to place a workflow tile on the Landing Page. Additional Smart View parameters are available to configure the user's experience with the tile. You must have at least queue manager permission to access this widget. **Note:** **Date sent** information for in-progress workflows depends on activity type and dynamic activity configuration. **Date sent** does not appear at the activity-level after a workflow is completed.
WorkflowPreviewWidget	Graphical workflow	Displays a graphical view of the workflow. Use this widget to quickly determine the current status of a task in a workflow and the upcoming steps.
TasksWidget	Workflow task list	Displays a list of tasks that have been assigned to you. Use this widget to access the workflow context menu.

Name in client configuration	Name in the client	Description
TaskOverviewWidget	Task overview	Aggregates information from other task widgets into one widget for easy access. Includes a toolbar that allows the user to perform common actions on tasks, a Details panel containing summary and instruction content, a Note panel for annotating tasks, and full access to task reference documents.
TaskDetailsWidget	Task details	Displays the subject and message of selected task.
TaskNotesWidget	Task notes	Displays a list of accompanying notes to the selected workflow.
TaskAttachmentWidget	Task attachment	Displays a list of content attached to the selected task.
TaskPerformersWidget	Workflow performers	Displays a list of users organized by groups assigned to the workflow.

Use administration widgets as described in the following table:

Name in client configuration	Name in the client	Description
AdminUsersWidget	User	Displays a list of users for the repository. Use this widget to add, edit, and remove user accounts and to change group memberships. Classic client only.
AdminGroupsWidget	Group	Displays a list of user groups for the repository. Use this widget to add or edit user groups. Classic client only.

Name in client configuration	Name in the client	Description
AdminDictionariesWidget	Dictionaries	Displays a list of non-DQL based Dictionary configurations by configuration name and description. Use this widget to view, export, and configure dictionaries and values. **Note:** AdminDictionariesWidget is very slow with large dictionaries (300+ entries). Classic client only.
AdminTaxonomiesWidget	Taxonomies	Displays a list of repository taxonomies. Use this widget to view, export, and configure taxonomies and values. Classic client only.

Name in client configuration	Name in the client	Description
AdminWorkflowWidget	Admin Workflow history	Displays a list of past and current events to show workflow progress like `WorkflowOverviewWidget` but also allows managers and administrators to make changes to workflows that they do not own. Requires membership to `dmc_business_manager_role_dynamic` to view. For more information on this configuration, see “Configuring Business Admin Actions” on page 341. **Note:** Configuring a `AdminWorkflowWidget` requires the Smart View client to be approved as a privileged client. To approve it, the administrator must run the `d2privilegedclient` utility for Smart View. For help, see “d2privilegedclient” on page 621 in the Utilities Reference. Smart View client only.

Name in client configuration	Name in the client	Description
AdminCheckedOutDocumentWidget	Admin Checkout	Displays a list of checked-out content based on DQL like `CheckoutDocumentWidget` but also allows managers and administrators to cancel checkouts performed by other users. Requires membership to `dmc_business_manager_role_dynamic` to view. For more information on this configuration, see "Configuring Business Admin Actions" on page 341. **Note:** Configuring a `AdminCheckedOutDocumentWidget` requires the Smart View client to be approved as a privileged client. To approve it, the administrator must run the `d2privilegedclient` utility for Smart View. For help, see "d2privilegedclient" on page 621 in the Utilities Reference. Smart View client only.
AdminMipWidget	Admin MIP	Displays the Microsoft Purview Information Protection sensitivity labels widget that allows an administrative user to configure how the client handles documents that use sensitivity labels.

LocateTarget iURL support for widgets is listed in the following table:

Widgets that Work with LocateTarget	Widgets that do not work with LocateTarget
BrowserWidget	DetailsAuditWidget
DoclistWidget	ThumbnailWidget

Widgets that Work with LocateTarget	Widgets that do not work with LocateTarget
DocgalleryWidget	DetailsRetentionsWidget
FavoritesWidget	TasksWidget
CheckoutDocumentWidget	
CenterStageWidget	
CommentsWidget	
RecycleBinWidget	
OpenURLWidget	
QuickSearchWidget	
SearchWidget	
FacetsWidget	
PropertiesWidget	
DetailsRelationsWidget	
VirtualDocumentWidget	
DetailsRetentionsWidget	
TaskFoldersWidget	
TaskDetailsWidget	
TaskAttachmentWidget	
AdminUsersWidget	
AdminGroupsWidget	
AdminDictionariesWidget	
AdminTaxonomiesWidget	
DumpWidget	

Chapter 32

Jobs Reference

32.1 List of Jobs

Jobs are repository objects that automate when and how to run method objects. Methods associated with jobs are run automatically on a user-defined schedule. The properties of a job define if jobs should run and when they run.

OpenText Documentum CM installs the jobs to the repository during installation. You can configure jobs in an external software, such as OpenText Documentum CM Administrator, to perform scheduled tasks.

Reports are only generated for jobs when the `-save_report` parameter is set to `true` in arguments for the method. Jobs do not generate reports by default.

The following table describes the available jobs.

Name: D2CreateRendition **Description**: Request rendition of content and its components. **Default frequency**: Once per hour **Default state**: Inactive	**Parameters** • `id`: The content ID.

Name: D2ImportDictionary **Description**: Import or update dictionaries from Excel or .csv documents. **Default frequency**: Once per day **Default state**: Inactive	**Parameters** • `folder`: Path of folders containing the files separated by comma. • `separator`: Defines the separating character for .csv documents. • `overwrite`: Enables erasing the dictionary. • `sorted`: Enables sorting the dictionary. • `update`: Enables updating autolinking with the imported dictionary.

Name: D2JobBocsPreCache **Description**: The job will pre-cache documents on all BOCS servers using the BOCS Pre-Cache mechanism. The documents that are pre-cached are based on the BOCS Cache configurations and Document Contexts applied to them within the configuration matrix. For example, if a BOCS Cache configuration has been created with a network location, then that configuration on the matrix is checkmarked against a Context for documents which have a published status. **Default frequency**: Daily **Default state**: Inactive	**Parameters** None.

Name: D2JobClean **Description**: Delete unused `dm_acl` objects and annotations. **Default frequency**: Once per day **Default state**: Active	**Parameters** • `clean_native_note`: By default, enables cleaning of annotations. Set false to ignore annotations. • `clean_hyperlink`: Set to true to clean the d2_link orphan objects. Default is false. • `-clean_acl`: Flag used to destroy orphan acls. • `-clean_contentless`: Set to *`true`* to delete unprocessed and empty objects that are created by unsuccessful file import operations.

Name: D2JobCleanOtivPublication **Description**: Cleans the OpenText Intelligent Viewing Publications cache when publications become invalid because of document modifications or to create space when the cache is near capacity. **Default frequency**: Every 30 minutes **Default state**: Inactive	**Parameters** Configured in client configuration by selecting **Tools** > **OpenText Viewing**. The following options are available: • **Publication Cache Deletion Limit Percentage**: The percentage of storage to clear when the Publications cache reaches the maximum storage threshold, as defined by **Publication Cache Size Threshold Percentage**. Default is 25 percent. • **Publication Cache Size (in MB)**: The maximum storage amount for the Publication cache volumes. Volume paths are configured in the `MKONDO_BLOB_ROOTS` and `MKONDO_BLOB_ROOT_STORAGE_LIMITS` fields during OpenText Viewing Services installation. Default is 0 MB. • **Publication Cache Size Threshold Percentage**: The maximum storage threshold percentage. When the Publications cache reaches this threshold, the `D2JobCleanOtivPublication` job runs and clears the amount of space defined by **Publication Cache Deletion Limit Percentage**. Default is 90 percent.

Name: D2JobCore **Description**: Re-apply configurations to existing objects. **Default frequency**: Once per day **Default state**: Inactive	**Parameters** • `dql_filter select r_object_id from dm_document`: Set a DQL filter to select objects related to the job. • `create`: By default, does not apply the job to content during creation. Set true to apply during creation. • `naming`: If create is true, by default apply naming during creation. Set false to avoid applying naming. • `autolink`: By default, applies autolink for property regeneration. Set false to avoid property regeneration. • `security`: By default, applies security to the job. Set false to avoid applying security. • `apply_for_vd`: By default, does not apply the job to child content. Set true to apply to child content.

Name: D2JobCoreDeQueue **Description**: Apply D2CoreMethod on content located in d2c_mass_create_queue. **Default frequency**: Once per day **Default state**: Active	**Parameters** • scan_queue: By default, runs a scan on the queue. Set false to prevent the scan. • naming: By default, apply naming after the queue is scanned. Set false to apply naming when properties are regenerated. • create: Runs naming by simulating content creation.

Name: D2JobDelegation **Description**: Delegate all pending tasks for a user. **Default frequency**: Once per hour **Default state**: Active	**Parameters** None.

Name: D2JobDocbaseUpdate **Description**: Update autolinking after a dictionary modification. Update requests are posted in the d2_update_docbase_queue **Default frequency**: Once per day **Default state**: Active	**Parameters** None.

Name: D2JobInProcessObjectClean **Description**: In the Smart View client, when a user initiates creation or importation of documents a multistep process is started where objects are created in the docbase before the user clicks **Finish**. If the process is interrupted and the client is not able to cancel the operation, these objects will remain in the docbase and require cleaning. When this job is run, it will delete objects older than the supplied -age Custom Argument. If -age is not entered, the job will delete all dangling objects. **Default frequency**: Once per day **Default state**: Active	**Parameters**: Custom Argument: -age <Age of Objects in Minutes> Example: -age 120

Name: D2JobImportMassCreate **Description**: Import and fill content with properties based on XML files linked to each file. XML files must be composed of a root element, document attribute files, and repeating attributes listed under the value tag. <*filename*>-meta.xml enables import of XML content as well as metadata. **Default frequency**: Once per day **Default state**: Inactive	**Parameters** • `folder`: Path of folders containing the files separated by comma. • `imported_file_folder`: Optional parameter for defining the folder to place successfully imported files. The default setting creates a folder within the source folder. • `createconfig`: Optional parameter for setting the `object_name` of the creation profile to use. Default setting uses the `r_object_type` found in the XML file. *Documentum CM Server Fundamentals* contains further information about `r_object_type`. • `args`: Sets the document type to use for values. The parameter is only required when you set a `createconfig`. • `async`: Use asynchronous creation. Default setting is false. This parameter is no longer used due to the usage of **D2JobCoreDeQueue**. • `naming`: By default, apply naming during the creation process. Set false to apply naming when properties are regenerated. • `unicity`: By default, control is deactivated after naming. Set true to activate control.

Name: D2JobImportTaxonomy **Description**: Import or update taxonomies from Excel or .csv documents. **Default frequency**: Once per day **Default state**: Inactive	**Parameters** • `folder`: Path of folders containing the files separated by comma. • `separator`: Defines the separating character for .csv documents.

Name: D2JobLifecycleBatch **Description**: Change the lifecycle state of content based on context. **Default frequency**: Once per day **Default state**: Active	**Parameters** None.

Name: D2JobMailing **Description**: Send mail based on context. **Default frequency**: Once per day **Default state**: Active	**Parameters** None.

Name: D2JobRemoveEmptyFolder **Description**: Remove empty folders from the repository. **Default frequency**: Once per day **Default state**: Inactive	**Parameters** • `paths`: List of paths in the repository to scan separated by a comma. • `parameters`: Specifies the `dm_folder` subtype to delete. Default setting is `dm_folder`.

Name: D2JobSubscription **Description**: Process items from d2_subscription_queue to send notifications for events. **Default frequency**: Every 5 minutes **Default state**: Active	**Parameters** None.

Name: D2JobSubscriptionSendMail **Description**: Process items from `d2_subscription_queue` to send notifications for events if notification differed is selected in the subscription component. **Default frequency**: Every 5 minutes **Default state**: Active	**Parameters** None.

Name: D2JobWFCleanerWorkflows **Description**: Clean the data of groups and aliases created dynamically by workflows when the workflow has ended. `d2_workflow_tracker` changes to the ended state. **Default frequency**: Once per day **Default state**: Active	**Parameters** None.

Name: D2JobWFCleanPopHistory **Description**: Clean email history table `d2_wf_pop` and alert administrators of possible bugs in reject and accept messages. Specific to Domino server. **Default frequency**: On demand, or once per month. **Default state**: Active	**Parameters** • `month_offset`: Sets the number of months to clean. • `full_purge`: By default, deletes all mails flagged as a potential problem. Set to `false` to prevent full purge.

Name: D2JobWFLaunchScheduledWorkflows **Description**: Scan all `d2_workflow_tracker` states for waiting and launches workflows. `d2_workflow_tracker` changes to the ongoing state. **Default frequency**: Once per day **Default state**: Active	**Parameters** None.

Name: D2JobWFReceiveTaskMail **Description**: Scan email accounts, recover messages, and extract job identification of tasks. Receive responses from external tasks by mail. **Default frequency**: Every 15 minutes **Default state**: Active	**Parameters** None.

Name: D2JobWFReportingUpdate **Description**: This job makes sure that Workflow reporting is synchronized with real time data for completed Workflows when the JMS was down while Workflows were completed. It looks for unprocessed Finished Workflow notifications, updates the `d2_workflow_tracker` state to finished, and updates the end date and supervisor name. If you are aware of an extended period of JMS outage while Workflows were completed, run this job proactively. Failure to run this job might cause slow Workflow listings for completed Workflows. **Default frequency**: Once per day **Default state**: Active	**Parameters** None.

Name: D2JobWFSendTaskMail **Description**: Scan email accounts, recover messages, and extract job identification of tasks. Send external tasks by mail. **Default frequency**: Every 15 minutes **Default state**: Active	**Parameters** None.

Name: D2JobWFWorkflowsNotifications **Description**: Send notification if workflow has not ended after the planned end date. **Default frequency**: Once per day **Default state**: Active	**Parameters** • `pause`: Sets an interval in milliseconds between each mail sent to reduce overload.

Name: D2JobCleanUpAsyncObjectsApplyVD
 	
Description: Cleans up the unused async objects created to track the async operation for apply Virtual Document template operation.
 	
Default frequency: Special, see `age`.
 	
Default state: Active
	Parameters
	
	• `age`: Specifies the threshold for the age of the `async_objects`. Any object older than this age will be deleted provided the status of it was success or failure.
	
	`<methodArguments> -age 1440 </methodArguments>`
	
	• `operation_type`: The type of the async_operation.
	
	`<methodArguments> -operation_type APPLY_VD_TEMPLATE </methodArguments>`
	
	• `abandoned_operation_age`: Specifies the age of stale async_objects that got stuck in running state due to any error in JMS or any other exception.
	
	`<methodArguments> -abandoned_operation_age 20160</methodArguments>`

Name: D2JobCoreSignStatusUpdate **Description**: Checks signature provider to get the status of pending signature requests. **Default frequency**: Configurable. **Default state**: Active	**Parameters** None.

Name: D2JobZipAndDownloadCleanup	Parameters
Description: Cleans the tracking object entries and corresponding compressed files and folders for the zip and download feature. This job collects directory information to delete based on the following conditions: • If status is `INPROGRESS` or `STARTED`, then the method checks **DATEDIFF** with **zip_process_init_time**. • If status is `COMPLETED`, then the method checks the **DATEDIFF** with **archive_end_date**. • If the status is `CANCELLED` then the method always collects the rows from DB. Once the method collects the Object IDs and directories, then the job deletes all the corresponding rows from the DB and deletes the directories. In this process this job assumes that all `DATE`s are saved in the DB in UTC format (a common format for APP server and Method Server), so that the date comparison is uniform. The zip and download init process saves the base directory location in the **directory_location** field of the tracker object. **Default frequency**: Once per day. **Default state**: Active	`-age`: Specifies the threshold for the age of the tracking object entries and corresponding files and folders. Any object older than this age will be deleted provided the other conditions are met. Default age is one day or 1440 minutes.

Name: D2BinJobBulkDeleteRestore **Description**: Bulk clean or restoration of the user's bins. User names can be provided as an argument along with the operation to be enabled. **Default frequency**: On demand. **Default state**: Inactive	**Parameters** • `-empty_bin`: Flag used to enable or disable the empty bin operation. For example: `True/False` • `-restore`: Flag used to enable or disable the restore bin operation. For example: `True/False` • `-dql_empty_bin`: A valid DQL that fetches valid users or group names. For example: `select owner_name as user_name from dm_sysobject` • `-dql_restore`: A valid DQL that fetches valid users or group names. For example: `select owner_name as user_name from dm_sysobject`. • `-name_restore`: A valid user or group name. • `-name_empty_bin`: A valid user or group name.

Name: D2BinJobDeleteExpiredObject **Description**: Cleans objects from the Recycle Bin. Runs when an object exceeds the number of days defined in **Max days in Recycle bin for user** or the `-global_max_days` passed in an argument. If you pass a `-global_max_days`, it overrides the defined value in the configuration. **Default frequency**: Once per day. **Default state**: Active	**Parameters** `-global_max_days`: Number of days a object can exist in the recycle bin. For example: `-global_max_days 10`

OpenText OpenText Documentum CM Administrator User Guide provides details about creating and using jobs.

32.2 List of PDF Configuration Jobs

Name: C2JobRendition **Description**: Create a static rendition if a PDF configuration method does not create the file. Launching the job adds a static rendition request to the queue. **Frequency**: Once per hour **Default state**: Active	**Parameters** None.

Chapter 33

Widget Communication Channel Reference

33.1 List of Widget Communications Channels

Widgets use communication channels to send messages. The messages contain strings of information that dictate interaction between widgets and repositories. There are two types of communication channels:

- *Event Channels*: These channels are used to communicate about an event that has already occurred. Several widgets can send a message in the same channel. Sometimes the content of the message can be slightly different, with few or more parameters. Every message has data filled in its built-in parameters.
- *Action Channels*: These channels are used to request certain actions from the portal. Several widgets can send a message to the same channel and sometimes the content of the message can be slightly different, with few or more parameters. Almost every message has data filled in its built-in parameters.

Select a communication channel when creating and configuring external widgets as described in the following table:

Channel name	Event it communicates	Parameters
D2_EVENT_CUSTOM	Uses an external widget. The event communicates between widgets.	Parameters depend on the custom event and should be added to allow subscribers to identify specific events.
D2_EVENT_DM_TICKET_GENERATED	A message has been sent using `D2_ACTION_DM_TICKET_GENERATE`.	`ticket`: ticket. `ticket_timeout`: ticket timeout in milliseconds. `ticket_time_generated`: Time of ticket generation in posix time. `ticket_time_expiration`: Time of ticket expiration in posix time.
D2_EVENT_SELECT_OBJECT	An object (folder or document) has been selected.	`parentId`: ID of the parent object. `parentType`: Type of the parent object.
D2_EVENT_SELECT_OBJECTS	Multiple objects have been selected.	`oam_id`: List of object IDs separated by `SEPARATOR_VALUE`.

Channel name	Event it communicates	Parameters
D2_EVENT_LOCATED_OBJECT	An object (folder or document) has been located.	`oam_id`: ID of the object. `locationType`: Location type, such as the widget type in which the content is shown. `locateId`: ID of the object to locate. `locateName`: Name of the object to locate. `version`: Version of the object to locate. `chronicleId`: ChronicleID of the object to locate. `collapseId`: List of IDs of the folders to reload.
D2_EVENT_LOCATED_SEARCH	A search has been located.	`oam_id`: ID of the object.
D2_EVENT_LOCATED_TASK	A task has been located.	`oam_id`: ID of the object.
D2_EVENT_OBJECT_DESTROYED	An object has been destroyed. It can be a folder, a document, a relation, and so on.	`oam_id`: List of object IDs separated by `SEPARATOR_VALUE`.
D2_EVENT_OBJECT_CREATED	An object has been created. It can be a folder, a document, and so on.	The channel only applies to the Doclist widget. `oam_id`: ID of the folder. `pathId`: Path ID of the object. For example, `/reponame/0c0d04e580000105/0b0d04e580055dd8`
D2_EVENT_COMMENT_ADDED	Creates a comment.	The channel only applies to the Comments widget. `comment`: ID of the comment.
D2_EVENT_FILTER_CHANGED	The global filter of the workspace has been changed.	
D2_EVENT_CHECKOUT_STATE_CHANGED	Changes content checkout state.	
D2_EVENT_PREFERENCES_CHANGED	The preferences of the user have been changed.	

Channel name	Event it communicates	Parameters
D2_EVENT_PREFERENCES_LOADED	The preferences of the user have been loaded.	
D2_EVENT_WIDGET_GET_FOCUS	A widget gets the focus.	
D2_EVENT_WIDGET_FOCUS	A widget has been set to active.	`config`: Widget configuration name. `widgetType`: set the value to the widget type. If you want to set a specific instance of a widget, use `config` and set the value to the widget name.
D2_EVENT_WIDGET_DISPLAYED	A widget is displayed.	`config`: Widget configuration name. `widgetType`: Widget type of the event source.
D2_EVENT_WIDGET_HIDDEN	A widget is hidden. The event also gets published when an external widget is closed and a parameter by name external_widget_closed (whose value is set to true) is made available through the event message. This allows external widget developers to distinguish between a widget going out of the view and a widget closing.	`config`: Widget configuration name. `widgetType`: Widget type of the event source.
D2_EVENT_SEARCH_RUN	A search has been run.	The channel only applies to the Searches widget and the Doclist widget. `oam_id`: ID of the search.. `searched_type`: Search type.
D2_EVENT_SEARCH_SAVED	A search has been saved.	The channel only applies to the Searches widget. `oam_id`: ID of the search.. `is_public`: Set to `true` if the search is public or shared.
D2_EVENT_SEARCH_CATEGORY_CREATED	A search category has been created.	The channel only applies to the Searches widget.

Channel name	Event it communicates	Parameters
D2_EVENT_WORKFLOW_ABORTED	A workflow has been aborted.	The channel only applies to the Searches widget. `oam_id`: List of object IDs separated by `SEPARATOR_VALUE`. `widgetType`: set the value to the widget type associated with the event.
D2_EVENT_SELECT_TASK	A task has been selected.	`oam_id`: ID of the task.
D2_EVENT_SELECT_TASK_FOLDER	A task folder has been selected in the task manager widget.	The channel activates task details, task attachments, and task notes. `oam_id`: ID of the task folder.
D2_EVENT_SELECT_TASK_NOTE	A task note has been selected.	The channel only applies when communicating from the Task browser widget to the Task list widget. `oam_id`: ID of the task note.

D2_EVENT_SELECT_TASK_PERFORMER	A task performer has been selected in the task performers widget.	`oam_id`: ID of the task performed.
		
		The following are copied from the selected item:
		
		`icon`
		
		`id`
		
		`locked`
		
		`style`
		
		`thumbnail`
		
		`version`
		
		`immutable`: Set to `true` if the item is immutable.

D2_EVENT_TASK_READ_STATE_CHANGED	The read state of a task changed.	`oam_id`: ID of the task. `locateId`: ID of the task to locate. `parentId`: ID of the parent task.

Channel name	Event it communicates	Parameters
D2_EVENT_TASK_PRIORITY_CHANGED	The priority of a task changed.	`oam_id`: ID of the task. `parentId`: ID of the parent task.
D2_EVENT_TASK_PROCESSED	A task has been forwarded or rejected.	`oam_id`: ID of the task.
D2_EVENT_WIDGET_INITIALIZED	A widget has been initialized.	
D2_EVENT_WORKSPACE_CLOSED	A workspace has been closed.	`oam_cuid`: Uid of the workspace tab that was closed.

Channel name	Event it communicates	Parameters

D2_EVENT_THEME_CHANGED	The theme has been changed. The theme data is included within the message.	`oam_cuid`: Uid of the container.
		
		The following are copied from the theme:
		
		`background_color`
		
		`background_color_type`
		
		`background_color_gradient_end`
		
		`background_color_gradient_start`
		
		`background_color_texture_name`
		
		`header_color`
		
		`label`: Theme label.
		
		`name`: Theme name.
		
		`selection_blur_color`
		
		`selection_color`
		
		`border_color`
		
		`title_color`
		
		`tab_active_color`
		
		`tab_border_color`
		
		`tab_inactive_color`
		
		`dialog_background_white`: OpenText Documentum CM sets this to `true` if the dialog background is white.
		
		`tab_colored_text`: Boolean for whether the tabs use colored text.

D2_EVENT_RENDITION_ADDED	A rendition has been added to a document.	The channel only applies to the Renditions widget. `oam_id`: ID of the content.

Channel name	Event it communicates	Parameters
D2_EVENT_IFRAME_ACTIVE	An iframe has been activated.	`oam_value`: Boolean to indiciate if the iframe is activated or deactivated.
D2_EVENT_IFRAME_INIT	An iframe has been opened.	`channels`: List of events. `D2_EVENT_THEME_CHANGED` contains information about the iframe theme.
D2_EVENT_SELECT_TAXONOMY	A taxonomy has been selected.	The channel only applies to the Properties widget. `oam_id`: ID of the taxonomy. The event copies the taxonomy item attributes.
D2_EVENT_SELECT_DICTIONARY	A dictionary has been selected.	The channel only applies to the Properties widget. `oam_id`: ID of the dictionary. The event copies the dictionary item attributes.
D2_EVENT_SELECT_USER	A user has been selected.	The channel only applies to the Properties widget. `oam_id`: ID of the user. The event copies the user item attributes.
D2_EVENT_USER_CREATED	A user has been created.	The channel only applies to the Users widget. `oam_id`: ID of the user.
D2_EVENT_SELECT_GROUP	A group has been selected.	The channel only applies to the Properties widget. `oam_id`: ID of the group. The event copies the group item attributes.
D2_EVENT_GROUP_CREATED	A group has been selected.	The channel only applies to the Groups widget. `oam_id`: ID of the group.

Channel name	Event it communicates	Parameters
D2_EVENT_SELECT_RELATION	A relation has been selected.	The channel only applies to the Relations widget. `oam_id`: ID of the relation. The event copies the relation item attributes.
D2_EVENT_SELECT_RENDITION	A rendition has been selected.	The channel only applies to the Renditions widget. `oam_id`: ID of the rendition. The event copies the rendition item attributes.
D2_EVENT_SELECT_RETENTION	A retention policy has been selected.	The channel only applies to the Retentions widget. `oam_id`: ID of the retention policy. The event copies the retention policy item attributes.
D2_EVENT_SELECT_MARKUP	A markup policy has been selected.	`oam_id`: ID of the markup policy. The event copies the markup policy item attributes.
D2_EVENT_SELECT_SEARCH	A search has been selected.	The channel only applies to the Doclist widget. `oam_id`: ID of the search. The event copies the search item attributes.
D2_EVENT_SELECT_FACET	A facet has been selected.	The channel only applies to the Doclist widget. `oam_id`: ID of the facet. The event copies the facet item attributes.
D2_EVENT_SELECT_QUERYFORM	A query form has been selected.	The channel only applies to the Doclist widget. `oam_id`: ID of the query form search. The event copies the query form search item attributes.

Channel name	Event it communicates	Parameters
D2_EVENT_SELECT_DISTRIBUTION	A distribution configuration has been selected.	`oam_id`: ID of the distribution. The event copies the distribution item attributes.
D2_EVENT_SELECT_WORKFLOWTRACKER	A workflow tracker has been selected.	`oam_id`: ID of the workflow overview. The event copies the workflow overview item attributes.
D2_EVENT_SELECT_WORKFLOW_ACTIVITY	A workflow activity has been selected.	`oam_id`: ID of the workflow activity. The event copies the workflow activity item attributes.
D2_EVENT_SELECT_AUDIT	An audit event has been selected.	`oam_id`: ID of the audit. The event copies the audit item attributes.
D2_EVENT_SELECT_SNAPSHOT	A virtual document snapshot has been selected.	The channel only applies to the Snapshots widget. `oam_id`: ID of the virtual document snapshot. The event copies the virtual document snapshot item attributes.
D2_EVENT_VD_TEMPLATE_SELECTED	A virtual document template form has been selected.	The channel only applies to the Snapshots widget. `oam_id`: ID of the virtual document template. The event copies the virtual document template item attributes.
D2_EVENT_DRAG_DROP_STARTED	A drag and drop event started.	`oam_id`: List of content IDs.
D2_EVENT_DRAG_DROP_ENDED	A drag and drop event stopped.	`oam_id`: List of content IDs.

Channel name	Event it communicates	Parameters
D2_EVENT_IMPORTED_FROM_URL	Content from a URL has been imported.	`oam_id`: View position (0 if none). `cancelReason`: Reason for cancellation if the operation is canceled. `result`: Contains the result if the operation is completed. `error`: Contains the error if the operation fails.
D2_EVENT_EXPORTED_FROM_URL	Content from a URL has been exported.	`oam_id`: View position (0 if none). `cancelReason`: Reason for cancellation if the operation is canceled. `result`: Contains the result if the operation is completed. `error`: Contains the error if the operation fails.
D2_EVENT_DOWNLOAD_URLS_RECEIVED	The `D2_ACTION_DOWNLOAD_URLS_REQUEST` has been published.	`urls`: Contains a concatenated string of URLs.

Send messages through channels to provide information about an action to be performed on content as described in the following table:

Note: The CLIPBOARD_SERVLET_VIEW actions listed below require the user to copy URLs from a pop-up dialog. Non-SERVLET actions do not feature a pop-up.

Channel name	Request	Parameters
D2_ACTION_OPEN_FOLDER	Open a folder in the portal.	The channel only applies to the Doclist widget. `oam_id`: ID of the folder to open. `collapseId`: List of IDs of the folders to reload. `pathId`: Path ID of the object. For example, `/reponame/0c0d04e580000105/0b0d04e580055dd8`

Channel name	Request	Parameters
D2_ACTION_OPEN_URL	Open a URL in a new browser window.	`url`: the URL to open.
D2_ACTION_OPEN_VD	Open a virtual document in the portal.	The channel only applies to the Virtual Documens widget. `oam_id`: ID of the virtual document to open.
D2_ACTION_OPEN_SNAPSHOT	Open a virtual document snapshot.	The channel only applies to the Snapshots widget.
D2_ACTION_LOCATE_OBJECT	Locate an object in the portal.	The channel only applies to the Locations widget. `oam_id`: ID of the object to locate. `openFolderOnLocate`: Boolean value to open folder during locate, if oam_id is a folder ID.
D2_ACTION_LOCATE_TASK	Locate a task in the portal.	The channel only applies to the Locations widget. `oam_id`: ID of the task to locate.
D2_ACTION_LOCATE_SEARCH	Locate a saved search in the portal.	The channel only applies to the Locations widget. `oam_id`: ID of the task to locate.
D2_ACTION_REFRESH_DOCUMENT	Refresh the widget where the document is displayed.	`oam_id`: ID of the document. `import`: Present if the refresh is related to an import.
D2_ACTION_COPY_LINK_IN_CLIPBOARD	Copy the smartlink to the object in the clipboard.	`oam_id`: ID of the object to locate.
D2_ACTION_COPY_LINK_IN_CLIPBOARD_LOCATE	Copy the smartlink to the object in the clipboard.	`oam_id`: ID of the object to locate.
D2_ACTION_COPY_LINK_IN_CLIPBOARD_VIEW	View the document.	`oam_id`: ID of the object to locate.
D2_ACTION_COPY_LINK_IN_CLIPBOARD_VIEW_NATIVE	View the document without renditions or metadata.	`oam_id`: ID of the object to locate.
D2_ACTION_COPY_LINK_IN_CLIPBOARD_VIEW_WITH_NATIVE_ANNOTATION	View the document with native annotations.	`oam_id`: ID of the object to locate.

Channel name	Request	Parameters
D2_ACTION_COPY_LINK_IN_CLIPBOARD_SERVLET_VIEW	View the document in the servlet view. **Note:** The servlet view opens the file and performs the action without login and without having to go through the user interface.	`oam_id`: ID of the object to locate.
D2_ACTION_COPY_LINK_IN_CLIPBOARD_SERVLET_VIEW_NATIVE	View the document in the servlet view without renditions or metadata.	`oam_id`: ID of the object to locate.
D2_ACTION_COPY_LINK_IN_CLIPBOARD_SERVLET_VIEW_CURRENT	View the current version of the document in the servlet view.	`oam_id`: ID of the object to locate.
D2_ACTION_COPY_LINK_IN_CLIPBOARD_SERVLET_VIEW_NATIVE_CURRENT	View the current version of the document with native content in the servlet view.	`oam_id`: ID of the object to locate.
D2_ACTION_DUMP	Display a dump of the object.	`oam_id`: ID of the object to dump.
D2_ACTION_CONTENT_CHECKIN	Check in a document.	`oam_id`: ID of the document to check in.
D2_ACTION_CONTENT_IMPORT_AS_VERSION	Import a new version of a document.	`oam_id`: ID of the document.
D2_ACTION_CONTENT_CHECKIN_NATIVE_ANNOTATION	Check in the native annotation of a document.	`oam_id`: ID of the document.
D2_ACTION_CONTENT_PRINT	Print a document.	`oam_id`: ID of the document.
D2_ACTION_CONTENT_COMPARE	Compare two documents.	`oam_id`: IDs of the documents to compare.
D2_ACTION_CONTENT_EXPORT	Export a document.	`oam_id`: ID of the document to export. `DOWNLOAD_LOCATION`: Location for download.
D2_ACTION_CONTENT_EXPORT_NATIVE	Export a document without including renditions or metadata.	`oam_id`: ID of the document. `DOWNLOAD_LOCATION`: Location for download.
D2_ACTION_CONTENT_EXPORT_FOLDER	Export a folder.	`oam_id`: ID of the folder. `DOWNLOAD_LOCATION`: Location for download.

Channel name	Request	Parameters
D2_ACTION_CONTENT_EXPORT_FOLDER_NATIVE	Export a folder without including renditions or metadata.	`oam_id`: ID of the folder. `DOWNLOAD_LOCATION`: Location for download.
D2_ACTION_CONTENT_VIEW	View a document.	`oam_id`: ID of the document to view. `DOWNLOAD_LOCATION`: Location for download.
D2_ACTION_CONTENT_VIEW_NATIVE	View a document without renditions or metadata.	`oam_id`: ID of the document to view. `DOWNLOAD_LOCATION`: Location for download. **Note:** : The minimum permission required for D2_ACTION_CONTENT_VIEW_NATIVE to work is read.
D2_ACTION_CONTENT_EDIT	Edit a document.	`oam_id`: ID of the document. `DOWNLOAD_LOCATION`: Location for download.
D2_ACTION_CONTENT_CHECKOUT	Check out a document.	The channel only applies to the Checkout widget. `oam_id`: ID of the document to check out. `DOWNLOAD_LOCATION`: Location for download.
D2_ACTION_CONTENT_CANCEL_CHECKOUT	Cancel the check out of a document.	`oam_id`: ID of the document. `locateAndRefresh`: Boolean to determine if the location and refresh occurred. `refreshCheckoutState`: Boolean to determine if the checkout state is refreshed.
D2_ACTION_CONTENT_EDIT_WITH_NATIVE_ANNOTATION	Edit a document with native annotations.	`oam_id`: ID of the document.
D2_ACTION_CONTENT_VIEW_WITH_NATIVE_ANNOTATION	View a document with native annotations.	`oam_id`: ID of the document.

Channel name	Request	Parameters
D2_ACTION_CONTENT_EDIT_NATIVE_ANNOTATION	Edit the native annotations of a document.	`oam_id`: ID of the document.
D2_ACTION_COMMENT	Show the comment creation dialog box. Set `parentCommentId` as the parent comment.	The channel only applies to the Comments widget. `oam_id`: ID of the document. `parentCommendId`: ID of the parent comment. `id`: ID of the comment. `editor`: Richtext body of the comment.
D2_ACTION_CONTENT_CREATE	Show the content creation dialog box. Set `parentId` as the parent folder as the content location if no autolink is set.	`oam_id`: ID of the base document used for inheritance. `parentId`: ID of the parent folder. The system creates content inside the specified folder if no autolink is set.
D2_ACTION_CONTENT_IMPORT	Show the content import dialog box. Set `parentId` as the parent folder. This folder is used to create the document inside it if no autolink is set.	`oam_id`: ID of the base document used for properties inheritance. `parentId`: ID of the parent folder. The system creates content inside the specified folder if no autolink is set. `files`: list of files on the client machine.
D2_ACTION_CONTENT_EXPORT_C2_OVERRIDE	Allow users who would normally download a configured PDF of a document due to read-only access restrictions to override that restriction so that they can download native content.	`oam_id`: ID of the content.
D2_ACTION_CONTENT_VIEW_C2_OVERRIDE	Allow users who would normally view a configured PDF controlled version of a document due to read-only access restrictions to override that restriction so that they can view native content.	`oam_id`: ID of the content.

Channel name	Request	Parameters
D2_ACTION_FOLDER_CREATE	Show the folder creation dialog box. Set the `root_object_type` to `dm_folder` or `dm_cabinet`.	`oam_id`: ID of the folder in which the new folder is being created. `root_object`: set to `dm_folder` or `dm_cabinet` to specify the folder type.
D2_ACTION_OBJECT_DESTROY	Show the content destruction dialog box. Set `parentId` as the id of the parent of the object.	`oam_id`: ID of the object to destroy.
D2_ACTION_PERMISSIONS_SHOW	Show the permissions dialog box.	`oam_id`: ID of the object.
D2_ACTION_RENDITION_ADD	Show the add rendition dialog box.	`oam_id`: ID of the content.
D2_ACTION_RENDITION_REQUEST	Request a rendition.	`oam_id`: ID of the content.
D2_ACTION_RELATION_CREATE	Show the relationship creation dialog box.	`oam_id`: ID of the content.
D2_ACTION_RELATION_DESTROY	Show the relationship destruction dialog box. Set `associate_id` as the ID of the relation to destroy.	`oam_id`: ID of the content. `associate_id`: ID of the relation.
D2_ACTION_MASS_UPDATE	Run a mass update configuration. Set `config_name` as the name of the mass update configuration.	`oam_id`: ID of the content. `config_name`: name of the mass update configuration.
D2_ACTION_SYNCPLICITY	Show the Syncplicity dialog box.	`oam_id`: ID of the content.
D2_ACTION_CUT	Perform a cut operation.	`oam_id`: ID of the content. `parentId`: ID of the parent content.
D2_ACTION_COPY	Perform a copy operation.	`oam_id`: ID of the content. `parentId`: ID of the parent content.
D2_ACTION_PASTE	Perform a paste operation.	`parentId`: ID of the parent content.
D2_ACTION_PASTE_LINK	Paste a link of the clipboard content.	`parentId`: ID of the parent content.

Channel name	Request	Parameters
D2_ACTION_CLIPBOARD_GET	Retrieve the clipboard content. This action triggers `D2_ACTION_CLIPBOARD_CONTENT`.	
D2_ACTION_CLIPBOARD_CONTENT	Perform an operation on the clipboard content.	`clipboard-operation`: Operation performed from the clipboard. `clipboard-contentId`: ID of the clipboard content. `clipboard-value`: Value of the clipboard content.
D2_ACTION_DOWNLOAD_URLS_REQUEST	Request a list of document URLs for content.	`oam_id`: ID of the content. `format`: Format of the content. `pageModifier`: Page modifier. `pagenumber`: Page number.
D2_ACTION_IMPORT_FROM_URL	Import content from a URL.	`oam_id`: ID of the content. `url`: URL of the content. `isModel`: Set to `true` to make the content modal.
D2_ACTION_EXPORT_FROM_URL	Export content from a URL.	`oam_id`: ID of the content. `url`: URL of the export location.. `useViewLocation`: Set to `true` to use the view location, otherwise the export opens the application selection dialog box. `hasDataPost`: Boolean to indicate whether the export operation has information in the `dataPost` parameter. `dataPost`: The values to post.

Channel name	Request	Parameters
D2_ACTION_SENDMAIL	Send mail.	`oam_id`: ID to send. You do not need to use `oam_id` if you used `id`. `id`: ID to send. `type`: Object type. `parentType`: Parent content type. Optional.
D2_ACTION_FAVORITE_ADD	Add content to the list of favorites.	`oam_id`: ID of the content.
D2_ACTION_FAVORITE_REMOVE	Remove content from the list of favorites.	`oam_id`: ID of the content.
D2_ACTION_ACROBAT_ANNOTATION_OPEN	Open an Adobe Acrobat annotation.	`oam_id`: ID of the content.
D2_ACTION_DM_TICKET_GENERATE	Generate a ticket.	
D2_ACTION_SHARE_TO_CORE	Share selected content to OpenText Core Share.	Smart View action
D2_ACTION_SHARE_TO_CORE_C2_OVERRIDE	Share selected content to OpenText Core Share and overrides PDF Configuration rules such as watermark.	Smart View action. This action will not function unless PDF Configuration override settings in D2FS.properties are also enabled.

Send messages through channels to perform interface actions as described in the following table:

Channel name	Request	Parameters and notes for focus events
D2_ACTION_DISPLAY_DIALOG	Show a dialog box.	`oam_id`: ID of the dialog box. `raw`: A boolean that retrieves the raw `oam_id` content when set to `true`. Optional. `DIALOG_NAME`: Dialog box name. `MANAGER`: Manager class name. Optional. `parentType`: Parent object type. Optional. `mode`: Create, edit, or import mode of the dialog box. Optional.
D2_ACTION_EXECUTE_MANAGER	Execute a manager class for a dialog box.	`oam_id`: ID of the dialog box. `raw`: A boolean that retrieves the raw `oam_id` content when set to `true`. Optional. `DIALOG_NAME`: Dialog box name. `MANAGER`: Manager class name.

Channel name	Request	Parameters and notes for focus events
D2_ACTION_EXECUTE	Execute an action.	`rId`: Returned ID. By default, this is set to the message ID. `rType`: Returned type (JS, NATIVE, or EVENT). By default, this does nothing. `rAction`: Returned action (inline Javascript, native function name, or event name). By default, this does nothing. `eMode`: Entry mode (set as SINGLE to call the action for one ID, MONO to call the action once for each ID, or MULTI to call the action once with a list of all provided IDs). By default, this is set to MULTI. `eCall`: Entry call for when eMode is set to `MONO` (set as SEQUENTIAL to call the actions one at a time by waiting for each service call to return or PARALLEL to call the actions at the same time). By default, this is set to SEQUENTIAL. `eMethod`: Name of the entry method. This is only required for D2FS services in conjunction with the eService parameter. `eService`: Name of the plug-in interface service without the `I` prefix and `Service` suffix.
D2_ACTION_REFRESH_WIDGET	Refresh the widget.	`oam_id`: ID of the widget. `oam_target_type`: Widget type.
D2_ACTION_RELOAD_WORKSPACE	Refreshe the active workspace.	`oam_id`: ID of the workspace. `option_same_repository`: Determines if the workspace is in the same repository.

Channel name	Request	Parameters and notes for focus events
D2_ACTION_WIDGET_FULLSCREEN	Set the widget to fullscreen view.	`original_widget`: Original widget ID. `original_event`: Original event ID. `original_cuid`: Original cuid. `original_global`: Original global ID. `url`: URL of the widget.

Send messages through channels to provide information about an action to be performed on distributions as described in the following table:

Channel name	Request	Parameters and notes for focus events
D2_ACTION_DISTIBUTION_ACCEPT	Accept the distribution. Set `config_name` as the distribution configuration name.	`oam_id`: ID of the content. `config_name`: name of the distribution.
D2_ACTION_DISTRIBUTION_REJECT	Reject the distribution. Set `config_name` as the distribution configuration name.	`oam_id`: ID of the content. `config_name`: name of the distribution.
D2_ACTION_DISTRIBUTION_PREPARE	Prepare the distribution. Set `config_name` as the distribution configuration name.	`oam_id`: ID of the content. `config_name`: name of the distribution.
D2_ACTION_DISTRIBUTION_LAUNCH	Launch the distribution. Set `config_name` as the distribution configuration name.	`oam_id`: ID of the content. `config_name`: name of the distribution.
D2_ACTION_DISTRIBUTION_STOP	Stop the distribution. Set `config_name` as the distribution configuration name.	`oam_id`: ID of the content. `config_name`: name of the distribution.
D2_ACTION_DISTRIBUTION_REPORT	Report the distribution. Set `config_name` as the distribution configuration name.	`oam_id`: ID of the content. `config_name`: name of the distribution.

Channel name	Request	Parameters and notes for focus events
D2_ACTION_SUBSCRIBE	Subscribe to a distribution.	`oam_id`: ID of the distribution.

Send messages through channels to provide information about an action to be performed on searches as described in the following table:

Channel name	Request	Parameters and notes for focus events
D2_ACTION_SEARCH_DOCUMENT	Search the repository for content.	`oam_id`: ID of the search. `locateId`: ID of the object to search. `searched_type`: Search type name. `facet_attr_name`: Facet attribute name. `facet_attr_value`: Facet attribute value. `reset_facet_attr_name`: Reset the facet attribute name.
D2_ACTION_SEARCH_HIGHLIGHT_TERMS	Highlight search terms.	`oam_value`: Binary value for whether to highlight search terms (0 or 1).
D2_ACTION_ADVANCED_SEARCH	Show the advanced search dialog box.	`oam_id`: ID of the search. If you set this to `null`, the action prompts the user to create a new advanced search.
D2_ACTION_SEARCH_CATEGORY_DIALOG	Show the advanced search category dialog box.	
D2_ACTION_PREFERENCES_SHOW	Show the user preferences dialog box.	
D2_ACTION_QUERYFORM_SHOW	Show the query form search dialog box.	`oam_id`: ID of the query form.
D2_ACTION_QUERY_CATEGORY_DIALOG	Show the query form search category dialog box.	

Send messages through channels to provide information about an action to be performed on virtual documents as described in the following table:

Channel name	Request	Parameters and notes for focus events
D2_ACTION_CONVERT_VD	Convert a document to a virtual document.	`oam_id`: ID of the content.
D2_ACTION_CONVERT_DOC	Convert a virtual document to a simple document.	`oam_id`: ID of the virtual document.
D2_ACTION_SNAPSHOT_CREATE	Create a virtual document snapshot.	The channel applies only to the Snapshots widget. `oam_id`: ID of the virtual document.
D2_ACTION_VD_ADD_CHILD_SELECTED	Show the add child dialog box for the currently selected item in the VD widget.	
D2_ACTION_VD_REMOVED_CHILD_SELECTED	Show the remove child dialog box for the currently selected item in the VD widget.	
D2_ACTION_VD_SET_CHILD_BINDING_VERSION	Show the binding version dialog box to set the binding version of the currently selected item in the VD widget.	
D2_ACTION_VD_INSERT_INHERITED_COMPONENT	Show the insert inherited component dialog box for the currently selected item in the VD widget.	
D2_ACTION_VD_CLEAR_VD	Clear a virtual document of elements.	

Send messages through channels to provide information about an action to be performed on lifecycles and workflows as described in the following table:

Channel name	Request	Parameters and notes for focus events
D2_ACTION_LIFECYCLE_DCTM	Run a lifecycle operation on a OpenText Documentum CM lifecycle.	`oam_id`: ID of the content. `operation`: the operation to perform. `policyId`: the policy to use.
D2_ACTION_LIFECYCLE_D2	Run a lifecycle operation on a OpenText Documentum CM lifecycle.	`oam_id`: ID of the content. `operation`: the operation to perform. `policyId`: the policy to use.

Channel name	Request	Parameters and notes for focus events
D2_ACTION_WORKFLOW_LAUNCH	Launch a workflow for a document.	`oam_id`: ID of the document. `config`: name of the workflow configuration.
D2_ACTION_WORKFLOW_LAUNCH_SCHEDULED	Launch a scheduled workflow for a document.	`oam_id`: ID of the `queue_item`.
D2_ACTION_WORKFLOW_UPDATE_PERFORMERS	Show the update performers dialog box.	`oam_id`: ID of the `queue_item`. `workflowTrackerId`: ID of the workflow tracker used if `oam_id` is not an `ID2cWorkfowTracker` or `IDfQueueItemworkflow`. Optional.
D2_ACTION_WORKFLOW_ABORT	Show the abort workflow dialog box.	`oam_id`: ID of the workflow tracker.
D2_ACTION_TASK_ACQUIRE	Set the task as acquired.	`oam_id`: ID of the task. `parentId`: ID of the parent task. `forceAcquire`: set to `true` for force acquisition. `forceRead`: set to `true` for force reading.
D2_ACTION_TASK_READ	Toggle the read or unread status of the task.	`oam_id`: ID of the task. `parentId`: ID of the parent task. `read`: set to `true` or `false`.
D2_ACTION_TASK_PRIORITY	Set the task priority.	`oam_id`: ID of the task. `parentId`: ID of the parent task. `priority`: set to the priority integer.
D2_ACTION_TASK_FORWARD	Show the task forward dialog box.	`oam_id`: ID of the task.
D2_ACTION_TASK_REJECT	Show the task rejection dialog box.	`oam_id`: ID of the task.

Channel name	Request	Parameters and notes for focus events
D2_ACTION_TASK_DELEGATE	Show the task delegation dialog box.	`oam_id`: ID of the task. `source`: set to `tracker` or `task`.
D2_ACTION_TASK_NOTE	Show the task note dialog box.	The channel only applies to the Task notes widget. `oam_id`: ID of the task.
D2_ACTION_TASK_PROCESS	Show the task processing dialog box. Set `operation` as the operation to perform.	`oam_id`: ID of the task. `operation`: the operation to perform.
D2_ACTION_WORKFLOW_MANAGE_ATTACHMENTS	Show the Manage Workflow Attachment dialog box for a workflow, which is used to add a new document to the running workflow.	`oam_id`: the queue item ID for the attachment. **Note:** Documents are not directly related to the queue item but to the running workflow.

Send messages through channels to provide information about an action to be performed on taxonomies and dictionaries as described in the following table:

Channel name	Request	Parameters and notes for focus events
D2_ACTION_TAXONOMY_EXPORT	Show the taxonomy export dialog box.	`oam_id`: ID of the taxonomy.
D2_ACTION_TAXONOMY_IMPORT	Show the taxonomy import dialog box.	`oam_id`: ID of the taxonomy.
D2_ACTION_TAXONOMY_CHECKIN	Show the taxonomy checkin dialog box.	`oam_id`: ID of the taxonomy.
D2_ACTION_TAXONOMY_SAVE	Show the taxonomy save dialog box.	`oam_id`: ID of the taxonomy.
D2_ACTION_TAXONOMY_PROPERTIES	Show the taxonomy properties dialog box.	`oam_id`: ID of the taxonomy.
D2_ACTION_TAXONOMY_LOCATE	Search for the taxonomy.	The channel only applies to the Locate widget. `oam_id`: ID of the taxonomy.
D2_ACTION_TAXONOMY_RESTORE	Show the taxonomy restoration dialog box.	`oam_id`: ID of the taxonomy.
D2_ACTION_DICTIONARY_EXPORT	Show the dictionary export dialog box.	`oam_id`: ID of the dictionary.

Channel name	Request	Parameters and notes for focus events
D2_ACTION_DICTIONARY _IMPORT	Show the dictionary import dialog box.	`oam_id`: ID of the dictionary.
D2_ACTION_DICTIONARY _SAVE	Show the dictionary save dialog box.	`oam_id`: ID of the dictionary.
D2_ACTION_DICTIONARY _RESTORE	Show the dictionary restoration dialog box.	`oam_id`: ID of the dictionary.
D2_ACTION_DICTIONARY _UPDATE	Show the dictionary update dialog box.	`oam_id`: ID of the dictionary.
D2_ACTION_DICTIONARY _PROPERTIES	Show the dictionary properties dialog box.	`oam_id`: ID of the dictionary.

Send messages through channels to provide information about an action to be performed on users and groups as described in the following table:

Channel name	Request	Parameters and notes for focus events
D2_ACTION_OPEN_GROU P	Open a group	`oam_id`: ID of the group.
D2_ACTION_GROUP_CRE ATE	Create a group.	The channel only applies to the Groups widget.
D2_ACTION_GROUP_PROP ERTIES	Set the properties of a group.	The channel only applies to the Properties widget. `oam_id`: ID of the group.
D2_ACTION_USER_CREAT E	Creat a user.	The channel only applies to the Users widget. `oam_id`: ID of the user.
D2_ACTION_USER_PROPE RTIES	Set the properties of a user.	The channel only applies to the Properties widget. `oam_id`: ID of the user.

Chapter 34

Reserved Words

34.1 List of Reserved Words when Using Composer

Do not use the following words when defining objects and relations in Composer, as they conflict with the property page component:

• abstract • boolean • break • byte • case • catch • char • class • const • continue • default • delete • do • double • else • export • extends • false • final • finally • float • for • function • goto • if • implements • import • in	• instanceof • int • long • native • new • null • package • private • protected • public • return • short • static • super • switch • synchronized • this • throw • throws • transient • true • try • typeof • var • void • while • with

Chapter 35

PDF Fields Reference

35.1 List of PDF Fields

PDFs can be created with PDF configurations to add dynamic field targets so that they can be populated with data from OpenText Documentum CM. See “Configuring View Configuration” on page 353 for more information on PDF Configuration.

Field	Description
#page	Current page
#page_count	Number of pages
#chapter	Chapter name
#datetime	Date and time of the client host **Note:** The PDF fields containing the client date (keyword #datetime) do not use the date format set in the PDF fields, and instead follow the simple format `d MMMMM yyyy h:m:s a`
#username	Current user
#controled_username	Name of the user printing the content
#controled_username_count	Number of controlled prints by the current user
#controled_count	Total number of controlled prints
#reason	Reason for printing
#folder_path	First path of the document
#complete_name	First path of the document and name
#server_datetime	Server date and time
#docbase	Repository name
#version	Version number of the content
	Volume auto-link
<Property name> (for example, object_name)	Single-value attribute
<Property name>.<dictionary name>.<language or alias name> (for example, object_name.dico1.fr)	Dictionary mapping
<Property name> (for example, keywords)	Comma-separated, repeating-value attribute

Field	Description
<*Property name*>[index] (for example, keywords[0])	Repeating-value attribute
<*Property name*>.<*dictionary name*>.<*language or alias name*>[index] (for example, keyword[0].dico1.fr)	Dictionary mapping
<*ID*>.<*Property name*> (for example, dqlkey.object_name)	All recorded comma-separated values
<*ID*>.<*Property name*>[index] (for example, dqlkey.object_name[0])	Recorded value
<*ID*>.<*Property name*>.<*dictionary name*>.<*language or alias name*>[index] (for example, keyword[0].dico1.fr)	Dictionary mapping

Chapter 36

Smart View Support Reference

Consult the following tables to understand specific feature and configuration differences between Classic and Smart View. All items should be considered supported in Classic unless noted. For general support information, refer to "Understanding the Smart View client" on page 237 and "Support for Classic features in Smart View" on page 237.

36.1 Smart View: Go to... Configuration Support

The following table highlights the current Smart View client support for functionality configured through the **Go to...** menu in client configuration.

Configuration	Smart View Client Support	Notes
Auto naming	Supported	
Property Page	Supported	See Property page table for detailed information on Smart View handling of property page layout and attributes.
Inheritance	Supported	
Security	Supported	
Auto link	Supported	
Template list	Supported	
Virtual Document Template	Supported	
Folder Structure Import	Supported	Supports convert to Virtual Document option.
Folder Structure conversion	Supported	Supports convert to Virtual Document option.
Linked Document	Supported	
Check-In	Supported	
Check out	Supported	
Extended Creation Profile	Supported	
Lifecycle	Supported	
Lifecycle Batch	Supported	
Workflow	Supported	
Menu Smart View	Supported	

Configuration	Smart View Client Support	Notes
User Settings	Partial Support-see notes	User Settings supports the enablement of Client Manager, the setting of Checkout and Temp paths, Accessibility toggle, Sort Folder/Files toggle, and Language for the UI.
Filter	Supported	
Search	Supported	
Query Form	Supported	
Mailing List	Supported with Lifecycles	
Send Mail	Supported - see notes	When a user sends mail, Smart View will use the registered default mail client on the users system.
Subscription	Not Supported	
Distribution	Not Supported	
Uniqueness Check	Supported	
Mass update	Supported	
Audit	Supported	
Rendition Server	Supported	
BOCS Cache	Not Supported	
Plugin	Supported	The Client Manager (WSCTF) can be configured for use.

36.2 Smart View: Search Behavior Differences

The following chart outlines the differences that might be encountered when running various types of searches:

Type of Search	Classic client	Smart View client
In Search Results	• Date grouping by months • Simple facet constraints • First level facets in Classic	• Relative date grouping • Complex facet constraints involving multiple facet attributes • Hierarchical facets
From Last Saved Search	Retrieving last save search that used facets returns the original search results with selected facets retained.	Retrieving last save search that used facets returns the results of the last faceted search but facets are reset.

Type of Search	Classic client	Smart View client
Facets in doclist widgets	Not supported	Supported
Query Form searches	Supported	Supported

36.3 Smart View: Main Menu Support

The following tables highlight the current Smart View support for functionality configured through client configuration **Main** menu.

The following settings are available under the **Data > Dictionary** menu:

Data > Dictionary Setting	Smart View Client Support
Groups, Admin Group	Supported
Groups, Search Dictionary…	Supported
Labels	Supported
Manual sort	Supported
Create a registered table	Supported
DQL dictionary	Supported
Key	Supported
Language	Supported
Alias	Supported

The following settings are available under the **Data > Taxonomy** menu:

Data > Taxonomy Setting	Smart View Client Support
Support for using taxonomies during creation flows	Supported
Support for property page default taxonomy selection	Supported

The following settings are available under the **Data > Registered Tables** menu:

Data > Registered Tables Setting	Smart View Client Support
Support for using registered tables	Supported

The following settings are available under the **Creation > Creation Profile** menu:

Creation > Creation Profile Setting	Smart View Client Support
Available For	Supported
Folder properties used for creation	Supported
Users group	Supported

Creation > Creation Profile Setting	Smart View Client Support
Skip edit content step	Supported
Hide inheritance tab	Supported
Default options	Supported
Dictionary	Supported
Properties	Supported
Type	Supported
Transfer Config	Supported
Property Pages	Supported
Version	Supported
Inheritance	Supported
Default values template	Supported
Lifecycle, single document	Supported
Lifecycle, multi-document	Supported
Workflow	Supported

The following settings are available under the **Creation > Default values template** menu:

Creation > Default values template Setting	Smart View Client Support
Supports using default values during creation	Supported

The following settings are available under the **Creation > Mail Attachments** menu:

Creation > Mail Attachments Setting	Smart View Client Support
Import attachments	Supported
Keep initial mail format	Supported
Make a rendition	Supported
Excluded attachment document format	Supported

The following settings are available under the **Interface > Display Configuration** menu:

Interface > Display Configuration	Smart View Client Support
Document Type/Selected Type	Supported
Properties, Dictionary, Alias/Locale	Supported

The following settings are available under the **Interface > Search** menu:

Interface > Search Configuration	Smart View Client Support
Dictionary for type labels	Supported
Alias/Locale	Supported
Groups allowed to create public searches	Supported
Maximum results returned by search	Supported
Enable full text search	Supported
Quick search on all versions	Supported
Use Documentum CM Search / xPlore Search for all searches	Supported
Enable facets	Supported **Note:** Facets can be used in search results and in doclist widget.
Maximum number of results by facet	Partially Supported **Note:** Defaults to 5 results, then 25 'more' loaded.
Limit Documentum CM Search / xPlore term highlighting	Supported

The following settings are available under the **Transfer Configuration > Transfer Configuration** menu:

Transfer Configuration > Transfer Configuration	Smart View Client Support
Format of date property type	Not Supported
Trigger events	Not Supported
Properties Transfer	Supported
Query results in Office	Not Supported

The following settings are available under the **PDF Configuration > View/Print/Export/Rendition** menus:

PDF Configuration > View/Print/Export/Rendition Configurations	Smart View Client Support
Support for PDF Configuration View configs	Supported
Support for PDF Configuration Print configs	Supported
Support for PDF Configuration Export configs	Supported
Support for PDF Configuration Rendition configs	Supported

The following settings are available under the **Recycke Bin > Options** menus:

Recycle Bin > Options Configurations	Smart View Client Support
Support for Recycle Bin configs	Supported

The following settings are available under the **Tools** menus:

Tools Configurations	Smart View Client Support
Support for email configuration	Supported **Note:** Emails are sent by user's native email client.
Support for relation configuration	Supported
Support for using Module descriptions in Specifications Report	Not Supported
Support for Options	Supported

The following settings are available under the **Specifications > Generate PDF.../ Compare Configs** menus:

Specifications Configurations	Smart View Client Support
Support for PDF Specifications	Supported
Support for PDF Specifications Compare	Supported
Support for using Module descriptions in Specifications Report	Not Supported
Support for Options	Supported

36.4 Smart View: Menu Config Actions and Events Support

The following table describes Classic menu configuration actions and events support in Smart View.

Classic Menu Action	Smart View Menu Action	Can Be Used In
Add version action : $value(action)	D2_ACTION_CONTENT_IMPORT_AS_VERSION	Action Toolbar **Add Version** consolidates the Check-In action.
Acquire Task : force acquire $value(forceAcquire) and force read $value(forceRead)	Available	Action Toolbar

Classic Menu Action	Smart View Menu Action	Can Be Used In
Calling service's method : $value(eService). $value(eMethod)() on selection : $value(eMode), Type: $value(rType) doing Action: $value(rAction)	Not Available	
Check-in action : $value(action)	No, D2_ACTION_CONTENT_CHECKIN	Check-in action is Consolidated into one Add Version action. Single select only.
Checkout action : $value(action)	D2_ACTION_CANCEL_CHECKOUT	Action Toolbar. Users can also click the lock icon to cancel checkout.
com.emc.d2fs.dctm.ui.dynamicactions.workflows.ManageWorkflowAttachments	Available	Action Toolbar, as **Manage Workflow Attachments**
Compare two contents or two versions selected	Not Available	
Merge multiple files	D2_ACTION_DOCUMENT_MERGE	Action toolbar Only supported with Brava! Enterprise Viewer.
Convert folder to VD action : $value(action) (using Configuration $value(config_name))	Yes, D2_ACTION_FOLDERTOVD_CONVERT	Action toolbar
Lifecycle change state operation : $value(operation) using the target state $value(targetState)	Available	Action toolbar
Dictionary administration action : $value(action)	Not Available	
Distribution action : $value(action) (using Configuration $value(config_name) for prepareDistribution only)	Not Available	
Lifecycle change state to : $value(operation) (using Policy Id $value(policyId) for attach only)	Available	

Classic Menu Action	Smart View Menu Action	Can Be Used In
Export selected content(s)	Yes, D2_ACTION_CONTENT_EXPORT No, D2_ACTION_CONTENT_EXPORT_NATIVE	Action Toolbar Versions Details
Export selected folder and its content: $value(action)	Not Available	
Hyperlink PDF action : $value(action)	Not Available	
Mass update action : $value(action) (using Configuration $value(config_name))	Not Available	
Native annotation action : $value(action)	Not Available	
Print selected content(s)	Single-select print only	Action Toolbar
Publish event : $value(event)	A limited list of events is available. See next table.	Action Toolbar – see Events
Publish event : $value(event)event)	Not Available	
Publish event for reference : $value(event)ence : $value(event)	Not Available	
Relation action : $value(action)	Yes, D2_ACTION_RELATION_CREATE	Relations Details
Rendition action : $value(action)action)	Yes, D2_ACTION_REMOVE_RENDITION	Renditions Details
Import Rendition	Yes, D2_ACTION_RENDITION_ADD	
Set priority Task to : $value(priority)	Available	
Set Task to Read : $value(read)	Available	
Show dialog : $value(dialog)	PropertiesDialog, {Custom Plugin Dialog}	Action Toolbar
Smart View	Equivalent is 'Classic View' action	

Classic Menu Action	Smart View Menu Action	Can Be Used In
Show dialog with manager : $value(manager) : $value(manager)	Yes;CommentManager	Action Toolbar
Show dialog with manager : $value(manager) : $value(manager)	Yes;SendmailManager	Action Toolbar
Show dialog with manager : $value(manager) : $value(manager)	Yes;DeleteManager	Action Toolbar
Show dialog with manager : $value(manager) : $value(manager)	Yes:ImportManager	+ Menu Action Toolbar Add to the Action toolbar if creation inheritance from highlighted object in the doclist is needed.
Show dialog with manager : $value(manager) : $value(manager)	Yes:CreateFolderManager	+ Menu Folder and cabinet support.
Show dialog with manager : $value(manager) : $value(manager)	Yes, CreateManager	+ Menu
Show dialog with manager : $value(manager) : $value(manager)	Yes, ImportStructureManager	+ Menu Action Toolbar
Taxonomy administration action : $value(action)lue(action)	Not Available	
View selected content(s): $value(action)	Yes, D2_ACTION_CONTENT_VIEW_NATIVE	Action Toolbar
Workflow management action : $value(action) (using Configuration $value(config)	Available D2_ACTION_WORKFLOW_UPDATE_PERFORMERS D2_ACTION_WORKFLOW_ABORT	Tasks Toolbar

Classic Menu Action	Smart View Menu Action	Can Be Used In
Workflow task action : $value(action) (using Activity name $value(activityName) and Display dialog $value(displayDialog) for FORWARD and REJECT only)	Available D2_ACTION_TASK_FORWARD D2_ACTION_TASK_REJECT D2_ACTION_TASK_DELEGATE D2_ACTOIN_TASK_NOTE	Tasks Toolbar
Message: Help	Equivalent: Smart view Help	User Menu
Message: Logout	Logout	User Menu

The following table describes support for **Menu Actions Events**:

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_ADVANCED_SEARCH	Not Supported		
D2_ACTION_ADVANCED_SEARCH_CATEGORY_DIALOG	Not Supported		
D2_ACTION_CLIPBOARD_CHANGED	Not Supported		
D2_ACTION_CLIPBOARD_CONTENT	Not Supported		
D2_ACTION_CLIPBOARD_GET	Not Supported		
D2_ACTION_COMMENT	See note		When comment action is present in Action toolbar menu, then if Smart View doclist widget allows comment column, and user has relate or greater to the document, they can see/add comments. This menu item should always be marked as promoted.
D2_ACTION_COMPONENT_NATIVE_ACTION	Not Supported		

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_CONTENT_ANNOTATE_NATIVE_ANNOTATION	Not Supported		
D2_ACTION_CONTENT_CANCEL_CHECKOUT	Supported	Action Toolbar - see note	Click on lock icon or **Unreserve** menu action
D2_ACTION_CONTENT_CHECKIN	Not Supported		Consolidated into D2_ACTION_IMPORT_AS_VERSION
D2_ACTION_CONTENT_CHECKIN_NATIVE_ANNOTATION	Not Supported		
D2_ACTION_CONTENT_CHECKOUT	Supported	Action Toolbar	
D2_ACTION_CONTENT_COMPARE	Not Supported		
D2_ACTION_CONTENT_CREATE	Not Supported		
D2_ACTION_CONTENT_EDIT	Supported	Action Toolbar	
D2_ACTION_CONTENT_EDIT_NATIVE_ANNOTATION	Not Supported		
D2_ACTION_CONTENT_EDIT_WITH_NATIVE_ANNOTATION	Not Supported		
D2_ACTION_CONTENT_EXPORT	Supported	Action Toolbar Rendition Details Version Details	PDF export configuration does not get applied to renditions export.
D2_ACTION_CONTENT_EXPORT_BULK	Not Supported		Bulk export is enabled with `D2_ACTION_CONTENT_EXPORT`
D2_ACTION_CONTENT_EXPORT_FOLDER	Not Supported		Bulk export is enabled with `D2_ACTION_CONTENT_EXPORT`

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_CONTENT_EXPORT_FOLDER_NATIVE	Not Supported		Bulk export is enabled with `D2_ACTION_CONTENT_EXPORT`
D2_ACTION_CONTENT_EXPORT_NATIVE	Not Supported		
D2_ACTION_CONTENT_IMPORT	Supported	Action Toolbar	Used as Add version and consolidates Check-in action with import new version actions
D2_ACTION_CONTENT_IMPORT_AS_VERSION	Supported	Action Toolbar	
D2_ACTION_CONTENT_IMPORT_STRUCTURE	See notes		For import folder, use Show dialog with manager action with manager as ImportStructureManager.
D2_ACTION_CONTENT_PRINT	Supported		
D2_ACTION_CONTENT_VIEW	See notes		Default action when clicking a file object will **View** file. PDF view configuration may be applied. If it is a PDF or basic image file, then Brava CSR is used, otherwise **View** is downloaded using browser controls.
D2_ACTION_VIEW_NATIVE	Supported	Action Toolbar	
D2_ACTION_VIEW_WITH_NATIVE_ANNOTATION	Not Supported		
D2_ACTION_CONVERT_DOC	Supported	Action Toolbar Virtual Document Toolbar	

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_CONVERT_VD	Supported	Action Toolbar Virtual Document Toolbar	
D2_ACTION_COPY	Supported	Action Toolbar	
D2_ACTION_COPY_LINK_IN_CLIPBOARD	Supported	Action Toolbar	For additional ways to customize this action, see "Customizing document link behavior in Smart View" on page 318.
D2_ACTION_COPY_LINK_IN_CLIPBOARD_LOCATE	Not Supported		You can add additional parameters to clipboard URL in the **Message** field by adding the `add_url_params`. For example: `add_url_params==version=CURRENT&logLevel=FINEST&debug=true&docbase=testenv` **Note:** Do not use `!!` to construct a URL.
D2_ACTION_COPY_LINK_IN_CLIPBOARD_SERVLET_VIEW	Not Supported		
D2_ACTION_COPY_LINK_IN_CLIPBOARD_SERVLET_VIEW_CURRENT	Not Supported		
D2_ACTION_COPY_LINK_IN_CLIPBOARD_SERVLET_VIEW_NATIVE	Not Supported		

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_COPY_LINK_IN_CLIPBOARD_SERVLET_VIEW_NATIVE_CURRENT	Not Supported		
D2_ACTION_COPY_LINK_IN_CLIPBOARD_VIEW	Not Supported		
D2_ACTION_COPY_LINK_IN_CLIPBOARD_VIEW_NATIVE	Not Supported		
D2_ACTION_COPY_LINK_IN_CLIPBOARD_VIEW_WITH_NATIVE_ANNOTATION	Not Supported		
D2_ACTION_CUT	Supported	Action Toolbar	
D2_ACTION_DELEGATION_APPLY	Not Supported		
D2_ACTION_DELEGATION_CREATE	Not Supported		
D2_ACTION_DELEGATION_DELETE	Not Supported		
D2_ACTION_DELEGATION_PROPERTIES	Not Supported		
D2_ACTION_DELEGATION_UNDO	Not Supported		
D2_ACTION_DICTIONARY_EXPORT	Not Supported		
D2_ACTION_DICTIONARY_IMPORT	Not Supported		
D2_ACTION_DICTIONARY_PROPERTIES	Not Supported		
D2_ACTION_DICTIONARY_RESTORE	Not Supported		
D2_ACTION_DICTIONARY_SAVE	Not Supported		
D2_ACTION_DICTIONARY_UPDATE	Not Supported		

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_DISPLAY_DIALOG	Not Supported		
D2_ACTION_DISTRIBUTION_ACCEPT	Not Supported		
D2_ACTION_DISTRIBUTION_LAUNCH	Not Supported		
D2_ACTION_DISTRIBUTION_PREPARE	Not Supported		
D2_ACTION_DISTRIBUTION_REJECT	Not Supported		
D2_ACTION_DISTRIBUTION_REPORT	Not Supported		
D2_ACTION_DISTRIBUTION_STOP	Not Supported		
D2_ACTION_DM_TICKET_GENERATE	Not Supported		
D2_ACTION_DOWNLOAD_URLS_REQUEST	Not Supported		
D2_ACTION_DUMP	Supported		
D2_ACTION_EXECUTE	Not Supported		
D2_ACTION_EXECUTE_MANAGER	Not Supported		
D2_ACTION_EXPORT_FROM_URL	Not Supported		
D2_ACTION_FAVORITE_ADD	See notes		Click Star icon in Smart View
D2_ACTION_FAVORITE_REMOVE	See notes		Click Star icon in Smart View
D2_ACTION_FOLDER_CREATE	Not Supported		For import folder, use **Show dialog** with a manager action with manager as `CreateFolderManager`.
D2_ACTION_FOLDERTOVD_CONVERT	Supported	Action Toolbar Virtual Document Toolbar	

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_GROUP_CREATE	Not Supported		
D2_ACTION_GROUP_PROPERTIES	Not Supported		
D2_ACTION_HYPERLINK_COPY_TO_CLIPBOARD	Not Supported		
D2_ACTION_HYPERLINK_CREATE	Not Supported		
D2_ACTION_HYPERLINK_DESTROY	Not Supported		
D2_ACTION_HYPERLINK_LOCATE	Not Supported		
D2_ACTION_HYPERLINK_PAGE_SHIFT	Not Supported		
D2_ACTION_HYPERLINK_PDF_CANCEL_CHECKOUT	Not Supported		
D2_ACTION_HYPERLINK_PDF_CHECKIN	Not Supported		
D2_ACTION_HYPERLINK_PDF_CHECKOUT	Not Supported		
D2_ACTION_HYPERLINK_VERSION_CHANGE	Not Supported		
D2_ACTION_HYPERLINK_VIEW	Not Supported		
D2_ACTION_IMPORT_URL	Not Supported		
D2_ACTION_LIFECYCLE_DCTM	Not Supported		

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_LOCATE_OBJECT	See notes		In Smart View there is a **Location** column in many views that allows the user to access the location of a given object quickly. Go to Location Action menu item is also available on tile objects.
D2_ACTION_LOCATE_OBJECT_DIALOG	Not Supported		
D2_ACTION_LOCATE_SEARCH	Not Supported		
D2_ACTION_LOCATE_TASK	Not Supported		
D2_ACTION_MASS_UPDATE	Not Supported		
D2_ACTION_OBJECT_DESTROY	Not Supported		
D2_ACTION_OPEN_FOLDER	Not Supported		
D2_ACTION_OPEN_GROUP	Not Supported		
D2_ACTION_OPEN_PUBLISH	Not Supported		
D2_ACTION_OPEN_SNAPSHOT	Supported	Action Toolbar	
D2_ACTION_OPEN_URL	Supported	Action Toolbar	
D2_ACTION_OPEN_VD	Supported	Action Toolbar	
D2_ACTION_PASTE	Supported	Action Toolbar	
D2_ACTION_PASTE_LINK	Supported	Action Toolbar	

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_PERMISSION_SHOW	See notes		For viewing permission, use **Show dialog** with a manager action with manager as `PermissionsManager`.
D2_ACTION_PREFERENCES_SHOW	Not Supported		
D2_ACTION_PUBLISH_PROFILE	Not Supported		
D2_ACTION_QUERY_CATEGORY_DIALOG	Not Supported		
D2_ACTION_QUERYFORM_SHOW	Not Supported		
D2_ACTION_REFERENCE_CREATE	Not Supported		
D2_ACTION_REFERENCE_REFRESH	Not Supported		
D2_ACTION_REFRESH_DOCUMENT	Not Supported		
D2_ACTION_REFRESH_WIDGET	Not Supported		
D2_ACTION_RELATION_CREATE	Supported	Action Toolbar	
D2_ACTION_RELATION_DESTROY	Supported	Relations Details	
D2_ACTION_RELOAD_WORKSPACE	Not Supported		
D2_ACTION_REMOVE_RENDITION	Supported	Renditions Details	
D2_ACTION_RENDITION_ADD	Supported	Action Toolbar Task Documents Actions Toolbar Collection Items Actions Toolbar Renditions	
D2_ACTION_RENDITION_REQUEST	Not Supported		

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_SEARCH_CLEAR	Not Supported		
D2_ACTION_SEARCH_DOCUMENT	Not Supported		
D2_ACTION_SEARCH_HIGHLIGHT_TERMS	Not Supported		
D2_ACTION_SELECT_CONTENT_WITH_DQL	Not Supported		
D2_ACTION_SENDMAIL	See notes		For send mail, use Show dialog with manager action with manager as SendMailManager. For additional ways to customize this action, see "Customizing document link behavior in Smart View" on page 318.
D2_ACTION_SNAPSHOT_CREATE	Supported	Virtual Document Toolbar	
D2_ACTION_SUBSCRIBE	Not Supported		
D2_ACTION_SYNCPLICITY	Not Supported		
D2_ACTION_TASK_ACQUIRE	Supported	Through Set Task to Read action	
D2_ACTION_TASK_DELEGATE	Supported		
D2_ACTION_TASK_FORWARD	Supported		
D2_ACTION_TASK_NOTE	Supported		
D2_ACTION_TASK_PRIORITY	Supported		
D2_ACTION_TASK_PROCESS	Not Supported		

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_TASK_READ	Supported, through Set Task to read action		
D2_ACTION_TASK_REJECT	Not Supported		
D2_ACTION_TAXONOMY_CHECKIN	Not Supported		
D2_ACTION_TAXONOMY_EXPORT	Not Supported		
D2_ACTION_TAXONOMY_IMPORT	Not Supported		
D2_ACTION_TAXONOMY_LOCATE	Not Supported		
D2_ACTION_TAXONOMY_PROPERTIES	Not Supported		
D2_ACTION_TAXONOMY_RESTORE	Not Supported		
D2_ACTION_TAXONOMY_SAVE	Not Supported		
D2_ACTION_USER_CREATE	Not Supported		
D2_ACTION_USER_PROPERTIES	Not Supported		
D2_ACTION_VD_ADD_CHILD_SELECTED	Supported	Virtual Document Toolbar	
D2_ACTION_VD_CLEAR_VD	Not Supported		
D2_ACTION_VD_INSERT_INHERITED	Not Supported		
D2_ACTION_VD_INSERT_INHERITED_COMPONENT	Supported	Virtual Document Toolbar	
D2_ACTION_VD_REMOVE_CHILD_SELECTED	Supported	Virtual Document Toolbar	
D2_ACTION_VD_SET_BINDING	Not Supported		

Classic Menu Action	Smart View Menu Action	Can Be Used In	Notes
D2_ACTION_VD_SET_CHILD_BINDING_VERSION	Supported	Virtual Document Toolbar	
D2_ACTION_WIDGET_FULLSCREEN	Not Supported		
D2_ACTION_WORKFLOW_ABORT	Supported		
D2_ACTION_WORKFLOW_LAUNCH	Not Supported		
D2_ACTION_WORKFLOW_LAUNCH_SCHEDULED	Not Supported		
D2_ACTION_WORKFLOW_MANAGE_ATTACHMENTS	Supported		
D2_ACTION_WORKFLOW_UPDATE_PERFORMERS	Supported		
D2_EVENT_<various>	Not Supported		
D2_ACTION_SHARE_TO_CORE	MenuDocumentShare sub menu		Not Supported in Classic
D2_ACTION_SHARE_TO_CORE_C2_OVERRIDE	MenuDocumentShare sub menu		Not Supported in Classic
D2_ACTION_MANAGE_SHARE	MenuDocumentShare sub menu		Not Supported in Classic
D2_ACTION_STOP_SHARE	MenuDocumentShare sub menu		Not Supported in Classic

36.5 Smart View: Widget View Menu Config Support

The following table identifies Widget View Menu Config support in the Smart View client.

Configuration	Specific Setting	Smart View Client	Notes
Options...			
User Settings	General	Partial Support - see notes	Temporary path Checkout path Default language

Configuration	Specific Setting	Smart View Client	Notes
User Settings	User Interface	Partial Support - see notes	Group folders, then files Accessibility enabled
User Settings	Events	Not Supported	
Workspace	Any	Not Supported	
Theme	Any	Supported	Smart View supports theme configuration through Smart View Theme.
Graphics	Any	Not Supported	
Content Assist	List Assistance Options	Supported	
Smart View Landing Page	Any	Supported - see notes	Smart View equivalent, Smart View Landing Page configuration.
Smart View Theme	Any	Supported - see notes	Smart View equivalent, Smart View Theme configuration
Widget...			
AdminDictionariesWidget		Not Supported	
AdminGroupsWidget		Not Supported – see notes	Smart View SDK contains an example widget for group administration.
AdminTaxonomiesWidget		Not Supported	
AdminsUsersWidget		Not Supported	
AdvPublishComponentWidget		Not Supported	
AdvPublishDetailWidget		Not Supported	
AdvPublishManagerWidget		Not Supported	
AdvPublishOVerlayWidget		Not Supported	

Configuration	Specific Setting	Smart View Client	Notes
BravaCSRViewerWidget		Supported	This viewer widget is supported in both Classic and Smart View. BravaCSR is implemented in Smart View. See "List of Widgets" on page 457 for more information.
BravaEnterpriseViewerWidget		Supported	
BrowserWidget		Supported - see notes	The doclist widget has an option for Smart View to display a tree browser.
CenterStageWidget		Not Supported - see notes	Use CommentsWidget
CheckoutDocumentWidget		Supported - see notes	Is a tile that can be configured on Smart View landing page.
CommentsWidget		Supported - see notes	Comments are not displayed in a specific widget but through a comments dialog found in a doclist when the Comment action is enabled in Menu Smart View.
DelegationWidget		Not Supported	
DetailsAuditWidget		Supported	
DetailsHyperlinksWidget		Not Supported	
DetailsMarkupsWidget		Not Supported	
DetailsLocationsWidget		Supported - see notes	For Smart View client, use `LocationsWidget`.

Configuration	Specific Setting	Smart View Client	Notes
DetailsRelationsWidget		Supported - see notes	Through Smart View information (properties) detail pages, a relations details page can be added based on this widget configuration.
DetailsRenditionsWidget		Supported - see notes	Through Smart View information (properties) detail pages, a renditions details page can be added based on this widget configuration.
DetailsRetentionsWidget		Not Supported - see notes	OpenText Documentum CM honors retentions from Retention Policy Services.
DetailsSnapshotsWidget		Supported - see notes	The VirtualDocWidget configuration is reused for displaying Snapshot details in Smart View instead of having two separate widgets.
DetailsVersionsWidget		Supported - see notes	Through Smart View information (properties) detail pages, a versions details page can be added based on this widget configuration.
DistributionWidget		Not Supported	
DocgalleryWidget		Supported - see notes	The doclist widget can be configured to allow for a grid view which is the Smart View equivalent of the Docgallery Widget.

Configuration	Specific Setting	Smart View Client	Notes
DoclistWidget		Supported - see notes	Through Smart View Landing page configuration file, doclist widgets can be included for use to browse the entire repository, users home directory, shortcuts to specific locations in the repository, or to display results of DQL query. In Doclist Parameters only Content type is supported parameter for Smart View client.
DumpWidget		Supported - see notes	See “Debugging configuration and content issues using Dump” on page 343 for more information on configuring dump functionality.
ExternalWidget		Supported - see notes	Supported via OpenText Documentum CM Smart View Software Development Kit
FacetsWidget		Supported - see notes	Facets can be used in both search results and in doclist widget.
FavoritesWidget		Supported - see notes	Favorites is a tile that can be configured on Landing Page. Parameters are not supported.
PropertiesWidget		Supported - see notes	Access to properties is available by clicking the information icon in the Action Toolbar. Parameters are not supported
QuickSearchWidget		Supported - see notes	In Smart View, quick search is available in the header of the page.

Configuration	Specific Setting	Smart View Client	Notes
RecentlyAccessedWidget		Supported - see notes	Smart View client only. Recently Accessed Widget can be configured to display on the Smart View landing page.
RecyleBinWidget		Supported	
SearchFormWidget		Supported	Query forms can be launched from landing page shortcut or from Saved Searches Tile.
SearchWidget		Supported - see notes	Saved searches widget can be configured to display on the Smart View landing page. Parameters are not supported.
TaskAttachmentWidget		Supported - see notes	The Smart View client implements support for a Task inbox and associated Task details differently than Classic. While some widgets are used, others are just part of the overall feature. See the *OpenText Documentum CM Smart View Client User Guide* for a description of the Task interface.
SharedDocumentWidget		Supported	

Configuration	Specific Setting	Smart View Client	Notes
TaskDetailsWidget		Supported - see notes	The Smart View client implements support for a Task inbox and associated Task details differently than Classic. While some widgets are used, others are just part of the overall feature. See the *OpenText Documentum CM Smart View Client User Guide* for a description of the Task interface.
TaskFoldersWidget		Not Supported	
TaskNotesWidget		Supported - see notes	The Smart View client implements support for a Task inbox and associated Task details differently than Classic. While some widgets are used, others are just part of the overall feature. See the *OpenText Documentum CM Smart View Client User Guide* for a description of the Task interface.
TaskOverviewWidget		Supported - see notes	The Smart View client implements support for a Task inbox and associated Task details differently than Classic. While some widgets are used, others are just part of the overall feature. See the *OpenText Documentum CM Smart View Client User Guide* for a description of the Task interface.

Configuration	Specific Setting	Smart View Client	Notes
TaskPerformersWidget		Supported - see notes	The Smart View client implements support for a Task inbox and associated Task details differently than Classic. While some widgets are used, others are just part of the overall feature. See the *OpenText Documentum CM Smart View Client User Guide* for a description of the Task interface.
TasksWidget		Supported - see notes	The Smart View client implements support for a Task inbox and associated Task details differently than Classic. While some widgets are used, others are just part of the overall feature. See the *OpenText Documentum CM Smart View Client User Guide* for a description of the Task interface.
ThumbnailWidget		Supported - see notes	Supported in doclist widget, but not in checkout and search widgets.
VirtualDocWidget		Supported	
WorkflowOverview Widget		Supported	
WorkflowPreviewWidget		Supported - see notes	The Smart View client implements support for preview differently. In a workflow report, Smart View uses a graphical timeline.

36.6 Smart View: Property Page Config Support

The following layouts/containers are supported:

Layouts/Containers	Smart View Client Support	Notes
Containers	Supported	Page, Column (1,2, or more), Tabs, FieldSets, Grids span across the entire page.
Tab Group	Supported	1st level tabs are moved into the Spy scroll. 2nd - 4th level tabs are nested across the property page with left and right padding. 5th level and beyond appear nested without padding. Attributes in a tab: • All fields flow across the two available columns, left to right, top to bottom. • Nested containers take the full width.
Field set	Supported	1st level field sets moved into Spy scroll. Attributes in a fieldset: • All fields flow top to bottom. • Right side space remains empty. • Nested containers take the full width.

Layouts/Containers	Smart View Client Support	Notes
Grid	Supported	Grid width behavior depends on the **Auto-fit (Smart View)** setting. • If enabled, grids vary in width based on columns. Grids sizes are 1/2, 1/4, 1/3, or the full width of the property page. Grids in a column container span the available page width based on these grid size rules. • If disabled, default column widths are defined in the **Width** field of each grid item. Users can adjust column widths in the client.
Columns set	Supported	Column (with size 1): • All fields flow top to bottom in left hand column. • Right side space remains empty. • Nested containers take the full width. Column (with size 2 or more): • All fields flow across the two available columns left to right, top to bottom • Nested containers take the full width **Note:** Smart View only displays up to two columns.

Layouts/Containers	Smart View Client Support	Notes
Containerless fields	Supported	All the Containerless fields will be raised to the top of the General property page. The Containerless fields will be laid out using the CSUI layout, where up-to 5 fields will flow top to bottom and beyond will move the the right column when the total fields are 8 or more. 5 fields – all left column 6 fields – all left column 7 fields – all left column 8 fields – 5 in left and 3 in right column 9 fields – 5 in left and 4 in right column, etc

The following controls are supported in Smart View property pages:

Controls	Smart View Client Support	Notes
Comment field (not HTML enabled)	Supported	A comment field where the HTML option is not selected will be constrained to the available space where an attribute value is presented.
Comment field (HTML enabled)	Supported	A comment field where the HTML option is enabled will span the widget of the property page.
Separator	Supported	
Label	Supported	A label field is treated as a Text Field set for read only.
Text Field	Supported	
Password Field	Supported	
Memo	Supported	
Rich text editor	Supported - See notes	Not supported in mass update.
Boolean	Supported	A toggle represents the two possible states of a boolean field.

Controls	Smart View Client Support	Notes
Date	Supported	
URL	Supported	
Combo	Supported	
Editable Combo	Supported	
List Field	Supported	
Editable List Field	Supported	The configuration option 'Enable values import export' is supported.
Radio	Supported	Two values will be located horizontal to each other. Three to five values will display vertically as radio buttons. Six or more values will automatically convert to a combo list.
Checkbox	Supported	Up to five values will be display vertically as checkboxes. Six or more values will automatically convert to a combo list.
Relation	Supported	
Editable List field with no DQL or Dictionary	Supported	

The following data sources are supported in Smart View property pages:

Data Source	Smart View Client Support	Notes
DQL	Supported	
Dictionary	Supported	
Taxonomy	Supported	
Asynchronous	Supported	

The following expressions are supported in Smart View property pages:

Expression	Smart View Client Support	Notes
Enabled When	Supported	
Visible When	Supported	
Required When	Supported	
Validation constraint	Supported	
Electronic Signature	Supported	

Chapter 37

Brava! Client Parameter Configuration

37.1 Brava! Client parameters overview

This document details the Brava! Client parameters that can be used to customize your for Brava! product configuration. The Brava! parameters affect the state of the Brava! Clients. The audience for this document includes developers, programmers, and integrators with a solid knowledge of COM API programming and ActiveX/ HTML technology who are configuring and customizing their implementations for the specific needs of their environment.

The optional Brava! parameters affect the state of the Brava! ActiveX, HTML, and HTML Video Clients and can be used to enable or disable (remove) specific features and functionality from the user interface.

37.2 ActiveX Client Parameters

For the Brava! integration, parameters that control the ActiveX Client are stored in the `brava_parameters.properties` file which can be edited to meet your requirements. It should be noted that the `brava_parameters.properties` file contains parameters for setting behaviors that are unique to Brava!. Core Brava! Enterprise parameters that are not listed in the`brava_parameters.properties` file are not applicable to the integration.

37.2.1 General Control Parameters

ActiveXControl Width	**Value:** *<percentage>* **Description:** Specifies the width percentage of the ActiveX Client. **Example:** `ActiveXControlWidth=100%`
ActiveXControl Height	**Value:** *<percentage>* **Description:** Specifies the height percentage of the ActiveX Client. **Example:** `ActiveXControlHeight=100%`
BravaClientXR esourceVersion	**Value:** *<URL>*: not set **Description:** Controls whether the `BravaClientXResource_XYZ.bin` component needs to be downloaded to a particular client or not, depending on version match. The default value is the value specified by the required parameter *`IntegrationVersion`*. **Example:** `BravaClientXResourceVersion=`

brava_cab_version	**Value:** *<version number>* **Description:** Indicates the version number of the Brava! cab file. *brava_cab_version* numbers are separated by commas. **Example:** `brava_cab_version=16,4,0,178`
ClientLogging	**Value:** *<0-5>* **Description:** Determines the level of logging that can be done by the Brava! Client: • **0** = ALL: designates all levels of messages are to be logged • **1** = DEBUG: designates fine-grained informational events that are most useful to debug an application • **2** = INFO: designates informational messages that highlight the progress of the application at coarse-grained level • **3** = WARN: designates potentially harmful situations • **4** = ERROR: (default) designates error events that might still allow the application to continue running • **5** = FATAL: designates very severe error events that cause the application to abort Only ERROR and FATAL messages are logged when the default log level is set to (`4`) ERROR. The *ClientLogging* setting is also used to control the level of log detail that the Brava! system logs output. Increasing the level impacts the output of the *logback.xml* configured logging. **Example:** `ClientLogging=4`
ClientLogFileLength	**Value:** *<bytes>* **Description:** Indicates the maximum length of the file in bytes. If the file grows past this size, the current information in the log file is removed before new information is written to the log. **Example:** `ClientLogFileLength=1000`
ClientLogFile	**Value:** *<file path>* **Description:** Indicates file path and name on the end user's machine where a log file containing Client information can be written to. **Example:** `ClientLogFile=%USERPROFILE%/Application Data/D2/Logs/Clientlog.txt`

CompareView Mode	**Value:** *<string>* **Description:** Determines the display mode in Brava! Client for files being compared. Possible values for this parameter are: • OVERLAY DIFFERENCES • OVERLAY • SIDE-BY-SIDE • DELETIONS • ADDITIONS • UNCHANGED • OPEN FILE • COMPARE FILE • TEXT COMPARE **Example:** `CompareViewMode=Overlay Differences`
DefaultBackgro undColor	**Value:** *<WHITE|GRAY (or GREY)|BLACK|DEFAULT>* **Description:** This parameter can be used to specify the background color for Brava! to use by default. If the *`EnableBackground`* parameter is set to `true`, the user can still change the background color via the **Background Color** tool on the client toolbar. A value of `default` means that the background color displays as determined by the file format driver on conversion. **Example:** `DefaultBackgroundColor=DEFAULT`
DefaultMono	**Value:** *<TRUE|FALSE>* **Description:** This setting can be added to specify whether Brava! loads color vector files in monochrome mode or in color. The monochrome mode can still be changed using the **Monochrome** tool on the toolbar. `False` indicates for Brava! to load the files in color. **Example:** `DefaultMono=FALSE`

<table>
<tr><td>DateFormat</td><td>Value: <<i>date codes</i>>

Description: Determines user-defined pattern for date and time formatting. The <code>DateFormat</code> only applies to places where the date and time are shown together, such as the Changemark display, the appended redaction and Changemark notes pages, etc. <code>DateFormat</code> does not apply to the <code>%date</code> and <code>%time</code> tokens in the banners and watermarks since <code>%date</code> shows only date and <code>%time</code> shows only time

Time Format Syntax: To specify the time zone see the <code>TimeZone</code> setting. The formatting codes are preceded by a percent sign (%). Characters that do not begin with % are copied unchanged as text. The formatting codes for <code>DateFormat</code> are listed below:

<pre>%a   Abbreviated weekday name
%A   Full weekday name
%b   Abbreviated month name
%B   Full month name
%c   Date and time representation appropriate for locale
%d   Day of month as decimal number (01 - 31)
%H   Hour in 24-hour format (00 - 23)
%I   Hour in 12-hour format (01 - 12)
%j   Day of year as decimal number (001 - 366)
%m   Month as decimal number (01 - 12)
%M   Minute as decimal number (00 - 59)
%p   Current locale's A.M./P.M. indicator for 12-hour clock
%S   Second as decimal number (00 - 59)
%U   Week of year as decimal number, with Sunday as 1st day of week
 (00 - 53)
%w   Weekday as decimal number (0 - 6; Sunday is 0)
%W   Week of year as decimal number, with Monday as 1st day of week
 (00 - 53)
%x   Date representation for current locale
%X   Time representation for current locale
%y   Year without century, as decimal number (00 - 99)
%Y   Year with century, as decimal number
%z, %Z Either the time-zone name or time zone abbreviation,
 depending on
registry settings; no characters if time zone is unknown
%%   Percent sign</pre>
Example: <code>DateFormat=%x %H:%M</code></td></tr>
<tr><td>EnableThreadedXdlLoad</td><td>Value: <<i>TRUE|FALSE</i>>

Description: When <code>true</code>, the Brava! Client control loads documents in a separate thread and does not allow focus to be set to the IE browser during the loading process. When <code>false</code>, the document loading process happens in the IE browser's main thread, which can cause conflicts with other applications that need to receive events from Windows main event loop.

Example: <code>EnableThreadedXdlLoad=TRUE</code></td></tr>
</table>

<table>
<tr>
<td>HyperlinkLaunch</td>
<td>Value: <<i>ONE|DEFAULT</i>>

Description: This setting determines how hyperlinks launch in Brava!. The valid values are:

• one = All documents launch in a single separate window as Brava! (two windows).
• default = All documents launch in multiple separate windows (multiple windows).

Example: <code>HyperlinkLaunch=DEFAULT</code></td>
</tr>
<tr>
<td>NotificationListener</td>
<td>Value: <<i>URL</i>>

Description: This parameter allows the user to specify a URL to receive notification of integration application events

The events are:

<code>DOC_CLOSED</code>
<code>DOC_LOADED</code>
<code>MARKUP_CLOSE_TOOL</code>
<code>MARKUP_OPEN_COMPLETE</code>
<code>MARKUP_OPEN_FOR_REVIEW_COMPLETE</code>
<code>MARKUP_SAVE_COMPLETE</code>
<code>MARKUP_SAVE_AS_COMPLETE</code>
<code>DOC_SENT_TO_PRINTER</code>
<code>MARKUPSTAMPTEMPLATE_CLOSE_TOOL</code>
<code>MARKUPSTAMPTEMPLATE_OPEN_COMPLETE</code>
<code>MARKUPSTAMPTEMPLATE_SAVE_COMPLETE</code>
<code>MARKUPSTAMPTEMPLATE_SAVE_AS_COMPLETE</code>
<code>VIEWER_CLOSED</code>

The URL should take three parameters, <i><code>Event</code></i>, <i><code>Data</code></i>, and <i><code>User</code></i>. The <i><code>Event</code></i> parameter contains one of the events listed above. The data contains the file path of the file that the event is acting on. In the example below, the call would be: <code>http://localhost/NotificationListener?Event=value&Data=value&User=value</code>

Example: <code>NotificationListener=http://localhost/<notificationservlet></code>

Please refer to the <i><code>NotificationListenerCustomParams</code></i> setting below for additional information about this parameter.</td>
</tr>
</table>

NotificationListenerCustomParams	**Value:** <*URL*>: not set
	
	Description: This parameter allows you to specify parameters to be returned to you on notification events when used in conjunction with the *`NotificationListener`* parameter.
	
	The usage format is:
	
	`notificationlistenercustomparams=custom1=value1&custom2=value2`
	
	The value can also be written as:
	
	`&custom1=value1&custom2=value2` or `Custom1=9999&Custom2=C:/hello world`
	
	This parameter must be used in addition to:
	
	`notificationlistener=http://localhost:9090/NotificationServlet`
	
	The resulting call would be:
	
	`http://<hostname>/test.asp?Event=value&Data=value&User=value&custom1=value1&custom2=value2`
	
	Example: `NotificationListenerCustomParams=`
	
	`NotificationListenerCustomParams=Name1=Value1&Name2=Value2&Name3=Value3`
	
	`NotificationListener=http://MYMACHINE/Project1/MarkupNotification.asp`
	
	The query string returned would be:
	
	`Name1=Value1&Name2=Value2&Name3=Value3&Event=MARKUP_SAVE_COMPLETE&Data=%5Canother+red+cloud&User=John+Johnson`

<table>
<tr>
<td>PanelToDisplay</td>
<td>
Value: <0-8|LAST|NONE>

Description: This parameter controls which of the five available task pane panels display on startup. Valid values are:

<code>0</code> = last panel
<code>1</code> = no panel
<code>2</code> = Thumbnail panel
<code>3</code> = Bookmark panel
<code>4</code> = Changemark panel
<code>5</code> = Verify panel
<code>6</code> = Takeoff panel
<code>7</code> = Checkview panel
<code>8</code> = CAD Attributes panel. (Valid for CAD documents only.)
<code>LAST</code> = displays the panel that was open when the last Brava! session ended
<code>NONE</code> = starts a Brava! session with the Task Pane closed

Example: <code>paneltodisplay=4</code>
</td>
</tr>
<tr>
<td>PersistUserSettings</td>
<td>
Value: <TRUE|FALSE>

Description: When <code>true</code>, user settings (markup color, print settings, measurement settings, redaction reasons, etc.) are saved at the end of the session. If <code>false</code>, the settings are not saved. Settings are stored in the application’s <code>ViewerConfig.xml</code> file and two .ini files, installed by default to <user application data directory><code>\IGC\</code><product name><code>\</code> (<code>ViewerConfig.xml</code>, <code>reason.ini</code>, and <code>measure.ini</code>) or where specified by <code>UserSettingsPath</code>.

Note: Any settings that are specified through the Brava! client parameters override any locally persisted settings. All persistence files for the Brava! Client will either be in the default location as specified above, or in the directory as specified in <code>UserSettingsPath</code>, but NOT BOTH.

If set to <code>false</code>, the <code>ViewerConfig.xml</code>, <code>reason.ini</code>, and <code>measure.ini</code> settings are not read and current settings are not saved (persisted) in these files at the end of the session.

Example: <code>persistusersettings=TRUE</code>
</td>
</tr>
</table>

UserSettingsPath	**Value:** <*folder path*> **Description:** Specifies the path a user's persisted settings (contained in `ViewerConfig.xml`, `reason.ini`, and `measure.ini`) are read from and written to (when `PersistUserSettings=true`). Brava! must find these files in the directory where the Brava! Client is installed to, or a sub-directory below it, however, Brava! can use the *`UserSettingsPath`* setting to specify a custom location of where to find the persistence files. In this case, the custom location of persistence files must be specified in the *`UserSettingsPath`* for this feature to work. **Example:** `UserSettingsPath=%USERPROFILE%\Application Data\IGC\Brava! Client` In this sample, a full path is used to identify a custom ini location: `UserSettingsPath=%BRAVACLIENT_HOME%\x<version>\` **Note:** The persistence files can only be shared amongst multiple users on a machine if the *`UserSettingsPath`* is specified the same for all users.

<table>
<tr>
<td>SearchTextAtStartup</td>
<td>
Value: <<i>search term</i>>: not set

Description: Specifies the term to search to when the document is initially loaded into the viewer. The search term can be any valid word, phrase, or alpha numeric string. Below are the Brava! escape codes that can be used with <code>HighlightTerms</code> and <code>SearchTextAtStartup</code>.

<code>" (quote) is %26quot%3B</code>
<code>' (apostrophe) is %26apos%3B</code>
<code>& (ampersand) is %26amp%3B</code>
<code>< (less than) is %26lt%3B</code>
<code>> (greater than) is %26gt%3B</code>

Examples: <code>searchtextatstartup=</code>

To search a term at start up such as <code>"Law</code> the parameter would be:

<code>SearchTextAtStartup=%26quot%3BLaw</code>

Where <code>%26</code> and <code>%3B</code> are typical URL escape codes for ampersand and semi-colon.

When using the prefix-style search string commands (see Brava! Online help), you can highlight and search for multiple terms using the following syntax with the Brava! client parameters <code>HighlightTerms</code> and <code>SearchTextAtStartup</code>:

To highlight multiple terms:
<code>HighlightTerms=anyof:wi:"poor dog" cupboard</code>
This example highlights both the phrase “poor dog” and the word “cupboard”. Each term (i.e. “poor dog”) is highlighted in its own color.
To search for multiple terms:
<code>SearchTextAtStartup=anyof:wi:"poor dog" cupboard</code>
This example highlights the first of occurrence of either “poor dog” or “cupboard”, and allows the user to go to the next occurrence of either of these terms when they search for the next term.

</td>
</tr>
<tr>
<td>SecurityOptionsFromServer</td>
<td>
Value: <<i>TRUE|FALSE</i>>

Description: Determines if security settings come from the server, or can be set interactively by the user. When set to <code>false</code>, the viewer displays the Visual Rights options dialog when publishing a CSF file from a native file via the Publish to CSF menu item. The default is <code>true</code>, which means the rights applied to the CSF are returned from the Brava! Server's <code>security.xml</code>.

Example: <code>securityoptionsfromserver=TRUE</code>
</td>
</tr>
</table>

<table>
<tr>
<td>TimeZone</td>
<td>
Value: <TRUE|FALSE>
Description: Determines the time zone to use for the date and time stamp. For example, if you are working in one time zone on a project that is located in another time zone and would like to have all time stamps recorded on project time. The default value is empty meaning the <code>DateFormat</code> takes the local time zone on the client machine. If the <code>TimeZone</code> value is present, it overrides the client side local time zone.
The valid syntax to use is:
<code>TimeZone=tzn[+|-]hh[:mm[:ss]][dzn]</code>
Where:

<code>tzn</code> = Three-letter time-zone abbreviation.
<code>hh</code> = Difference in hours between UTC (Universal Time Coordinate) and local time. Optionally signed.
<code>mm</code> = Minutes. Separated from <code>hh</code> by a colon (:). Optionally signed.
<code>ss</code> = Seconds. Separated from <code>mm</code> by a colon (:).
<code>dzn</code> = The three-letter daylight-saving-time zone such as PDT. If daylight saving time is never in effect in the locality, set <code>TimeZone</code> without a value for <code>dzn</code>. The C run-time library assumes the United States' rules for implementing the calculation of Daylight Saving Time (DST).

If <code>TimeZone</code> is not set, the time zone information specified by the client’s operating system (in the Control Panel’s Date/Time application) is used.
Examples: <code>timezone=</code>

To set the <code>TimeZone</code> parameter to correspond to the current time zone in Germany, you can use the following statement:
<code>TimeZone=GST+1GTD</code>
This string uses GST to indicate German standard time, assumes that Germany is one hour ahead of UTC, and assumes that daylight savings time is in effect.
To set the <code>TimeZone</code> parameter for New York (Eastern Time) or California (Pacific Time), you can use:
<code>TimeZone=EST5EDT</code>
(Eastern Standard Time is 5 hours behind UTC time) or:
<code>TimeZone=PST8PDT</code>
(Pacific Standard Time is 8 hours behind UTC time)

</td>
</tr>
<tr>
<td>MachineUserName</td>
<td>
Value: <name>
Description: Overrides value determined from <code>::GetUserNameW()</code> Windows System Call.
Example: <code>MachineUserName=</code>
</td>
</tr>
</table>

MachineIP	**Value:** *<IP addres>* **Description:** Overrides value determined from Windows System Calls. **Example:** `MachineIP=123.456.789`
WatermarkBannerFontName	**Value:** *<font name>* **Description:** Specifies the default font name to use for Watermarks and Banners. Any valid font can be used (font must exist on all systems or the font reverts back to the default font if specified font is not available). **Example:** `WatermarkBannerFontName=Arial`
BannerFontSize	**Value:** *<font size>* **Description:** Specifies the default font size to use for banners. **Example:** `BannerFontSize=12`
WatermarkBannerFontStyle	**Value:** *<0-3>* **Description:** Controls the default style used for the Watermark and Banner font. Valid values are: • `0` = Regular • `1` = Bold • `2` = Italic • `3` = Bold Italic **Example:** `WatermarkBannerFontStyle=1`
WatermarkTileSize	**Value:** *<tile size>* **Description:** Specifies the width and height of the watermark. Use this setting to create multiple “tiled” watermarks in printed output or exported PDF and TIFF as opposed to just the standard single watermark. The format is: *`<tilewidth>`*, *`<tileheight>`*, *`<unit>`* Where `Unit` can be equal to `i` (for inches) or `m` (for millimeters). In the below example, the watermark should have a width of 2 inches and a height of 3 inches. The actual watermark tile can be wider and/or taller than the specified parameter value since the watermark tiles stretch so that the page is covered with the watermarks. The number of watermarks placed along the horizontal or vertical axis is equal to the size of the output page along one axis divided by the parameter value for that axis. (The value is truncated to the closest whole number and is equal to or greater than one). **Example:** `WatermarkTileSize=2,3,i`

useDocumentIdAsCacheFilename	**Value:** <*TRUE|FALSE*> **Description:** When enabled (`TRUE`) uses the document objectId as cacheFileName in Brava displayListCache. By default, the setting is `FALSE` and uses the document name as cacheFilename. **Example:** `useDocumentIdAsCacheFilename=TRUE`

37.2.2 General Integration Parameters

The Integration Brava! Server parameters in the following table are used in the ActiveX client when integrated with another 3rd party product.

IntegrationHttpProxyServer	**Value:** `name:`*<servername>*`:`*<port number>* **Description:** Sets HTTP proxy configuration for Brava! ActiveX Client. This parameter encodes two pieces of data: `name` and `bypasslist`. For example, `name:someproxyhostname:8888bypasslist:host1;host2;host3` configures Brava! to use port 8888 on port <*someproxyhostname*> as a proxy while bypassing the proxy for domains `host1`, `host2`, and `host3`. **Example:** `integrationhttpproxyserver=name:CARL:8080:8888`
IntegrationName	**Value:** <*unique name*> **Description:** Identifies the prefix for custom integration parameter names. **Example:** If `integrationname=exampleintegration`, custom integration parameters are specified with `exampleintegration.XYZ=ABC` where `XYZ` is the custom parameter name and `ABC` is the parameter value.
IntegrationVersion	**Value:** <*version number*> **Description:** Specifies the version of the `Integration.dll`. This lets the Brava! ActiveX control know when a new `Integration.dll` file needs to be downloaded. **Example:** `integrationversion=7.0.0.5`
IntegrationHttpTimeoutConnect	**Value:** <*number of minutes*> **Description:** Indicates the time-out value, in minutes, to use for Internet connection requests. **Example:** `integrationhttptimeoutconnect=1`
IntegrationHttpTimeoutReceive	**Value:** <*number of minutes*> **Description:** Indicates the time-out value, in minutes, to receive a response to a request. **Example:** `integrationhttptimeoutreceive=5`

IntegrationHttpTimeoutSend	**Value:** <*number of minutes*> **Description:** Indicates the time-out value, in minutes, to send a request. **Example:** `integrationhttptimeoutsend=5`

37.2.3 Toolbar and Menu Item Controls

EnableAnimationSetting	**Value:** <*TRUE|FALSE*> **Description:** Enables/disables the **Enable Animation** item in the right mouse button menu. **Example:** `EnableAnimationSetting=TRUE`
AnimationSetting	**Value:** <*ON|OFF|LAST*> **Description:** Determines the default state of the **Enable Animation** option when *`EnableAnimationSetting`* is `true`. `Last` (the default) indicates the state of the setting when the user exited the last Brava! session (either on or off). **Example:** `AnimationSetting=last`
EnableBackground	**Value:** <*TRUE|FALSE*> **Description:** Enables/disables the **Background Color** item in the right mouse button menu and bottom toolbar. **Example:** `EnableBackground=TRUE`
EnableBannerEdit	**Value:** <*TRUE|FALSE*> **Description:** Enables/disables the **Watermark/Banners** item in the **Print** button menu. When `false`, the menu item is removed and the **Watermark/Banners** button of the **Print** dialog does not open the **Banner** dialog. **Example:** `EnableBannerEdit=TRUE`
EnableCopyRegion	**Value:** <*TRUE|FALSE*> **Description:** When set to `false`, all UI and functionality associated with ability to copy view regions as bmp data to the Window's clipboard is removed. **Copy Region** is disabled from the right mouse button menu. **Example:** `EnableCopyRegion=TRUE`
EnableCopyText	**Value:** <*TRUE|FALSE*> **Description:** This parameter controls whether copying of text is allowed in Brava!. `True` permits copying of highlighted text. Use `false` to disable text copying functionality. **Example:** `EnableCopyText=TRUE`

<table>
<tr><td>EnableExternal Bookmarks</td><td>Value: <<i>TRUE|FALSE</i>>

Description: Reserved for integrations, this parameter controls whether the External Bookmarks functionality is enabled. <code>True</code> allows setting session based external bookmarks and enables a bookmarks button on the main toolbar. Use <code>false</code> to disallow setting external bookmarks.

Example: <code>EnableExternalBookmarks=TRUE</code></td></tr>
<tr><td>EnableEnhance dDisplay</td><td>Value: <<i>TRUE|FALSE</i>>

Description: Enables/disables the Smooth Edges (Slower) item in the right mouse button menu. This feature controls the rendering and edge smoothing.

Default: <code>EnableEnhancedDisplay=TRUE</code></td></tr>
<tr><td>EnhancedDispl aySetting</td><td>Value: <<i>ON|OFF|LAST</i>>

Description: Determines the default state of the Enable Enhanced Display option when <i><code>EnableEnhancedDisplay</code></i> is <code>true</code>. <code>Last</code> (the default) indicates the state of the setting when the user exited the last Brava! session (either on or off).

Example: <code>EnhancedDisplaySetting=last</code></td></tr>
<tr><td>EnableVerifyR edactPanel</td><td>Value: <<i>TRUE|FALSE</i>>

Description: When <code>true</code>, Brava! displays the Redaction Verify panel in the right side of the viewing window. When <code>false</code>, the Verify panel does not display in the viewer, nor does its corresponding menu items and hotkeys.

Default: <code>EnableRedactVerifyPanel=TRUE</code></td></tr>
<tr><td>EnableBookma rkPanel</td><td>Value: <<i>TRUE|FALSE</i>>

Description: When <code>true</code>, Brava! displays the Bookmark panel in the right side of the viewing window. When <code>false</code>, the Bookmark panel does not display in the viewer, nor does its corresponding menu items and hotkeys.

Default: <code>EnableBookmarkPanel=TRUE</code></td></tr>
<tr><td>EnableTakeoff Panel</td><td>Value: <<i>TRUE|FALSE</i>>

Description: When <code>true</code>, Brava! displays the Measurement Takeoff panel in the right side of the viewing window. When <code>false</code>, the Takeoff panel does not display in the viewer, nor does its corresponding menu items and hotkeys.

Note: When <code>EnableMeasurement=false</code>, features are disabled within the measurement Takeoff panel regardless of the <i><code>EnableTakeoffPanel</code></i> setting value.

Default: <code>EnableTakeoffPanel=TRUE</code></td></tr>
</table>

EnableChangemarkPanel	**Value:** *<TRUE|FALSE>* **Description:** When `true`, Brava! displays the **Changemark** panel in the right side of the viewing window. When `false`, the **Changemark** panel does not display in the viewer, nor does its corresponding menu items and hotkeys. If *`AutoloadChangemarkPanel`* or *`AutoloadChangemark`* are set, the value of *`EnableChangemarkPanel`* is ignored. **Default:** `EnableChangemarkPanel=TRUE`
EnableCheckviewPanel	**Value:** *<TRUE|FALSE>* **Description:** When `true`, Brava! displays the **Checkview** panel in the right side of the viewing window. When `false`, the **Checkview** panel does not display in the viewer, nor does its corresponding menu items and hotkeys. If *`AutoloadCheckview`* is set, the value of *`EnableCheckviewPanel`* is ignored. **Default:** `EnableCheckviewPanel=TRUE`
EnableDisplayPanelBar	**Value:** *<TRUE|FALSE>* **Description:** When set to `false`, the narrow vertical bar used to show/hide the panels is not displayed. **Default:** `EnableDisplayPanelBar=TRUE`
EnableLayers	**Value:** *<TRUE|FALSE>* **Description:** Controls whether you can view CAD layers on a drawing file. `True` allows the viewing of layers. Use `false` to disallow the viewing of layers. If set to `false`, exporting to PDF/CSF publishes to a single (flat) layer. **Default:** `EnableLayers=TRUE`
EnableMagnifier	**Value:** *<TRUE|FALSE>* **Description:** Controls whether the **Magnifier** tool is available in Brava!. Use `false` to disable the **Magnifier** tool. **Default:** `EnableMagnifier=TRUE`
EnableMeasurement	**Value:** *<TRUE|FALSE>* **Description:** This parameter controls whether **Measurement** tools and capabilities are available in Brava!. `True` permits measurements to be created. Use `false` to disable measurement functionality. **Default:** `EnableMeasurement=TRUE`

EnableMirror	**Value:** <*TRUE*|*FALSE*> **Description:** Controls whether **Mirror** tool is available in Brava! from the right mouse button menu. Use `false` to disable the **Mirror** tool. **Default:** `EnableMirror=TRUE`
EnableMono	**Value:** <*TRUE*|*FALSE*> **Description:** Enables/disables the **Monochrome** tool button. Use `false` to disable the **Monochrome** tool. **Default:** `EnableMono=TRUE`
EnablePageChangeOnScroll	**Value:** <*TRUE*|*FALSE*> **Description:** Enables/disables the Brava! page change on scroll feature. Use `false` to disable page change when scrolling beyond the current page limits. **Default:** `EnablePageChangeOnScroll=TRUE`
EnablePrinting	**Value:** <*TRUE*|*FALSE*> **Description:** Controls whether printing is available in Brava!. Use `false` to disable printing. See also: *`disable_printing_for_these_formats`*, *`printing_control_for_users/groups`* **Default:** `EnablePrinting=TRUE`
EnableRightMouseButtonMenu	**Value:** <*TRUE*|*FALSE*> **Description:** When `false`, the menu that normally appears when the right mouse button is clicked (while the cursor is in the drawing/ document viewing window) is suppressed. **Default:** `EnableRightMouseButtonMenu=TRUE`
EnableThumbnailsResize	**Value:** <*TRUE*|*FALSE*> **Description:** Controls whether the page thumbnails can be resized. When `true`, the images can be resized by clicking the right mouse button (over the **Page Thumbnails** panel) and selecting **Reduce Thumbnail Size**, or **Increase Thumbnail Size** from the menu. If `false`, the right mouse menu is disabled and images remain at the default size. **Default:** `EnableThumbnailsResize=TRUE`
EnableRotate	**Value:** <*TRUE*|*FALSE*> **Description:** Enables/disables the **Rotate** toolbar button. **Default:** `EnableRoate=TRUE`

<table>
<tr><td>EnableUserPlackeepers</td><td>Value: <TRUE|FALSE>

Description: When set to <code>false</code>, all UI and functionality associated with PlaceKeepers (listed below) is removed.
Right Mouse Button or Marks Menu items:
Add Placekeeper
GoTo Placekeepers
Remove Added Placekeepers
Ctrl Keys:
Add PlaceKeeper :CTRL + K
GoTo PlaceKeepers :CTRL + ALT + K
Default: <code>EnableUserPlacekeepers=TRUE</code></td></tr>
<tr><td>DisableClipboardImages</td><td>Value: <TRUE|FALSE>

Description: Enables/disables copying images to the Windows Clipboard. If set to <code>true</code>, and the Viewer control holding this parameter has a print-screen-protected file open, Brava! attempts to remove any bitmap data contained in the Clipboard.

Default: <code>DisableClipboardImages=FALSE</code></td></tr>
<tr><td>LoadFitWidth</td><td>Value: <TRUE|FALSE>

Description: Controls setting the document at Fit Width zoom factor when initially loaded.

Default: <code>LoadFitWidth=TRUE</code></td></tr>
<tr><td>EnableZoomWindow</td><td>Value: <TRUE|FALSE>

Description: Controls whether the Zoom Window tool is available in Brava!. Use <code>false</code> to disable this feature.

Default: <code>EnableZoomWindow=TRUE</code></td></tr>
<tr><td>ShowExitButton</td><td>Value: <TRUE|FALSE>

Description: When <code>true</code>, an exit button appears in the upper right corner of the Brava! view screen. Intended for use when the viewer is integrated as a component embedded in a browser page.

Default: <code>ShowExitButton=FALSE</code></td></tr>
<tr><td>PrintMeasurementSystem</td><td>Value: <0|1>

Description: Controls the measurement system initially displayed in the Print dialog. Valid values are:
<code>0</code> = English
<code>1</code> = Metric
Example: <code>PrintMeasurementSystem=0</code></td></tr>
</table>

<table>
<tr>
<td>MeasurementSettings</td>
<td>Value: <SYS>|<UNITS>|<DEC>|<PREC>

Description: Controls the initial values of the measurement settings dialog. This is a string of four integers, separated by a pipe character.

• The first integer is the measurement system. Valid values are:
– 0 = ENGLISH_ARCHITECTURAL
– 1 = ENGLISH
– 2 = METRIC
– 3 = UNITLESS
• The second integer is the units. Valid values are:
– 0 = INCHES
– 1 = FEET
– 2 = METERS
– 3 = MILLIMETERS
– 4 = CENTIMETERS
– 5 = MILES
– 6 = KILOMETERS
– 7 = UNITLESS
– 8 = YARDS
• The third integer is the distance precision (number of decimal places to display). Valid values are:
– 0 = MEAS_PREC_1
– 1 = MEAS_PREC_2
– 2 = MEAS_PREC_3
– 3 = MEAS_PREC_4
– 4 = MEAS_PREC_5
– 5 = MEAS_PREC_6
• The fourth integer is the architectural distance precision (accuracy). Valid values are:
– 0 = ARCH_PREC_ONE
– 1 = ARCH_PREC_ONEHALF
– 2 = ARCH_PREC_ONEQUARTER
– 3 = ARCH_PREC_ONEEIGHTH
– 4 = ARCH_PREC_ONESIXTEENTH

If the format is incorrect or if the values are inconsistent (i.e. system is <code>English</code> and <code>Units</code> are meters), the parameter value is ignored.

Example: <code>MeasurementSettings=0|0|0|0</code></td>
</tr>
</table>

37.2.4 Markup and Redaction Parameters

<table>
<tr><td>AutoloadChangemark</td><td>Value: <code>GUID</code>

Description: This is a markup entity GUID. If a Changemark note having this GUID exists in the currently open document (i.e. a CSF with burned in markups) or any preloaded markup file(s), that Changemark note is “executed” upon load of the document.

Note: If this parameter is specified but the GUID doesn't exist in any preloaded markup, the Changemark panel is opened as if the <code>AutoloadChangemarkPanel</code> parameter has been set to true and override any startup panel set in <code>paneltodisplay</code>. If GUID is set to <code>suppress</code>, the Changemark is not loaded.

Example: <code>AutoloadChangemark=GUID</code></td></tr>
<tr><td>AutoloadCheckview</td><td>Value: <code>GUID</code>

Description: This is a markup entity GUID, similar to <code>AutoloadChangemark</code>. If it is a valid GUID string of an existing Checkview, the viewer activates that Checkview. If it is set to anything that is not a valid GUID and is not <code>void</code>, <code>AutoloadCheckview</code> activates the first Checkview in the document (if any exist). If GUID is set to <code>suppress</code>, the Checkview is not loaded

Example: <code>AutoloadCheckview=GUID</code></td></tr>
<tr><td>AutoloadChangemarkPanel</td><td>Value: <<i>TRUE|FALSE</i>>

Description: When <code>true</code>, and any Changemark notes exist in the currently open document or in any preloaded markup file(s), the Changemark panel is displayed upon load of the document. If the Changemark panel is auto loaded, the Changemark panel displays and overrides the panel set by the <i><code>paneltodisplay</code></i> setting.

Example: <code>AutoloadChangemarkPanel=FALSE</code></td></tr>
</table>

CustomTextHeights	**Value:** *<list of text heights to use, in inches>* **Description:** Defines the list of text heights available in the markup text heights list. If specified, this list replaces the default list of text heights available in Brava!. If unspecified, left blank, or is improperly formatted, the text heights menu is populated with the default list of options. The flexibility of this parameter allows heights to be defined in increments as small as one hundredth (.01). **Text annotation size:** Font size values are a scale factor relative to the actual size of the drawing or document currently being viewed. This scaling allows the same font size value to work similarly in documents that are 8.5 x 11 inches and drawings that may be 200 miles wide in world-space height. **Example:** `CustomTextHeights=0.01|0.05|0.1|0.2|0.3` Text heights passed in the parameter are delimited by the "|" symbol, so for example, the setting `CustomTextHeights=0.01|0.05|0.1|0.2|0.3` configures the text heights menu to present the options: 0.01, 0.05, 0.1, 0.2, and 0.3.
DefaultTextMarkupFont	**Value:** *<font name>* **Description:** Determines the default font to use for the markup text. Valid values are any font name on the Brava! Client machine operating system. You can view the available font names from the font list of the markup text properties bar with markup **Text** as the active tool. **Example:** `DefaultTextMarkupFont=Arial`
EnableConsolidateIfMarkupOpenedForEdit	**Value:** *<TRUE|FALSE>* **Description:** If set to `false`, this parameter disables the **Consolidate Markups** item on the **Markup** menu if a markup is open for edit, regardless of the *`EnableMarkupConsolidate`* value. If set to `true`, the *`EnableMarkupConsolidate`* value must also be true, or this parameter value is ignored. **Example:** `EnableConsolidateIfMarkupOpenedForEdit=FALSE`
EnableCopyMarkupToAllPages	**Value:** *<TRUE|FALSE>* **Description:** If set to `true`, and a multi-page document is being viewed, the **Copy Entities to Designated Pages** button appears (which prompts an option dialog for copying) when one or more copy able markup entities are selected. Entities that cannot be copied include edit text entities and Changemark notes. **Note:** Altering the markup entities after they have been copied to all pages does not affect the copied entities. **Example:** `EnableCopyMarkupToAllPages=FALSE`

<table>
<tr><td>EnableMarkup</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Controls whether markup capabilities are available in Brava!. <code>True</code> permits markups to be created. Use <code>false</code> to disable markups.

Example: <code>EnableMarkup=True</code></td></tr>
<tr><td>EnableMarkup Close</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Enables/disables the Close item in the Markup menu.

Example: <code>EnableMarkupClose=True</code></td></tr>
<tr><td>EnableMarkup CloseForRevie w</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Enables/disables the Close Review item in the Markup menu.

Example: <code>EnableMarkupCloseForReview=True</code></td></tr>
<tr><td>EnableMarkup ColorPalette</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Enables/disables the color palette for markups. If <code>false</code>, the user can only use the default color to create markup entities.

Example: <code>EnableMarkupColorPalette=True</code></td></tr>
<tr><td>EnableMarkup ChangeOwner</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Enables/disables the ability to transfer ownership of a markup entity. If <code>true</code>, when a user shift-clicks on a markup entity (in a markup open for edit) using the markup Select tool, the author information of that entity changes to the current author.

Example: <code>EnableMarkupChangeOwner=False</code></td></tr>
<tr><td>EnableMarkup Consolidate</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Enables/disables the Consolidate Markups item on the Markup menu if at least two markups are open (for edit or for review).

Example: <code>EnableMarkupConsolidate=True</code></td></tr>
<tr><td>EnableMarkup FontAutosize</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Controls whether the markup text is resized when the bounding box changes height. When <code>true</code>, text does not resize and remains at the last text size that was established. When <code>false</code>, the markup text size gets larger as the bounding box is resized taller.

Example: <code>EnableMarkupFontAutosize=True</code></td></tr>
</table>

EnableMarkup <*toolname*>Tool	**Value:** <*TRUE|FALSE*> The following parameters enable/disable specific toolbar buttons from the **Markup** toolbar. While the default value for all `EnableMarkup<toolname>Tool` parameters is `true`, a value of `false` disables the tool or group of tools. **EnableMarkupAbcGroupTool** Enables/disables the ABC text group of markup tools (**Strikeout**, **Highlight**, and **Underline**). **EnableMarkupArrowTool** Enables/disables the **Arrow** markup tool. **EnableMarkupBlockoutTool** Enables/disables the **Redact Area** markup tool. **EnableMarkupBlockoutCarverTool** Enables/disables the **Allow Area** markup tool. **EnableMarkupChangemarkTool** Enables/disables the **Changemark** markup tool. **EnableMarkupCloudGroupTool** Enables/disables the **Cloud** group of the markup tool (**Cloud** and **Polygon Cloud**). **EnableMarkupHyperlinkTool** Enables/disables the **Hyperlink** markup tool. **EnableMarkupLinearGroupTool** Enables/disables the linear group of markup tools (**Line**, **Polyline**, **Arc**, **Crossout**, **Arrrow Pointer**, and **Scratchout**). **EnableMarkupRasterTool** Enables/disables the **Raster** markup tool. The markup status bar also contains a browse button allowing the user to select a raster image for insertion into the markup layer. The browse button does not display if the `MarkupRasterFile` parameter is set. **EnableMarkupShapeGroupTool** Enables/disables the drawing shapes group of markup tools (**Rectangle**, **Rounded rectangle**, **Polygon**, and **Ellipse**). **EnableMarkupSketchGroupTool** Enables/disables the sketch group of markup tools (**Sketch** and **Sketch Polygon**). **EnableMarkupStampTool** Enables/disables the **Stamp** markup tool. **EnableMarkupTextTool** Enables/disables the **Text** markup tool. **EnableMarkupSelectTool** Enables/disables the **Select** markup tool. **Example:** `EnableMarkup<toolname>Tool=True`

MarkupToolbarType	**Value:** *<0-3>* **Description:** Sets which markup toolbar displays by default (when `preloadmarkupname` is set). Valid values are: • `0` = Standard • `1` = Redaction • `2` = Measure • `3` = Measure Takeoff **Example:** `MarkupToolbarType=0`
MarkupBlockoutColor	**Value:** *<color name or hex RGB value>* **Description:** Use to specify the default redaction entity color. Valid values are the word descriptions of the color, or the hexadecimal RGB number for red, green, and blue components #RRGGBB. Please see the list of valid colors listed for *`MarkupColor`*. **Example:** `MarkupBlockoutColor=#000000`
MarkupTextBackgroundColor	**Value:** *<color name or hex RGB value>* **Description:** Use to specify the default markup text background color. Accepts the same syntax values as the *`MarkupColor`* parameter. **Example:** `MarkupTextBackgroundColor=white`

MarkupColor	**Value:** <*color name or hex RGB value*> **Description:** Use to specify the default markup color. Valid values are the word descriptions of the (Web Safe) color, or the hexadecimal RGB number for red, green, and blue components #RRGGBB. Below is the list of valid colors. • White : #FFFFFF • Yellow : #FFFF00 • Magenta : #FF00FF • Red : #FF0000 • Cyan : #00FFFF • Green : #00FF00 • Blue : #0000FF • DarkGrey : #666666 • Grey : #999999 • Olive : #666600 • Purple : #660066 • Maroon : #660000 • Teal : #006666 • Pine : #006600 • DarkBlue : #000066 • Black : #000000 **Example:** `MarkupColor=black` or `MarkupColor=#000000`
DisableRedactionBorder	**Value:** <*TRUE|FALSE*> Enables/disables the drawing of a thin border around redaction areas. The setting applies to all redactions drawn. **Example:** `DisableRedactionBorder=False`
MarkupRaster Directory	**Value:** <*folder path*> Defines the default path for the raster images used with the **Insert Image** markup tool. If set, when a user clicks the **Browse** button on the markup properties bar, an **Open Image** dialog displays to the directory set in this parameter. Users can select PNG and JPG raster files to be placed and sized within a Brava! markup layer. This parameter overrides the last directory browsed preference saved for the user. It is recommended that you use a path relative to and within the user's profile. The path starting point of `%USERPROFILE%\\` should help to ensure that the user has the proper permissions to the raster image files. **Example:** `Markuprasterdirectory=C:\Documents and Settings \<user_name>\IGC\StampImages`

MarkupTextBackgroundType	**Value:** <*TRANSPARENT|BACKGROUNDFILL|SOLID*> Defines the type of background for the markup text. Valid values are `Transparent`, `BackgroundFill` (Match Display Background Color), and `Solid` (custom color selection). **Example:** `MarkupTextBackgroundType=solid`
RedactionScriptsDirectory	**Value:** <*folder name(s)*> If a directory path value for this parameter is set, any redaction script files (.xrs) contained in this directory display in the Brava! **Redact Using Scripts** dialog for selection when a user clicks on the **Redact using script(s)/List(s)** button. This parameter overrides the last directory browsed preference that was saved for the users. A directory of sample files contained in `RedactionScripts.bin` can be downloaded from the web server via the *`download.url.RedactionScripts`* parameter and placed on Brava! Clients. **Example:** `RedactionScriptsDirectory=%USERPROFILE%\\IGC\\x`*`<version>`*`\\RedactionScripts`
TakeOwnershipOnMarkupConsolidation	**Value:** <*TRUE|FALSE*> If `true`, the markup consolidation action changes the ownership of ALL consolidate markups entities associated with the document or drawing to the current user. **Example:** `TakeOwnershipOnMarkupConsolidation=False`
UIEditRedlineText	**Value:** <*TRUE|FALSE*> If `true`, when a user edits or creates a new entity with the markup **Text** tool, a dialog box containing an edit box appears. The user can edit the markup text directly through the dialog. If the user's system is using a right to left language (such as Japanese or Arabic), the edit box allows right to left editing. If set to `false`, the text edit box is not available and editing/creating right to left markup text is not possible. **Example:** `UIEditRedlineText=False`
Add_These_Groups_To_Redacted_Rendition_View	**Value:** <*Group Names*> These groups will have a redacted view of a document based on the script configuration. **Example:** `Add_These_Groups_To_Redacted_Rendition_View=group1,group2`
Add_These_Users_To_Redacted_Rendition_View	**Value:** <*Group Names*> These users will have a redacted view of a document based on the script configuration. **Example:** `Add_These_Users_To_Redacted_Rendition_View=user1,user2`

Html_Redaction_Scripts_Path	**Value:** *<folder name(s)>* Stores the redaction scripts that Brava! uses for role-based redaction on load of document in the Brava! HTML Viewer. These scripts decide what part of a document is redacted for a particular set of users or groups. Scripts are placed in this server location and not on a client. It supports a path relative to the deployed directory. `D2/Igc/config` is the default directory. **Example:** `Html_Redaction_Scripts_Path=D2/Igc/config`

37.2.5 Publishing Parameters

37.2.5.1 Output Options

EnableSaveView	**Value:** *<TRUE|FALSE>* Enables/disables the **Publish** button from the toolbar. If `false`, the values of *`EnableCSFPublish`*, *`EnableSaveViewAsJPG`*, *`EnablePDFPublish`*, and *`EnableTIFFPublish`* are ignored since the submenu of the toolbar button is not available. **Example:** `EnableSaveView=TRUE`

37.2.5.2 PDF Export

PDFExportOutputChangemarkInfo	**Value:** *<TRUE|FALSE>* If `true` any Changemarks existing on the current document/drawing have their Changemark information appended to the end of the published file. *`PDFExportOutputChangemarkInfo`* affects the UI state of the **Append Changemarks notes** check box in the **PDF Publish Options** dialog. **Example:** `PDFExportOutputChangemarkInfo=FALSE`

37.2.5.3 JPG Export

EnableSaveViewAsJPG	**Value:** *<TRUE|FALSE>* Enables/disables ability for viewers to create JPG snapshots of the currently viewed CSF file. Default is `true` to allow capturing JPG snapshots. If *`EnableSaveView`* is `false`, this setting is ignored. **Example:** `EnableSaveViewAsJPG=TRUE`
SaveViewQuality	**Value:** *<percentage from 1 to 100>* Sets the amount of JPG compression to be used when the **Save View as JPG** option is run. The value is entered as a percentage and can range from 1 (poorest quality) to 100 (highest quality). **Example:** `SaveViewQuality=75`

37.3 HTML Client optional parameters

The parameters in this section are used with the Brava! HTML Client only.

<table>
<tr><td>HTMLViewer Width</td><td>Value: <percentage>

Specifies the width percentage of the Brava! HTML Client.

Example: <code>HTMLViewerWidth=100%</code></td></tr>
<tr><td>HTMLViewerHeight</td><td>Value: <percentage>

Specifies the height percentage of the Brava! HTML Client.

Example: <code>HTMLViewerHeight=100%</code></td></tr>
<tr><td>Html.viewer.backgroundcolor</td><td>Value: <color name or hex RGB value>

Specifies the default Brava! HTML Client background color, expressed as either a hexadecimal RGB number, or as a standard HML color label.

Example: <code>Html.viewer.backgroundcolor=#ff0000</code> or <code>Html.viewer.backgroundcolor=purple</code></td></tr>
<tr><td>html.searchFeature.<<i>option</i>></td><td>Value: <i><TRUE|FALSE></i>

Enables/disables the specified Search features.

<code>html.searchFeature.Enabled=TRUE</code>

<code>html.searchFeature.EnablePresets=TRUE</code>

<code>html.searchFeature.EnableTermHits=TRUE</code>

<code>html.searchFeature.SearchCap=100</code></td></tr>
<tr><td>html.searchFeature.DefaultTermHits</td><td>Value: <TRUE|FALSE>

Determines whether the Brava! HTML Viewer initially performs term hit highlighting. Corresponds to the Brava! API method <code>DEFAULT_TERMHITS</code>.

Defaults: <code>html.searchFeature.DefaultTermHits=TRUE</code></td></tr>
<tr><td>html.searchFeature.IgnoreTermHitsScale</td><td>Value: <TRUE|FALSE>

This optional setting determines whether to change scale while navigating to term hits in the Brava! HTML Viewer. Corresponds to the Brava! API method <code>IGNORE_TERMHITS_SCALE</code>. When <code>False</code>, scaling is active.

Defaults: <code>html.searchFeature.IgnoreTermHitsScale=TRUE</code></td></tr>
<tr><td>html.searchFeature.MatchCase</td><td>Value: <TRUE|FALSE>

Determines the default setting for the Match Case item of the Brava! HTML Viewer Search list. Corresponds to the Brava API method <code>MATCH_CASE</code>.

Defaults: <code>html.searchFeature.MatchCase=TRUE</code></td></tr>
</table>

<table>
<tr><td>html.searchFeature.SetSearchPlaceHolderEmpty</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

By default, if a user performs a search and then opens a file in the viewer, the search field and the search terms display in the viewer. Set this to <code>true</code> to leave the search field empty.

Defaults: <code>html.searchFeature.SetSearchPlaceHolderEmpty=FALSE</code></td></tr>
<tr><td>html.searchFeature.ShowDialog</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Determines the default setting for the Show Dialog item of the Brava! HTML Viewer Search list. If <code>true</code>, the Search Results dialog is automatically shown after a search is run. Corresponds to the Brava! API method <code>SHOW_DIALOG</code>.

Defaults: <code>html.searchFeature.ShowDialog=TRUE</code></td></tr>
<tr><td>html.searchFeature.WholeWord</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Determines the default setting for the Whole Word item of the Brava! HTML Viewer Search list. Corresponds to the Brava! API method <code>WHOLE_WORD</code>.

Defaults: <code>html.searchFeature.WholeWord=TRUE</code></td></tr>
<tr><td>html.markupsFeature.<<i>option</i>></td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Enables/disables the specified markup edit and review features.

Defaults: <code>html.markupsFeature.EnableEdit=TRUE</code>

<code>html.markupsFeature.EnableReview=TRUE</code>

<code>html.markupsFeature.EnableChangemarkReplies=TRUE</code>

<code>html.markupsFeature.EnablecloseReview=TRUE</code></td></tr>
<tr><td>html.markupsFeature.EnableMarkupOverwrite</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Determines if existing markups are overwritten in the Documentum CM Server when markup save requests are received. If <code>false</code>, new markups are created. If <code>true</code>, existing markups are overwritten.

Default: <code>html.markupsFeature.EnableMarkupOverwrite=TRUE</code></td></tr>
<tr><td>html.markupsFeature.EnableDefaultVersion</td><td>Value: <<i>TRUE</i>|<i>FALSE</i>>

Selects or deselects the Save All Version check box option in the markup Save as dialog of the HTML viewer. <code>True</code> is selected, and <code>false</code> is deselected. This setting is applicable to both the Classic and Smart View clients.

Default: <code>html.markupsFeature.EnableDefaultVersion=FALSE</code></td></tr>
</table>

<table>
<tr><td>html.markupsFeature.EnablePrivate</td><td>Value: <TRUE|FALSE>

Enables/disables the display of the Save Private Markup check box option in the markup Save as dialog of the HTML viewers. <code>True</code> is on, and <code>false</code> is off.

Default: <code>html.markupsFeature.EnablePrivate=TRUE</code></td></tr>
<tr><td>html.markupsFeature.EnableDefaultPrivate</td><td>Value: <TRUE|FALSE>

Selects or deselects the default selection of the Save Private Markup check box option in the markup Save as dialog of the HTML viewer. <code>True</code> is selected, and <code>false</code> is deselected. This setting is applicable to both the Classic and Smart View clients.

Default: <code>html.markupsFeature.EnableDefaultPrivate=FALSE</code></td></tr>
<tr><td>html.markupsFeature.MarkupSaveDescription</td><td>Value: <TRUE|FALSE>

Changes the “Markup Name” string within the HTML viewer’s Markup > Save As dialog to the string “Markup Description” and disables the dialog’s Save with all Versions and Save Markup as Private options. This setting is applicable to both the Classic and Smart View clients.

Default: <code>html.markupsFeature.MarkupSaveDescription=FALSE</code></td></tr>
<tr><td>html.markupsFeature.SYMBOL_CONFIG_URI</td><td>Value: <URI path>

The URI to your symbol webservice. If left blank (the default), the Symbols annotation button is hidden in the Brava! HTML Client.

Default: <code>html.markupsFeature.SYMBOL_CONFIG_URI=</code></td></tr>
<tr><td>html.markupsFeature.EnableMarkupStampTool</td><td>Value: <TRUE|FALSE>

Enables/disables the display of the Stamp tool selection in the HTML viewer. <code>True</code> is on, and <code>false</code> is off. This setting is applicable to both the Classic and Smart View clients.

Default: <code>html.markupsFeature.EnableMarkupStampTool=TRUE</code></td></tr>
<tr><td>html.markupMenu.Enable<<i>option</i>></td><td>Value: <i><TRUE|FALSE></i>

These parameters specify which markup menu items (New, Open, Save, Save As, Close, and Consolidate) are on (<code>true</code>) or off (<code>false</code>) in the Brava! HTML Client.

Defaults: <code>html.markupMenu.EnableNew=TRUE</code>

<code>html.markupMenu.EnableOpen=TRUE</code>

<code>html.markupMenu.EnableSave=TRUE</code>

<code>html.markupMenu.EnableSaveAs=TRUE</code>

<code>html.markupMenu.EnableClose=TRUE</code>

<code>html.markupMenu.EnableConsolidate=TRUE</code></td></tr>
</table>

<table>
<tr><td>html.enable.Markup.changeOwner</td><td>Value: <<i>TRUE|FALSE</i>>

Enables/disables the ability to take ownership of any markup entities that are shift-clicked during a markup consolidation.

Default: <code>html.enable.Markup.changeOwner=FALSE</code></td></tr>
<tr><td>html.enable.Markup.TakeOwnershipOnConsolidate</td><td>Value: <<i>TRUE|FALSE</i>>

If <code>true</code>, the markup consolidation action on HTML Viewer changes the ownership of all consolidate markups entities associated with the document or drawing to the current user.

Example: <code>html.enable.Markup.TakeOwnershipOnConsolidate=TRUE</code></td></tr>
<tr><td>html.markup.enable.consolidate.changeownership.with.higher.permission.than</td><td>Value: <<i>Numeric Permission Level</i>>

Enables Markup Consolidate and ChangeOwnership for users only having higher permissions than the provided value. Accepts values <code>3</code> or <code>4</code> or <code>5</code> or <code>6</code> for permissions <code>DF_PERMIT_READ,DF_PERMIT_RELATE,DF_PERMIT_VERSION,DF_PERMIT_WRITE</code> respectively.

Example: <code>html.markup.enable.consolidate.changeownership.with.higher.permission.than=4</code></td></tr>
<tr><td>html.printFeature.default.Grayscale</td><td>Value: <<i>TRUE|FALSE</i>>

Turns on/off default printing of grayscale in the HTML viewers. <code>true</code> selects the Grayscale option check box in the print options dialog, and <code>false</code> clears the check box.

Default: <code>html.printFeature.default.Grayscale=FALSE</code></td></tr>
<tr><td>html.publishFeature.Enabled</td><td>Value: <<i>TRUE|FALSE</i>>

Enables/disables publishing/exporting in the HTML viewers. <code>true</code> is on, and <code>false</code> is off.

Default: <code>html.publishFeature.Enabled=TRUE</code></td></tr>
<tr><td>html.publishFeature.EnablePDF</td><td>Value: <<i>TRUE|FALSE</i>>

Enables/disables publishing/exporting to PDF in the HTML viewers.

Default: <code>html.publishFeature.EnablePDF=TRUE</code></td></tr>
<tr><td>html.publishFeature.EnableTIF</td><td>Value: <<i>TRUE|FALSE</i>>

Enables/disables publishing/exporting to TIF in the HTML viewers.

Default: <code>html.publishFeature.EnableTIF=TRUE</code></td></tr>
</table>

html.publishFeature.ColorMode	**Value:** <*black|grayscale|original*> Sets the default Color Mode selection on the HTML viewer **Publish to TIFF** and **Publish to PDF** options dialogs. A value of `black` sets **Lines as Black**, `original` (or the default of blank) sets **Original Colors**, and `grayscale` sets **Grayscale**. **Default:** `html.publishFeature.ColorMode=`
docmerge.publishFeature.colorMode	**Value:** <*fullcolor, grayscale, monochrome*> Decides colorMode to be used while publishing on docmerge viewer. Once set, docmerge viewer will always publish the merged document using this specified color mode. Default is fullcolor if not set. **Default:** `fullcolor`
html.publishFeature.OverrideColorModeUserSetting	**Value:** <*TRUE|FALSE*> If `true`, stored Color Mode preferences for the HTML viewer **Publish to TIFF** and **Publish to PDF** options dialogs are ignored and the value specified in `html.publishFeature.ColorMode` is used. If `false`, user preferences are retained for each viewer launch. **Default:** `html.publishFeature.OverrideColorModeUserSetting=FALSE`
html.publishFeature.EnableUserControls	**Value:** <*TRUE|FALSE*> When `true` this property disables the **Publish to TIF** and **Publish to PDF** menu options for any users who are NOT listed in the *`html.publish.local`* or *`html.publish.vault`* properties. **Default:** `html.publishFeature.EnableUserControls=FALSE`
html.publish.local.users	**Value:** <*usernames*> A list of individual usernames that are permitted to use the Brava! **Publish to TIF** and **Publish to PDF** menu options to publish content to their local file system. Requires `html.publishFeature.EnableUserControls=TRUE` AND `html.publishFeature.Enabled=TRUE`. **Example:** `html.publish.local.users=user1,user2,user3`
html.publish.local.groups	**Value:** <*group names*> A list of group names that are permitted to use the Brava! **Publish to TIF** and **Publish to PDF** menu options to publish content to their local file system. Requires `html.publishFeature.EnableUserControls=TRUE` AND `html.publishFeature.Enabled=TRUE`. **Example:** `html.publish.local.groups=group1,group2,group3`

html.publish.vault.users	**Value:** <*usernames*> A list of individual usernames that are permitted to use the Brava! **Publish to TIF** and **Publish to PDF** menu options to publish content to the OpenText Documentum CM repository (to the users home cabinet or to a new Major or Minor version of the document). Requires `html.publishFeature.EnableUserControls=TRUE` AND `html.publishFeature.Enabled=TRUE`. **Example:** `html.publish.vault.users=user1,user2,user3`
html.publish.vault.groups	**Value:** <*group names*> A list of group names that are permitted to use the Brava! **Publish to TIF** and **Publish to PDF** menu options to publish content to the OpenText Documentum CM repository (to the users home cabinet or to a new Major or Minor version of the document). Requires `html.publishFeature.EnableUserControls=TRUE` AND `html.publishFeature.Enabled=TRUE`. **Example:** `html.publish.vault.groups=group1,group2,group3`
html.viewmodeFeature.Initialzoom	**Value:** <*extents|width*> Specifies the initial zoom mode set in the HTML viewer. **Default:** `html.viewmodeFeature.InitialZoom=extents`
html.viewmodeFeature.Grayscale	**Value:** <*TRUE|FALSE*> Enabling this will default the grayscale view mode for the HTML viewer. **Default:** `html.viewmodeFeature.Grayscale=true`
html.viewmodeFeature.default.BlackandWhite	**Value:** <*TRUE|FALSE*> Enabling this will default the display black and white view mode for HTML viewer. **Default:** `html.viewmodeFeature.default.BlackAndWhite=true`
html.viewer.BOOKMARK_CONTROLS	**Value:** <*TRUE|FALSE*> Specifies whether the **Bookmark** panel is enabled in the HTML viewer. **Default:** `html.viewer.BOOKMARK_CONTROLS=TRUE`
html.viewer.Thumbnail_CONTROLS	**Value:** <*TRUE|FALSE*> Specifies whether the **Thumbnail** panel is enabled in the HTML viewer. **Default:** `html.viewer.Thumbnail_Controls=TRUE`

UsePDFRendition	**Value:** <*TRUE|FALSE*> If `true`, Brava! (ActiveX and HTML Client) uses the PDF Rendition of a file if available. If `false` Brava! uses the original file. This setting is ignored (disabled) if launching Brava! through the renditions screen. See also *`PDF_Rendition_Control_For_Groups`* and *`PDF_Rendition_Control_For_Users`* **Default:** `UsePDFRendition=FALSE`
html.max.cache.size.file.not.cached	**Value:** <*whole number*> Determines the maximum number of pages held in the HTML viewer's memory when the file is not already cached on the Brava! Server. **Example:** `html.max.cache.size.file.not.cached=8`
html.number.preloaded.pages.file.not.cached	**Value:** <*whole number*> Determines the number of pages automatically loaded in the HTML viewer when first loading a document when the file is not already cached on the Brava! Server. **Example:** `html.number.preloaded.pages.file.not.cached=3`
html.showInSimpleMode	**Value:** <*TRUE|FALSE*> Use `true` to display the HTML viewer in simple mode. **Default:** `html.showInSimpleMode=FALSE`
html.showThemeInLightColor	**Value:** <*TRUE|FALSE*> Use `true` to display the HTML viewer in light theme mode. **Default:** `html.showThemeInLightColor=FALSE`
html.textCompare.formats	**Value:** <*comma delimited list of extensions*> Defines the document formats to launch, by default, using the HTML Text Compare viewer. This setting is used with the Brava! Viewer and Brava! Compare widgets of the application. The format extension of both documents selected for comparison must be listed here to launch the HTML Text Compare viewer. Formats not listed here launch using the graphical compare viewers (either the HTML Compare viewer or ActiveX Compare viewer, as determined by *`DefaultViewer`*). **Example:** `Html.textCompare.formats=pdf,doc,docx,rtf,txt`
html.target.new.window	**Value:** <*TRUE|FALSE*> When set to `TRUE`, allows the Brava! HTML Viewer to open in a new browser window. If set to the default, `FALSE`, the viewer will open in the standard venue, for example, a widget. **Example:** `Html.target.new.window=TRUE`

<table>
<tr><td>html.enable.copy.text</td><td>Value: <<i>TRUE|FALSE</i>>

When set to TRUE, allows the Brava! HTML Viewer user to copy text from the viewer. The default is TRUE. If set to FALSE (or no value), copy text capabilities are disabled.

Example: <code>Html.enable.copy.text=TRUE</code></td></tr>
<tr><td>html.enable.zoom.shortcut</td><td>Value: <<i>TRUE|FALSE</i>>

When set to TRUE, allows the Brava! HTML Viewer user to use the keyboard shortcuts CTRL + to zoom in and CTRL - to zoom out. The native select tools are disabled when this parameter is set as true and native select tools related to this function are inoperable. The default is FALSE. If set to FALSE (or no value), the keyboard shortcuts are disabled.

Example: <code>Html.enable.zoom.shortcut=TRUE</code></td></tr>
<tr><td>html.showInDesktopMode</td><td>Value: <<i>TRUE|FALSE</i>>

This is an option to enable or disable desktop ActiveX- style view. Set to <code>false</code> by default, the document will be viewed in SMART view mode. Enable Desktop view by setting the flag as <code>true</code>. Desktop View variant is set to HTML viewer configuration based on this flag.

Example: <code>html.showInDesktopMode=FALSE</code></td></tr>
<tr><td>html.markup.default.font.size</td><td>Value: <code>Number between 0.1 and 10</code>, default <code>0.3</code>

This is an option to customize the default text size of annotations in Brava! viewers. Set <code>html.markup.default.font.size</code> to a number between <code>0.1</code> and <code>10</code>. By default, the value is set to <code>0.3</code>.

Example: <code>html.markup.default.font.size=0.3</code>

Tip: For many users, setting the default value to 0.2 can help them fit more text in an annotation without making the text too small to read comfortably.</td></tr>
<tr><td>default_markup_tool</td><td>Value: <<i>Annotations, Changemarks, Lines, Redactions, Stamps, Text Editing</i>>

This is an option to customize the default tool in Brava! viewers.

The default tool is <code>annotations</code> and the list is ordered alphabetically. To change the default tool, uncomment the <i><code>default_markup_tool</code></i> parameter and enter the name of the tool you want set as the default. The tool names are listed in the <code>brava_parameter.properties</code> file above the <i><code>default_markup_tool</code></i> parameter. When you set a custom default tool, the remaining tools are still listed alphabetically.

Example: <code>default_markup_tool=annotations</code></td></tr>
</table>

37.3.1 Disabling HTML Client Markup and Redaction Tools

Administrators can disable any of the HTML 2D viewer's markup tools by adding the following parameters to their `brava_parameters.properties` file. These options correspond with the Brava! Enterprise API configuration parameters found in the `com.igc.be.api.client.ui.MarkupTool` and `com.igc.be.api.client.ui.RedactionTool` enumerations of the Brava! Enterprise API (see the Brava! SDK documentation API JavaDocs).

By default, all markup and redaction tools are available in the Brava! HTML Viewer without these configuration parameters appearing in the `brava_parameters.properties` file. Administrators need only add the tool's “enable = false” parameter(s) from the following list that they want to remove from the Viewer configuration:

Markup tool parameter	Description
`html.markupTools.Enabled.ARROW_CHANGEMARK=FALSE`	Disables the **Changemark Arrow** tool button.
`html.markupTools.Enabled.ARROW_LINE=FALSE`	Disables the **Arrow** tool button.
`html.markupTools.Enabled.BOOKMARK=FALSE`	Disables the **Bookmark** tool button.
`html.markupTools.Enabled.CHANGEMARK=FALSE`	Disables the **Changemark** tool button.
`html.markupTools.Enabled.CHECKVIEW=FALSE`	Disables the **Checkview** tool button.
`html.markupTools.Enabled.CLOUD=FALSE`	Disables the **Cloud** tool button.
`html.markupTools.Enabled.CLOUD_CHANGEMARK=FALSE`	Disables the **Changemark Cloud** tool button.
`html.markupTools.Enabled.CROSSOUT=FALSE`	Disables the **Crossout** tool button.
`html.markupTools.Enabled.ELLIPSE=FALSE`	Disables the **Ellipse** tool button.
`html.markupTools.Enabled.HIGHLIGHT=FALSE`	Disables the **Highlight** tool button.
`html.markupTools.Enabled.HIGHLIGHT_CHANGEMARK=FALSE`	Disables the **Changemark Highlight** tool button.
`html.markupTools.Enabled.HIGHLIGHT_TEXT=FALSE`	Disables the **Text Highlight** tool button.
`html.markupTools.Enabled.HIGHLIGHT_TEXT_CHANGEMARK=FALSE`	Disables the **Changemark Text Highlight** tool button.
`html.markupTools.Enabled.IMAGE=FALSE`	Disables the **Image** tool button.

Markup tool parameter	Description
html.markupTools.Enabled.LINE= FALSE	Disables the **Line** tool button.
html.markupTools.Enabled.POLYGON= FALSE	Disables the **Polygon** tool button.
html.markupTools.Enabled. POLYLINE=FALSE	Disables the **Polyline** tool button.
html.markupTools.Enabled. RECTANGLE=FALSE	Disables the **Rectangle** tool button.
html.markupTools.Enabled.ROUNDED_ RECTANGLE=FALSE	Disables the **Rounded Rectangle** tool button.
html.markupTools.Enabled. SCRATCHOUT=FALSE	Disables the **Scratchout** tool button.
html.markupTools.Enabled.SKETCH= FALSE	Disables the **Sketch** tool button.
html.markupTools.Enabled.SKETCH_ POLYGON=FALSE	Disables the **Sketch Polygon** tool button.
html.markupTools.Enabled. SQUIGGLE=FALSE	Disables the **Scratchout** tool button.
html.markupTools.Enabled.STAMP= FALSE	Disables the **Stamp** tool button.
html.markupTools.Enabled. STRIKEOUT=FALSE	Disables the text **Strikeout** tool button.
html.markupTools.Enabled. STRIKETHROUGH=FALSE	Disables the text **Strikethrough** tool button.
html.markupTools.Enabled. STRIKETHROUGH_CHANGEMARK=FALSE	Disables the **Changemark Text Strikethrough** tool button.
html.markupTools.Enabled.SYMBOL= FALSE	Disables the **CAD Symbol** tool button.
html.markupTools.Enabled.TEXT= FALSE	Disables the **Text** tool button.
html.markupTools.Enabled. UNDERLINE=FALSE	Disables the text **Underline** tool button.

Redaction tool parameter	Description
html.redactionTools.Enabled. POLYGON=FALSE	Disables the **Redact Polygon** tool button.
html.redactionTools.Enabled.FROM_ TO=FALSE	Disables the **Find & Redact From/To** tool button.
html.redactionTools.Enabled.AREA= FALSE	Disables the **Redact Area** tool button.

Redaction tool parameter	Description
`html.redactionTools.Enabled.SCRIPT=FALSE`	Disables the **Redact by Script** tool button.
`html.redactionTools.Enabled.TEXT=FALSE`	Disables the **Redact Text** tool button.
`html.redactionTools.Enabled.FIND_AND_REDACT=FALSE`	Disables the **Find and Redact** tool button.
`html.redactionTools.Enabled.PEEK_POLYGON=FALSE`	Disables the **Allow Polygon Area** tool button.
`html.redactionTools.Enabled.PAGE=FALSE`	Disables the **Redact Page** tool button.
`html.redactionTools.Enabled.PEEK_THROUGH=FALSE`	Disables the **Allow Area** tool button.

Example 37-1: Disable tool example

By adding `html.markupTools.Enabled.IMAGE=FALSE` to the `brava_parameters.properties` file, the HTML Viewer **Image** markup tool and palette are not loaded in the HTML Viewer.

37.3.2 HTML Video Parameters

Video parameters can be used to control feature availability in the Brava! HTML Video Client.

SupportedVideoFormats	**Value:** *<comma delimited list of extensions>* Specifies which video formats are supported in the HTML Video Client. **Example:** `SupportedVideoFormats=FLV,F4V,MOV,MP4,264,m4v`

37.4 Integration Specific Parameters

The settings throughout this section are used for both the Brava! ActiveX and HTML viewers, except where noted.

The Integration parameters in this section are only used when Brava! Enterprise is integrated with OpenText Documentum CM.

Follow the steps below to setup `brava_parameters.properties` to fit your configuration.

Note: The Brava! Enterprise install must be complete prior to customizing the Brava! Server parameters via the `brava_parameters.properties` file. In the Brava! install steps of the *OpenText Documentum Content Management Installation Guide*, you copied the contents of the version-appropriate Custom Directory that contains the integration components into the custom directory. It is very important that you modify the deployed `brava_parameters.properties` file and not the default file located in the Brava! Server install location. Modifying the un-deployed copy results in no changes taking place to the active deployed Brava! configuration. Some application servers may require a restart or redeployment the Web Application before the changes in the `brava_parameters.properties` takes effect.

1. Using a text editor such as Notepad, open `brava_parameters.properties`, located in the deployed directory

   `<drive>: \<jakarta install directory>\webapps\D2` or `\bea\user_projects\dctm\applications\D2`

2. Parameters contained in the section below must be edited with the correct parameter values as dictated by your specific environment and needs.

3. Save and close the .properties file.

The following tables list the specific parameters (and possible values) contained in the `brava_parameters.properties` page that you can customize for the Brava! viewers. Many of these parameters determine which features are available to the viewers. You can choose to enable or disable a feature by setting its value to TRUE or FALSE.

Note: Blank values for parameters cause the BravaX Client to use the default value in cases were values are defined values other then TRUE or FALSE, or where a blank value has the effect of turning the parameter off.

37.4.1 System Configuration Parameters

DefaultViewer	**Value:** *<ACTIVEX|HTML>* Determines which type of viewer is shown to the user (ActiveX or HTML). If set to activex, the Brava! ActiveX Client is used as the default Brava! Viewer. If the client browser is not Internet Explorer, Brava! HTML Viewer is used regardless of this setting. **Default:** `DefaultViewer=HTML`

37.4.2 Custom Parameters

The following custom integration parameters are set according to your specific environment and requirements. These settings apply to both the ActiveX and HTML viewers, excluding the settings listed in the ActiveX section.

ServerHostName	**Value:** ServerHostName=$PROTOCOL$://APP_SERVER:APP_PORT Usage of this parameter is being deprecated from version 20.2 and later. **Brava! Server URL** should be configured while creating the Brava! Viewer Widget in client configuration. If the Brava! Server URL is added here it will be preferred over the client configuration version.
D2URL	**Value:** http://*<localhost>*:*<port>*/D2 This parameter tells Brava! the default root URL to your webapp. This is an optional parameter. By default, Brava! integration works with the application relative URL. This should be only used when you want to overwrite the default relative URL (in the case of reverse proxy, for example) or as needed. **Note:** This setting is used by the client machines to talk to the OpenText Documentum CM Server so firewalls/routers might affect this value. **Example:** D2URL=http://TOMCAT20:8080/D2
BravaConfigurationGroups	**Value:** *<Documentum groups>* Comma delimited list of groups to allow access to the secured Brava! configuration endpoint http://<appserver>:<port>/D2/brava?jx=reload. This endpoint can be used to automatically reload all parameters within the brava_parameters.properties file, providing Administrators and Application Designers a quick method of testing changes to their Brava! configuration. Only logged in users that are members of one of the groups defined in this parameter are able to access the functionality contained in the brava?jx=reload endpoint. **Default:** BravaConfigurationGroups=grpup1,group2,group3
AutoNameMarkup	**Value:** *<TRUE|FALSE>* Turns (automatically) naming of all markup files saved to the Docbase on or off. If true, the name of the markup file is the same as the name of the document. **Default:** AutoNameMarkup=FALSE
ActiveX.Html.Excluded.Formats	**Value:** *<comma delimited list of extensions>* Provides the ability to restrict specified formats from opening in Brava! ActiveX and HTML Client viewers within OpenText Documentum CM. **Default:** activex.html.excluded.formats=avi,mpg,quicktime,java,win32shrlib,unknown,wmv,jar,wav,jsp,js,class,bin,exe,bat,tar,iso,zip

docmerge.thumb.publish.request.timeout	**Value:** <*whole integer*> Related to docmerge, this parameter specifies the time, in minutes, that the server waits for a publish request for thumbnail generation to complete. **Example:** `docmerge.thumb.publish.request.timeout=7`
UseACSIfAvailable	**Value:** <*TRUE|FALSE*> Setting this to `false` causes Brava! to not use ACS (even when OpenText Documentum CM notifies that ACS is working) when Brava! is launched. Sometimes OpenText Documentum CM /WDK sends a response that ACS is functioning but will fail to provide a valid ACS URL. This setting provides a means to get past this issue without requiring any code changes. **Default:** `UseACSIfAvailable=FALSE` To use ACS with the HTML Video Viewer, cross-origin requests must be configured.
UseBOCSIfAvailable	**Value:** <*TRUE|FALSE*> If `true`, Brava! checks the user's current network location and if an associated BOC Server is found, this is used as the location from which to retrieve files. Setting this to `false` causes Brava! to not use BOCS when Brava! is launched (even when receiving notification that BOCS is available). If you would like to use BOCS, you must set this to `true` and also provide the value for the *`bocs.network.location.id`* parameter. **Default:** `UseBOCSIfAvailable=FALSE` To use BOCS with the HTML Video Viewer, cross-origin requests must be configured.
bocs.network.location.id	**Value:** <*your BOCS network ID*> This parameter identifies a failover network location. Brava! tries to determine the users network location via API calls to the content server. If no network location can be found and BOCs is enabled, this value is used as the Network Location for the user. **Example:** `bocs.network.location.id=Corporate` In the example value, `Corporate`, is used as this user’s network location when requesting a BOCs url to the document.
WatermarkFolders	**Value:** <*TRUE|FALSE*> Enables/disables the ability to create watermarks based on the folder name where the content resides. Brava! checks the folder name set in *`WatermarkFolderName`* and compares it to the current folder to see if Brava! should display this folder name as a watermark. **Default:** `WatermarkFolders=FALSE`

WatermarkFolderName	**Value:** <*any valid document folder name*> If the document being viewed is in this folder and the *`WatermarkFolders`* parameter is set to `true`, the watermark is set to be the folder name. Multiple values are allowed, but must be separated by a comma (i.e. `folder1,folder2,folder3`). **Note:** The watermark is seen both on-screen and when printed. **Example:** `WatermarkFolderName=mywatermarks`
WatermarkLineSpacing	**Value:** <*0–3*> Specifies the print density of the print watermark. A value of `0` prints the most opaque watermark, while `3` prints the most transparent. **Example:** `WatermarkLineSpacing=2`
EnableConvertOnClientOfficeFormats	**Value:** <*TRUE|FALSE*> Enables/disables Convert on Client for Office Document formats. Additional purchase is required to license Office formats. **Default:** `EnableConvertOnClientOfficeFormats=FALSE`
ConvertOnClientOfficeFormats	**Value:** <*comma delimited list of extensions*> This parameter tells Brava! what Office formats can be converted on the client. If *`ConvertOnClient`* is `true` and the document extension is listed here, Brava! tries to convert the file on the client by passing the publishing event. Use of this parameter requires special licensing. **Default:** `ConvertOnClientOfficeFormats=pptx,docx,xlsx,cdr,dbf,doc,drw,fax,flw,fmt,fwk,htm,html,img,mpp,msg,pcd,pct,pic,pict,pps,ppt,prs,prz,rnd,rtf,sam,shw,vsd,vw,wb1,wb2,wdb,wk1,wk3,wp,wp5,wp6,wpd,wpf,wpg,wq1,wri,ws,xbm,xls`
EnableConvertOnClientCADFormats	**Value:** <*TRUE|FALSE*> Enables/disables Convert on Client for CAD formats. **Default:** `EnableConvertOnClientCADFormats=FALSE`
ConvertOnClientCADFormats	**Value:** <*comma delimited list of extensions*> This parameter tells Brava! what CAD formats can be converted on the client. If *`ConvertOnClient`* is `true` and the document extension is listed here, Brava! tries to convert the file on the client by passing the publishing event. **Default:** `ConvertOnClientCADFormats=000,906,907,bmp,cal,cg4,cgm,cit,cmi,mi,dcx,dft,dg,dgn,dgn7,dif,dwf,dwg,dx,dxf,edc,edm,ftk,g3,g4,gp4,igs,iso,jpg,jpeg,mil,mot,mvs,pcx,plt,prt,ps,rlc,rle,rnl,slddrw,tif,tiff,dsn,rtl,txt,png,hgl,idw,axp,gif,cc,psd`

EnableConvertOnClientImageFormats	**Value:** <*TRUE|FALSE*> Enables/disables Convert on Client for image formats. **Default:** `EnableConvertOnClientImageFormats=TRUE`
ConvertOnClientImageFormats	**Value:** <*comma delimited list of extensions*> This parameter tells Brava! what image formats can be converted on the client. If *`ConvertOnClient`* is `true` and the document extension is listed here, Brava! tries to convert the file on the client by passing the publishing event. **Default:** `ConvertOnClientImageFormats=tif,pdf,bmp,png,gif,jpg,csf`
ClientLoadedMaxFileSize	**Value:** <*whole integer*> Determines the maximum file size (in kilobytes) allowed for client side publishing (all formats). For example, if the file size of the file is less than this value and its format is TIFF or PDF, the original file is sent to the client for viewing. If the file size is greater than this value, the file is processed normally through the Brava! Server. **Example:** `ClientLoadedMaxFileSize=204800`
PreloadMarkupName	**Value:** <*file name*> Defines the markup file name to be loaded with the source file. The markup file is loaded as editable. If the file does not exist, it is created for you. **Example:** `PreloadMarkupName=InitialReview`
PreloadReadOnlyMarkupNames	**Value:** <*name(s)*> or `ALL` Specifies the markup file(s) to be loaded in review mode (read only) with the source file. Multiple markup files must be delimited by a comma. To load all markup files associated with the source file, use the value `ALL`. **Example:** `PreloadReadOnlyMarkupNames=InitialReview,FinalReview`
EnableEmailDocumentLink	**Value:** <*TRUE|FALSE*> Enables/disables the **Email Document Link** menu item in the Brava! HTML or ActiveX Viewers. When `true`, the user's email client composes an email containing a hyperlink reference to the current document. The recipient of this email, when clicking on this link, is directed to a secure sign in page to enter their credentials and view the specified document in Brava!. **Default:** `EnableEmailDocumentLink=TRUE`

enable.previous.version.publishing.as.new.version	**Value:** *<TRUE|FALSE>* Enables/disables previous version publishing as a new version using Brava! HTML or ActiveX Viewers. By default the parameter is `FALSE` and user cannot publish previous versions as a new version. When the parameter is set to `TRUE`, creation of a new version for previous versions of a document is allowed. **Default:** `enable.previous.version.publishing.as.new.version=FALSE`

37.4.3 3D Client Integration Parameters

The following options are used with both ActiveX and HTML **3D viewing**:

3d_format_name	**Value:** <*yourCustom3Dformat*> 3D documents can be viewed through the activeX and HTML viewers. Documents can be filtered by format name and routed to appropriate 3D viewer. Multiple formats can be entered as a comma delimited list. **Example:** `3d_format_name=acad,dwg`
3d_type_name	**Value:** <*yourCustom3Dtype*> Documents can be filtered by type name and routed to the appropriate 3D viewer. Multiple formats can be entered as a comma delimited list. **Example:** `3d_type_name=your3Dtype0,your3Dtype1`
3d_attribute_name	**Value:** <*title*> 3D documents can be filtered by attribute name and routed to the appropriate 3D viewer. When this parameter is populated, the parameter *`3d_attribute_value`* must also be populated. **Example:** `3d_attribute_name=title`
3d_attribute_value	**Value:** `3D` or <*your identifier for 3D*> This parameter provides the value for *`3d_attribute_name`*. When populated, the parameter*`3d_attribute_name`* must also be populated to filter 3D documents by attribute name. **Example:** `3d_attribute_value=3D`

37.4.4 ActiveX Client Specific Integration

The following integration parameters are used with the ActiveX Viewer only.

<table>
<tr><td>EnableVersion</td><td>Value: <TRUE|FALSE>

Enables/disables the Save markup with all versions function. For the permissions to work properly, you must have performed the required installation step (see OpenText Documentum CM installation guide).

Default: <code>EnableVersion=TRUE</code></td></tr>
<tr><td>EnableDefault Version</td><td>Value: <TRUE|FALSE>

Sets the Version option in the markup Save As dialog to be enabled by default.

Note: The markup Save As dialog does not display at all if <code>EnablePrivate</code> and <code>EnableVersion</code> are set to <code>false</code>, and <code>AutoNameMarkup</code> is set to <code>true</code>.

Default: <code>EnableDefaultVersion=FALSE</code></td></tr>
<tr><td>EnableFileExte nsions</td><td>Value: <TRUE|FALSE>

Enables/disables the default use of adding the original format file extension to PDF, TIF, and CSF output from Brava! ActiveX viewer.

Default: <code>EnableFileExtensions=TRUE</code></td></tr>
<tr><td>EnableSingleE ditableMarkup</td><td>Value: <TRUE|FALSE>

If <code>true</code>, only one markup can be created for each user for each file. It opens the existing markup created by the user for edit when one exists, or it creates a new markup for editing when none exists. To use this feature, <code>EnableMarkupNew</code> must be set to <code>false</code>.

Default: <code>EnableSingleEditableMarkup=FALSE</code></td></tr>
<tr><td>EnablePrivate</td><td>Value: <TRUE|FALSE>

Enables/disables the Save Markup as Private function

Default: <code>EnablePrivate=TRUE</code></td></tr>
<tr><td>EnableDefaultP rivate</td><td>Value: <TRUE|FALSE>

Sets the Private option in the markup Save As dialog to be enabled by default.

Default: <code>EnableDefaultPrivate=FALSE</code></td></tr>
</table>

<table>
<tr><td>ActiveX.Target.New.Window</td><td>Value: <TRUE|FALSE>

If <code>true</code>, this parameter enables the ActiveX viewer to open in a new window each time a document is selected. The parameter is set to <code>false</code> by default. If this parameter is not present, the value is automatically set to <code>false</code>.

To use this parameter:
1. From within IE, open Internet Options and on the General Tab click the Tabs button.
2. Deselect the Enable Tabbed Browsing option (requires restarting Internet Explorer).
3. Close any open Internet Explorer windows.
4. Open Task Manager and kill all <code>iexplorer.exe</code> processes.

Default: <code>ActiveX.Target.New.Window=FALSE</code></td></tr>
<tr><td>AuditBravaView</td><td>Value: <TRUE|FALSE>

A value of <code>true</code> logs a <code>brava_view</code> event to the audit trail of the document when Brava! is launched. This enables some basic auditing events such as opening a file and saving a markup file. When <code>false</code>, no audit trail is logged. <code>true</code> enables additional audit events.

Default: <code>AuditBravaView=TRUE</code></td></tr>
<tr><td>HideStampDialogOnPlacement</td><td>Value: <TRUE|FALSE>

If <code>true</code>, when the stamp is initially placed, the bi-directional stamp dialog does not display. The stamp must contain at least one token besides a <code>dbupdatestring</code> token to be able to detect whether stamp placement is occurring. To enable this functionality, an integration may want to add a <code>dbstring</code> token in the stamp, and have the integration display a blank string for that <code>dbstring</code> token.

Default: <code>HideStampDialogOnPlacement=FALSE</code></td></tr>
<tr><td>RefreshStampOnCancel</td><td>Value: <TRUE|FALSE>

If <code>true</code>, when a user cancels out of the bi-directional stamp dialog, the stamp is updated with the most current values from the server. On an initial stamp placement (there must be at least one token that is not a <code>dbupdatestring</code> for the integration to detect initial stamp placement), the stamp is always populated with the latest values from the server, regardless of whether the user selects Cancel.

Default: <code>RefreshStampOnCancel=TRUE</code></td></tr>
<tr><td>MarkupSaveDescription</td><td>Value: <TRUE|FALSE>

Changes the “Markup Name” string within the ActiveX Viewer Markup > Save As dialog to the string “Markup Description” and disables the dialog’s Save with all Versions and save markup as Private options.

Default: <code>MarkupSaveDescription=FALSE</code></td></tr>
</table>

<table>
<tr><td>CapitalizeDbupdateLabel</td><td>Value: <TRUE|FALSE>

When <code>true</code>, the labels for the dialog items are equal to the <code>DbUpdateToken</code> string value, except that underscores are replaced with spaces, and the first letter of every word is capitalized.

Default: <code>CapitalizeDbupdateLabel=FALSE</code></td></tr>
<tr><td>DbUpdateStringNotRetrieved</td><td>Value: <hide|not set>

If set to <code>hide</code>, the client does not display that field in the <code>DbUpdateString</code> dialog. If not set, the default behavior is to show the field in the dialog with a blank value.

Default: <code>DbupdatestringNotRetrieved=hide</code></td></tr>
<tr><td>BookmarkTitles<<i>n</i>></td><td>Value: <i><category|title></i>

Enables the end-user or administrative user to create a bookmark to an item currently displayed in the Brava! Client. Users can select a category and title from public bookmarks, including free text public bookmarks. To enable this feature, <i><code>EnableExternalBookmarks</code></i> must be <code>true</code>. To make public bookmarks available, define your desired bookmark titles and categories via the bookmarktitles parameters in the format <code>bookmarktitlesn=category|title</code>, where <code>n</code> is any integer.

Example: <code>bookmarktitles1=Authority|Temporary Authority</code>

<code>bookmarktitles2=Authority|Power of Attorney</code>

<code>bookmarktitles3=Home Loan Control Documents|Title Search</code></td></tr>
<tr><td>Theme</td><td>Value: <predefined or custom theme file name>

Determines the theme used by the Brava! ActiveX Client. Several predefined themes are defined and available for the default installation. If you would like to design custom themes to use, please see the <code>Brava! Extensible Interface Reference.pdf</code> guide, for custom theme design definitions

The XML theme file should be zipped and renamed to <code><theme name>.bin</code> and placed in your <code>custom/brava/component/markup</code> directory.

Example: <code>theme=D2</code>

Add your custom theme name to the theme parameters. For example, if your custom theme xml file was zipped and renamed to <code>my_custom_theme.bin</code>, then you would add the following parameters:

<code>theme=my_custom_theme</code>

<code>theme_version=1.1</code>

Note: If you change the theme name in the <code>theme</code> parameter, you must also increment and update the <code>theme_version</code> parameter so the new theme gets downloaded.</td></tr>
</table>

Theme_Version	**Value:** *<hide|not set>* Used with `theme`, the version number determines if the theme needs to be updated and downloaded again to the client if it has already been installed. **Example:** `theme_version=1.0`

37.4.5 Download File Parameters

The following parameters are used by the Client to download specified files.

download.url.<*name*>	**Value:** *<URL>* A URL pointing to a file to be downloaded to the user's directory. If the file is a zip file, it is unzipped. Name can be any value. **Example:** `download.url.{name}=` `download.url.Pdf2DL=http://<Server>/IGC/Pdf2DL.bin`

download.version.<*name*>	**Value:** *<version number>*
	
	The version of the file to be downloaded. This information is stored in the `version.txt` file located in the user's *<user profile>*\IGC *<version>*directory. The version number is used to determine whether an updated file needs to be downloaded. The value of {name} needs to match the value used for the corresponding *download.url.{name}* parameter.
	
	Examples: `download.version.{name}=`
	
	`download.version.Pdf2DL=2.5.7.2`
	
	If the file specified by the *download.url.Pdf2DL* parameter has not yet been downloaded, or if it has and its version is less than 2.5.7.2, then the Client downloads the file (`http://<Server>/IGC/Pdf2DL.bin`). Since this is a zip file, its contents will be unzipped to the user's *<user profile>*\IGC*<version>* directory, and the `version.txt` file will be updated to reflect version 2.5.7.2.
	
	The following example specifies location and version of Redaction Scripts bin file that can be downloaded from the web server and installed to the local client machines. Allows distribution of script files from server.
	
	`download.url.RedactionScripts=http://<server>/IGC/RedactionScripts.bin`
	
	`download.version.RedactionScripts=7.0.0.3`
	
	The following example downloads the current version of the standard reason codes used for setting redaction entities.
	
	`download.version.ReasonCode=1.0.0.1`

download.strictversionmatch. <*name*>	**Value:** <*TRUE*|*FALSE*> When `true`, if the value of the corresponding *`download.version.{name}`* parameter does not match the version listed in the local `version.txt` file, then the file specified by *`download.url.{name}`* is downloaded. If `false`, the client viewer only downloads the file if the version number in the corresponding *`download.version.{name}`* parameter is greater than the value in the local `version.txt` file. **Default:** `download.strictversionmatch.{name}=FALSE`

download.destinationname. <*name*>	**Value:** <*TRUE*|*FALSE*>
	
	If the file specified by the corresponding *`download.url.{name}`* parameter is not a zip file, then the file is renamed to the value of this parameter on the local machine once it has been downloaded.
	
	Default: `download.destinationname.{name}=`
	
	The following example downloads the Changemarks configuration file used for setting type and state options of the Changemarks dialog.
	
	`download.destinationname.ChangemarkConfig=ChangemarkConfig.xml`
	
	The following example downloads the standard theme file used for the Brava! ActiveX client, setting “Blue-Theme” as the default theme:
	
	`download.url.Theme=http://<server>/IGC/Themes/Blue-Theme.xml`
	
	`download.version.Theme=1.0.0.1`
	
	`download.destinationname.Theme=BravaClientTheme.xml`

37.5 Publishing Options

This section details the configurable options available for publishing supported formats to CSF, PDF, and TIFF with the Brava! viewers.

37.5.1 CSF Options

EnableSaveCSFFile	**Value:** <*TRUE*|*FALSE*> Enables/disables the ability to save a CSF file as a new file to the local system. **Example:** `EnableSaveCSFFile=TRUE`

<table>
<tr><td>EnableSaveCSF Email</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a CSF file as an attachment to an email message (MS Outlook or MS Exchange).

Default: <code>EnableSaveCSFEmail=TRUE</code></td></tr>
<tr><td>EnableSaveCSF Rendition</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a CSF file as a Rendition of the file in the Docbase.

Default: <code>EnableSaveCSFRendition=TRUE</code></td></tr>
<tr><td>EnableSaveCSF NewDocFile</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a CSF file as a new file to the user’s home cabinet.

Default: <code>EnableSaveCSFNewDocFile=TRUE</code></td></tr>
<tr><td>EnableSaveAs MajorVersionC SF</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a CSF file as a new major version of the original file.

Note: The user needs to have at least “WRITE” privileges on the object for the Publish as Major Version item to be available as a publishing option.

Default: <code>EnableSaveAsMajorVersionCSF=TRUE</code></td></tr>
<tr><td>EnableSaveAsS ameVersionCS F</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a CSF file as the same version of the original file overwriting the current version. Use with care.

Default: <code>EnableSaveAsSameVersionCSF=FALSE</code></td></tr>
<tr><td>EnableSaveAs MinorVersionC SF</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a CSF file as the next logical minor version of the original file.

Default: <code>EnableSaveAsMinorVersionCSF=FALSE</code></td></tr>
<tr><td>AutoNameCSF Import</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to have the CSF file automatically name itself when saved. If set to <code>true</code>, user is not allowed to manually enter a name.

Default: <code>AutoNameCSFImport=FALSE</code></td></tr>
</table>

<table>
<tr><td>DefaultCSFPublishOption</td><td>Value: <FILE|EMAIL|RENDITION|NEWDOCFILE|MAJORVERSION|MINORVERSION|SAMEVERSION>

Sets the default Publish/Save method for the CSF Publish Option.

Tip: Setting a default method and turning off the individual options allows you to restrict the Publish Option to a single option. Set <code>AutonameCSFImport</code> to <code>true</code> to allow for only 1 method and without prompting.

• <code>File</code> – New file to local system
• <code>Email</code> – New file as email attachment (Exchange or Outlook only)
• <code>Rendition</code> – To Docbase as a CSF rendition
• <code>NewDocFile</code> – New File to Docbase Home Cabinet
• <code>MajorVersion</code> – As a New Major Version to the original file
• <code>MinorVersion</code> – As the next logical minor version of the original file
• <code>SameVersion</code> – Overwrites current file version

Example: <code>DefaultCSFPublishOption=FILE</code></td></tr>
</table>

37.5.2 PDF Options

<table>
<tr><td>UsePDFRendition</td><td>Value: <TRUE|FALSE>

If <code>true</code>, Brava! (ActiveX and HTML Client) uses the PDF Rendition of a file if available. If <code>false</code>, Brava! uses the original file. This setting is ignored (disabled) if launching Brava! through the renditions screen.

See also: <code>PDF_Rendition_Control_For_Groups</code> and <code>PDF_Rendition_Control_For_Users</code>

Example: <code>UsePDFRendition=FALSE</code></td></tr>
<tr><td>EnableSavePDFFile</td><td>Value: <TRUE|FALSE>

Enables/disable the ability to save a PDF file as a new file to the local system. Applies to both ActiveX and HTML viewers.

Example: <code>EnableSavePDFFile=TRUE</code></td></tr>
<tr><td>EnableSavePDFEmail</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a PDF file as an attachment to an email message (MS Outlook or MS Exchange).

Default: <code>EnableSavePDFEmail=TRUE</code></td></tr>
<tr><td>EnableSavePDFRendition</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a PDF file as a Rendition of the file in the Docbase. Applies to both ActiveX and HTML viewers.

Default: <code>EnableSavePDFRendition=TRUE</code></td></tr>
</table>

<table>
<tr><td>EnableSavePDFNewDocFile</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a PDF file as a new file to the user’s home cabinet. Applies to both ActiveX and HTML viewers.

Default: <code>EnableSavePDFNewDocFile=TRUE</code></td></tr>
<tr><td>EnableSaveAsMajorVersionPDF</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a PDF file as a new major version of the original file. Applies to both ActiveX and HTML viewers.

Note: The user needs to have at least “WRITE” privileges on the object for the Publish as Major Version item to be available as a publishing option.

Default: <code>EnableSaveAsMajorVersionPDF=TRUE</code></td></tr>
<tr><td>EnableSaveAsSameVersionPDF</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a PDF file as the same version of the original file overwriting the current version. Use with care. Applies to both ActiveX and HTML viewers.

Default: <code>EnableSaveAsSameVersionPDF=FALSE</code></td></tr>
<tr><td>EnableSaveAsMinorVersionPDF</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a PDF file as the next logical minor version of the original file. Applies to both ActiveX and HTML viewers.

Default: <code>EnableSaveAsMinorVersionPDF=FALSE</code></td></tr>
<tr><td>AutoNamePDFImport</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to have the PDF file automatically name itself when saved. If <code>true</code>, the user is not allowed to manually enter a name.

Default: <code>AutoNamePDFImport=FALSE</code></td></tr>
</table>

<table>
<tr><td>DefaultPDFPublishOption</td><td>Value: <FILE|EMAIL|RENDITION|NEWDOCFILE|MAJORVERSION|MINORVERSION|SAMEVERSION>

Sets the default Publish/Save method for the PDF Publish Option.

Tip: Setting a default method and turning off the individual options allows you to restrict the Publish Option to a single option. Set <code>AutonamePDFImport</code> to <code>true</code> allows for only 1 method and without prompting.

• <code>File</code> – New file to local system
• <code>Email</code> – New file as email attachment (Exchange or Outlook only)
• <code>Rendition</code> – To Docbase as a PDF rendition
• <code>NewDocFile</code> – New File to Docbase Home Cabinet
• <code>MajorVersion</code> – As a New Major Version to the original file
• <code>MinorVersion</code> – As the next logical minor version of the original file
• <code>SameVersion</code> – Overwrites current file version

Example: <code>DefaultPDFPublishOption=FILE</code></td></tr>
</table>

37.5.3 TIFF Options

<table>
<tr><td>EnableSaveTIFFile</td><td>Value: <TRUE|FALSE>

Enables/disable the ability to save a TIF file as a new file to the local system. Applies to both ActiveX and HTML viewers.

Example: <code>EnableSaveTIFFile=TRUE</code></td></tr>
<tr><td>EnableSaveTIFEmail</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a TIF file as an attachment to an email message (MS Outlook or MS Exchange).

Default: <code>EnableSaveTIFEmail=TRUE</code></td></tr>
<tr><td>EnableSaveTIFRendition</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a TIF file as a Rendition of the file in the Docbase. Applies to both ActiveX and HTML viewers.

Default: <code>EnableSaveTIFRendition=TRUE</code></td></tr>
<tr><td>EnableSaveTIFNewDocFile</td><td>Value: <TRUE|FALSE>

Enables/disables the ability to save a TIF file as a new file to the user’s home cabinet. Applies to both ActiveX and HTML viewers.

Default: <code>EnableSaveTIFNewDocFile=TRUE</code></td></tr>
</table>

<table>
<tr>
<td>EnableSaveAsMajorVersionTIF</td>
<td>Value: <TRUE|FALSE>

Enables/disables the ability to save a TIF file as a new major version of the original file. Applies to both ActiveX and HTML viewers.

Note: The user needs to have at least “WRITE” privileges on the object for the Publish as Major Version item to be available as a publishing option.

Default: <code>EnableSaveAsMajorVersionTIF=TRUE</code></td>
</tr>
<tr>
<td>EnableSaveAsSameVersionTIF</td>
<td>Value: <TRUE|FALSE>

Enables/disables the ability to save a TIF file as the same version of the original file overwriting the current version. Use with care. Applies to both ActiveX and HTML viewers.

Default: <code>EnableSaveAsSameVersionTIF=FALSE</code></td>
</tr>
<tr>
<td>EnableSaveAsMinorVersionTIF</td>
<td>Value: <TRUE|FALSE>

Enables/disables the ability to save a TIF file as the next logical minor version of the original file. Applies to both ActiveX and HTML viewers.

Default: <code>EnableSaveAsMinorVersionTIF=FALSE</code></td>
</tr>
<tr>
<td>AutoNameTIFImport</td>
<td>Value: <TRUE|FALSE>

Enables/disables the ability to have the TIF file automatically name itself when saved. If <code>true</code>, the user is not allowed to manually enter a name.

Default: <code>AutoNameTIFImport=FALSE</code></td>
</tr>
<tr>
<td>DefaultTIFPublishOption</td>
<td>Value: <FILE|EMAIL|RENDITION|NEWDOCFILE|MAJORVERSION|MINORVERSION|SAMEVERSION>

Sets the default Publish/Save method for the PDF Publish Option.

Tip: Setting a default method and turning off the individual options allows you to restrict the Publish Option to a single option. Set <code>AutonameTIFImport</code> to <code>true</code> allows for only 1 method and without prompting.

• <code>File</code> – New file to local system
• <code>Email</code> – New file as email attachment (Exchange or Outlook only)
• <code>Rendition</code> – To Docbase as a TIF rendition
• <code>NewDocFile</code> – New File to Docbase Home Cabinet
• <code>MajorVersion</code> – As a New Major Version to the original file
• <code>MinorVersion</code> – As the next logical minor version of the original file
• <code>SameVersion</code> – Overwrites current file version

Example: <code>DefaultTIFPublishOption=FILE</code></td>
</tr>
</table>

37.6 Permissions Parameters

Custom parameters are available for integrators to control Markup, Print, PDF Rendition, and Stamp permissions.

37.6.1 Markup Permissions

The following parameters control markup permissions and default markup permissions for users with READ access.

EnableMarkupReview	**Value:** <*TRUE*|*FALSE*> Enables/disables the ability to Review a markup. **Example:** `enablemarkupreview=FALSE`
RELATE_EnableMarkup<*option*>	**Value:** <*TRUE*|*FALSE*> The markup permissions in the section above are overwritten depending on the permission settings of the RELATE permission. The value of these parameters can be modified to fit your needs. **RELATE_EnableMarkupNew** Turns on/off the ability to create a **New** markup. **RELATE_EnableMarkupSaveAs** Turns on/off the markup **Save As** feature. **RELATE_EnableMarkupReview** Turns on/off the ability to open a markup for review. **RELATE_EnableMarkupSave** Turns on/off the ability to save a markup without giving a markup open for editing a new name. **RELATE_EnableMarkupOpen** Turns on/off the ability to open a markup for edit. **Example:** `RELATE_EnableMarkupNew=true`

HRELATE_EnableMarkup<*option*>	**Value:** <*TRUE*|*FALSE*>
	
	The HRELATE permissions override the RELATE permissions, if set.
	
	Defaults: `HRELATE_EnableMarkupNew=true`
	
	`HRELATE_EnableMarkupSaveAs=true`
	
	`HRELATE_EnableMarkupReview=true`
	
	`HRELATE_EnableMarkupSave=true`
	
	`HRELATE_EnableMarkupOpen=false`

37.6.2 Print, PDF Rendition, and Stamp Template Permissions

<table>
<tr><td>Disable_printing_
for_these_formats</td><td>Value: <coma separated list of formats>

Adding a format to this comma delimitated list of supported formats disables printing through Brava! if the current document's format is in this list.

Note: If printing is disabled for PDF files, then printing of PDF renditions is also disabled. For example, if <code>UsePdfRendition=true</code>, then all files that have PDF renditions have the PDF rendition loaded for viewing in Brava! and not the original file format. Because these are PDF files, which may not be obvious to the end-user, the disable printing for PDF format will be active.

Example: <code>disable_printing_for_these_formats=</code></td></tr>
<tr><td>Printing_Control_For_Groups</td><td>Value: <TRUE|FALSE>

Enables/disables print control for groups.

Default: <code>Printing_Control_For_Groups=FALSE</code></td></tr>
<tr><td>Printing_Control_For_Users</td><td>Value: <TRUE|FALSE>

Enables/disables print control for users.

Default: <code>Printing_Control_For_Users=FALSE</code></td></tr>
<tr><td>Disable_These_Groups_
For_Printing</td><td>Value: <coma separated list of groups>

Controls printing functionality of all valid groups that are listed. If you use the (") character for a value in this parameter, it must be encoded. For example, if <code>Group Name: Standard "Group" One</code> needs to have printing disabled. The correct value would be <code>Standard \"Group\"One</code>.

Example: <code>Disable_These_Groups_For_Printing="Standard \"Group\" One " ;</code></td></tr>
<tr><td>Disable_These_Users_
For_Printing</td><td>Value: <coma separated list of users>

Controls printing functionality of all valid users that are listed. If you use the (") character for a value in this parameter, it must be encoded. For example, if <code>User Name: Standard "User" One</code> needs to have printing disabled. The correct value would be <code>Standard \"User\"One</code>.

Example: <code>Disable_These_Users_For_Printing="Standard \"User\" One " ;</code></td></tr>
<tr><td>PDF_Rendition_Control_
For_Groups</td><td>Value: <TRUE|FALSE>

Enables/disables PDF rendition viewing access for groups when <code>UsePDFRendition=true</code>.

Default: <code>PDF_Rendition_Control_For_Groups=TRUE</code></td></tr>
</table>

PDF_Rendition_Control_For_Users	**Value:** <*TRUE|FALSE*> Enables/disables PDF rendition viewing access for users when `UsePDFRendition=true`. **Default:** `PDF_Rendition_Control_For_Users=FALSE`
Disable_These_Groups_For_Viewing_PDF_Rendition	**Value:** <*coma separated list of groups*> Controls group access to viewing the PDF Rendition of a file when `UsePDFRendition=true`. **Example:** `disable_These_Groups_For_Viewing_PDF_Rendition=`
Disable_These_Users_For_Viewing_PDF_Rendition	**Value:** <*coma separated list of users*> Controls user access to viewing the PDF Rendition of a file when `UsePDFRendition=true`. **Example:** `disable_These_Users_For_Viewing_PDF_Rendition=`
Allow_These_Groups_To_Control_Stamp_Templates	**Value:** <*coma separated list of groups*> Controls group access to creating stamp templates. **Example:** `Allow_These_Groups_To_Control_Stamp_Templates=`
Allow_These_Users_To_Control_Stamp_Templates	**Value:** <*coma separated list of users*> Controls user access to creating stamp templates. **Example:** `Allow_These_Users_To_Control_Stamp_Templates=`

Note: Based on the above permission sets with Printing, Stamp Templates, and Disabling PDF Renditions where Permissions can be set based on Groups and Users, setting control to both (by Group and User) result in the default (User) overriding the Group setting.

Example 37-2:

```
Printing_Control_For_Groups=TRUE

Printing_Control_For_Users=FALSE

PDF_Rendition_Control_For_Groups=FALSE

PDF_Rendition_Control_For_Users=TRUE
```

37.6.3 Unicode Parameter Support

Unicode Characters are supported for use with User and Group configurations. If you want to edit a configuration in the `brava_parameter.properties` file to restrict permissions for user/user groups, you can use Unicode characters in the parameter value by representing them in the format "\u" + hexadecimal code of the character.

Example 37-3:

A user or group with the name containing the Hangul characters 미민독동 should be provided within the `brava_parameter.properties` file as hexadecimal value `\ubbf8\ubbfc\ub3c5\ub3d9`

To restrict a user's access to the print feature in this case, the following parameter is set:

```
disable_These_Users_For_Printing=\ubbf8\ubbfc\ub3c5\ub3d9
```

Important
The encoding of the `brava_parameter.properties` file should not be modified from its installed form (UTF-8).

37.7 Licensing Parameters

The following parameters control licensing behavior:

Licensing_Control_For_CAD_Groups	**Value:** <*TRUE|FALSE*> Turns on license checking based on groups. If `true`, when Brava! is launched, it checks to see if the current user is listed in the groups for the *`License_These_Groups_For_CAD`* parameter. If found, the user is given a CAD license. **Default:** `licensing_control_for_cad_groups=FALSE`
Licensing_Control_For_CAD_Users	**Value:** <*TRUE|FALsE*> Turns on license checking based on users. If `true`, when Brava! is launched, it checks to see if the current user is listed in the groups for the *`License_These_Users_For_CAD`* parameter. If found, the user is given a CAD license. **Default:** `licensing_control_for_cad_users=FALSE`

Licensing_Control_For_Documents_Groups	**Value:** <*TRUE|FALSE*> Turns on license checking based on groups. If `true`, when Brava! is launched, it checks to see if the current user is listed in the groups for the *`License_These_Groups_For_Documents`* parameter. If found, the user is given a Document license. **Default:** `licensing_control_for_document_groups=FALSE`
Licensing_Control_For_Documents_Users	**Value:** <*TRUE|FALSE*> Turns on license checking based on users. If `true`, when Brava! is launched, it checks to see if the current user is listed in the *`License_These_Users_For_Documents`* parameter. If found, the user is given a Document license. **Default:** `licensing_control_for_document_users=FALSE`
License_These_Groups_For_CAD	**Value:** <*any valid group*> This defines the groups that Brava! checks to see if the user is a part of. If the user is found in one of these defined groups, the user receives a CAD license. The groups should be comma deliminated and assumes `Licensing_Control_For_CAD_Groups=true`. The value can be any valid group and the default is empty. The following lists the name Brava! is checking for with the various options: • stamp users : user_name • markup ownership : login_user_name • printing control users : user_name • license control users : login_user_name • rendition view users : user_name **Example:** `license_these_groups_for_cad=group1,group2,group3`
License_These_Groups_For_Documents	**Value:** <*any valid group*> This defines the groups that Brava! checks to see if the user is a part of. If the user is found in one of these defined groups, the user receives a Document license. The groups should be comma deliminated and assumes `Licensing_Control_For_Document_Groups=true`. The value can be any valid group in and the default is empty. The following lists the name Brava! is checking for with the various options: • stamp users : user_name • markup ownership : login_user_name • printing control users : user_name • license control users : login_user_name • rendition view users : user_name **Example:** `license_these_groups_for_documents=group4,group5,group6`

License_These_ Users_ For_Documents	**Value:** <*any valid group*> This defines the users that Brava! checks to see if the user is a part of. If the user is listed as one of these defined users, the user receives a Document license. The users should be comma deliminated and assumes `Licensing_Control_For_Document_Users=true`. The value can be any valid user in and the default is empty. **Note:** This parameter uses the Users short login name (e.g., `Log-in name = johnd`), and not the full OpenText Documentum CM name (e.g., `Full D2 name = John Tester Doe`). **Example:** `license_these_users_for_documents=user1,user2, user3`
License_These_ Users_ For_CAD	**Value:** <*any valid group*> This defines the users that Brava! checks to see if the user is a part of. If the user is listed as one of these defined users, the user receives a CAD license. The users should be comma deliminated and assumes `Licensing_Control_For_CAD_Users=true`. The value can be any valid user and the default is empty. **Note:** This parameter uses the Users short login name (e.g., `Log-in name = johnd`), and not the full name (e.g., `Full D2 name = John Tester Doe`). **Example:** `license_these_users_for_cad=user1,user2,user3`

37.7.1 Licensing Control Precedence

Based on the above permission sets, if the user/group falls into more than one possible user/group slot, they receive the last checked license.

For instance, if `Licensing_Control_For_Documents_Groups=true`, and the current user falls into one of the defined groups in the parameter *`License_These_Groups_For_Documents`*, while at the same time `Licensing_Control_For_CAD_Users=true` and the user also falls into that slot, the user receives a CAD license and NOT a Document license.

The order or precedence when license control is enabled for both Groups and Users is as follows:

`License_These_Groups_For_Documents`

`License_These_Groups_For_CAD`

`License_These_Users_For_Documents`

`License_These_Users_For_CAD`

The user receives the license for the last slot where found.

Chapter 38

BravaCSR Viewer, Brava! Enterprise Viewer, and Intelligent Viewing feature reference

The following tables compare feature support for BravaCSR Viewer, Brava! Enterprise Viewer, and Intelligent Viewing. *CL* denotes Classic client, and *SV* denotes Smart View client.

38.1 Viewer supported formats

For more information about currently supported viewer formats, visit the following resources:

- Brava! Enterprise supported formats (https://www.opentext.com/file_source/OpenText/en_US/PDF/opentext-so-brava-enterprise-supported-formats-en.pdf)
- OpenText Intelligent Viewing supported formats (https://www.opentext.com/assets/documents/en-US/pdf/opentext-intelligent-viewing-supported-formats-en.pdf)

  Note: OpenText Viewing supports all of the formats linked to above, except `.CAD`.

- BravaCSR Viewer supports `.PDF` (and any format that can be rendered to `.PDF`), `.PNG`, `.JPG`, `.BMP`, `.GIF` and `.TIFF`.

38.2 Client feature support

	BravaCSR Viewer		Brava! Enterprise		OpenText Intelligent Viewing
Viewer Feature	**CL**	**SV**	**CL**	**SV**	**SV**
View	YES	YES	YES	YES	YES
Text Search	YES	YES	YES	YES	YES
Annotate	YES	YES	YES	YES	YES
Redact	NO	NO	YES	YES	YES
Publish	NO	NO	`.PDF`, `.TIFF`	`.PDF`, `.TIFF`	`.PDF`, `.TIFF`
DocMerge	NO	NO	YES	YES	NO
Text Compare	NO	NO	YES	YES	YES
Graphical Compare	NO	NO	YES	YES	YES

	BravaCSR Viewer		Brava! Enterprise		OpenText Intelligent Viewing
Viewer Feature	**CL**	**SV**	**CL**	**SV**	**SV**
3D	NO	NO	YES	YES	NO
Video	NO	NO	YES	YES	NO
Print	YES	YES	YES	YES	YES
.zip Viewing	NO	NO	YES	NO	NO

38.3 Display feature support

	BravaCSR Viewer		Brava! Enterprise		OpenText Intelligent Viewing
Viewer Feature	**CL**	**SV**	**CL**	**SV**	**SV**
Thumbnails	YES	YES	YES	YES	YES
Bookmarks	NO	NO	YES	YES	Yes
Measure	NO	NO	YES	YES	YES
Takeoff	NO	NO	YES	YES	YES
Background color	NO	NO	YES	YES	YES
Raster background	NO	NO	NO	NO	NO
Display black and white	NO	NO	YES	YES	YES
Display grayscale	YES	YES	YES	YES	YES
Correct color collisions	NO	NO	YES	YES	YES
Correct color collisions	NO	NO	YES	YES	NO
Show layers	NO	NO	YES	YES	NO
Show line weights	NO	NO	YES	YES	YES
Minimum line thickness	NO	NO	YES	YES	YES

38.4 Annotation feature support

	BravaCSR Viewer		Brava! Enterprise		OpenText Intelligent Viewing
Viewer Feature	**CL**	**SV**	**CL**	**SV**	**SV**
Burn-in Markups	NO	NO	YES	YES	YES
Consolidate Markups	NO	NO	YES	YES	NO
Edit Text Markups	YES	YES	YES	YES	YES
Arc	NO	NO	NO	NO	YES
Arrow Pointer	YES	YES	YES	YES	YES
Changemark	YES	YES	YES	YES	YES
Changemark Arrow	YES	YES	YES	YES	NO*
Changemark Cloud	YES	YES	YES	YES	NO*
Changemark Highlight	YES	YES	YES	YES	NO*
Changemark Text Highlight	YES	YES	YES	YES	NO*
Changemark Text Strikethrough	YES	YES	YES	YES	NO*
Cloud	NO	NO	YES	YES	YES
Cloud Polygon	NO	NO	NO	NO	YES
Crossout	YES	YES	YES	YES	YES
Ellipse	YES	YES	YES	YES	YES
Highlight	YES	YES	YES	YES	YES
Insert Image	YES	YES	YES	YES	YES
Line	YES	YES	YES	YES	YES
Polygon	YES	YES	YES	YES	YES
Polyline	YES	YES	YES	YES	YES
Rectangle	YES	YES	YES	YES	YES
Round Rectangle	YES	YES	YES	YES	YES
Scratchout	YES	YES	YES	YES	YES
Signature	NO	NO	NO	NO	NO
Sketch	YES	YES	YES	YES	YES

	BravaCSR Viewer		Brava! Enterprise		OpenText Intelligent Viewing
Viewer Feature	**CL**	**SV**	**CL**	**SV**	**SV**
Stamps	YES	YES	YES	YES (partial collection**)	YES
Text	YES	YES	YES	YES	YES
Text Strikeout (text scratch)	YES	YES	YES	YES	YES
Text Strikethrough	YES	YES	YES	YES	YES
Text Underline	YES	YES	YES	YES	YES
Text Highlight	YES	YES	YES	YES	YES

* In Intelligent Viewing, collaboration through commenting is no longer limited to Changemark annotations. All annotations are capable of associating comments to them. These annotation types are no longer necessary and will not be implemented in Intelligent Viewing.

** Raster stamps are supported. Symbol stamps are not yet in the CAD Symbols library.

38.5 Redaction feature support

	BravaCSR Viewer		Brava! Enterprise		OpenText Intelligent Viewing
Viewer Feature	**CL**	**SV**	**CL**	**SV**	**SV**
Allow Area	NO	NO	YES	YES	YES
Find and Redact	NO	NO	YES	YES	YES
Redact Area	NO	NO	YES	YES	YES
Redact From-To	NO	NO	YES	YES	YES
Redact Text	NO	NO	YES	YES	YES
Redact Pages	NO	NO	YES	YES	NO
Redact Polygon	NO	NO	YES	YES	YES
Redact Privacy	NO	NO	YES	YES	NO
Publish Review Draft	NO	NO	NO	NO	NO
Redact Using Scripts	NO	NO	NO	NO	YES

	BravaCSR Viewer		Brava! Enterprise		OpenText Intelligent Viewing
Viewer Feature	**CL**	**SV**	**CL**	**SV**	**SV**
Verify Redactions	NO	NO	YES	YES	NO

38.6 Permission comparison

Permission	Annotate	Redact	Publish to PDF	Save to docbase	Version	New rendition
Classic Brava Viewer						
None	Document does not display.					
Browse	No preview.					
Read	No (only review)	No	Yes	Yes*	No	No
Relate	Yes	Yes	Yes	Yes (if the user has write permission in the file location)	No	No
Version	Yes	Yes	Yes	Yes (if the user has write permission in the file location)	Yes	No
Write	Yes	Yes	Yes	Yes (if the user has write permission in the file location)	Yes	Yes
Delete	Yes	Yes	Yes	Yes (if the user has write permission in the file location)	Yes	Yes
* If the user has write permission in the file location. **Smart View Brava Viewer**						
None	Document does not display.					
Browse	No preview.					

Permission	Annotate	Redact	Publish to PDF	Save to docbase	Version	New rendition
Read	No (only review)	No	Yes	Yes (if the user has write permission in the file location)	No	No
Relate	Yes	Yes	Yes	Yes (if the user has write permission in the file location)	No	No
Version	Yes	Yes	Yes	Yes (if the user has write permission in the file location)	Yes	No
Write	Yes	Yes	Yes	Yes (if the user has write permission in the file location)	Yes	Yes
Delete	Yes	Yes	Yes	Yes (if the user has write permission in the file location)	Yes	Yes
Permission	**Smart View Intelligent Viewing**					
None	Document does not display.			N/A		
Browse	No preview.			N/A		
Read	No (only review)	No	Yes	N/A		
Relate	Yes	Yes	Yes	N/A		
Version	Yes	Yes	Yes	N/A		
Write	Yes	Yes	Yes	N/A		
Delete	Yes	Yes	Yes	N/A		

Chapter 39

Workflow Manager (WM) and Business Process Management (BPM) comparison reference

39.1 General comparison

Function	Workflow (Content Server)	BPM (xCP User)	Technical Comments
Create new, open existing workflow template	Supported	Supported	
Save, validate, install, resume suspended processes	Supported	Supported	
Check-out/in process template, manage versions for templates in xCP Designer	Not Supported	Supported	
Multiple process initiate activities	Not Supported	Supported	WF only allows one start and one end for a process.
Create structured data types, no need for package definition	Not Supported	Supported	Variables can be defined outside package structure and used in specific process steps.
Create structured data type - import XML schema, not linked to forms	Not Supported	Supported	Can create variables in the process from external XML docs, such as invoice or PO.
Configure correlation sets, mapping incoming messages to process instances	Not Supported	Supported	Map invoice ID to the running process

39.2 Process templates

Function	Workflow (Content Server)	BPM (xCP User)	Technical Comments
Configure process properties - template owner, description, alias set, instructions	Supported	Supported	
Configure process properties - audit data collection	Supported	Supported	
Configure process properties - packages	Not Supported	Supported	Package defined at the process level
Configure process properties - store document name to package	Not Supported	Supported	Carry document name with package.
Configure process properties - event notification HTML email templates	Not Supported	Supported	In WF it means that the appearance of messages cannot be controlled.
Configure process variables	Not Supported	Supported	In WF, data containers not linked to a package and therefore not linked to the process
Configure process variable monitoring	Not Supported	Supported	WF does not allow mapping of these process variables to monitoring which limits business reporting.
Share process template with Process Analyzer	Not Supported	Supported	Process cannot be shared with Analyzer.
Print process template	Not Supported	Supported	Cannot print process diagrams from Workflow.
Snap to grid	Supported	Supported	
Aligning activities	Not Supported	Supported	
Zoom in and out	Supported	Supported	
Notes	Not Supported	Supported	
View multiple process templates	Not Supported	Supported	Can open multiple processes in tabs.

Function	Workflow (Content Server)	BPM (xCP User)	Technical Comments
Create flows and reject flows	Supported	Supported	
Configure flow style, package name display	Supported	Supported	
Configure flow custom label	Not Supported	Supported	Name display in a flow.

39.3 Activities

Function	Workflow (Content Server)	BPM (xCP User)	Technical Comments
Drag activity to template	Supported	Supported	
Create and add activity templates and palettes	Not Supported	Supported	Cannot reuse Activity designs in Workflow.
Create custom activity templates	Not Supported	Supported	Cannot create Activity designs with additional functionality in Workflow
Organize activity templates into palettes/groups	Not Supported	Supported	
Configure manual or automated activity	Not Supported	Not Supported	
Configure activity performer - work queue	Not Supported	Supported	
Configure activity performer - workflow supervisor	Supported	Supported	
Configure activity performer - docbaseowner	Supported	Supported	
Configure activity performer - previous activity's performer	Supported	Supported	
Configure activity performer - specific user	Supported	Supported	

Function	Workflow (Content Server)	BPM (xCP User)	Technical Comments
Configure activity performer - all users in group	Supported	Supported	
Configure activity performer - single user from a group	Supported	Supported	
Configure activity performer - some users from a group	Supported	Supported	
Configure activity performer - multiple sequential performers	Supported	Supported	
Configure activity performer - conditional from package attributes or process data	Not Supported	Supported	
Configure allow performer to delegate work	Supported	Supported	
Configure allow performer to have someone repeat work	Supported	Supported	
Configure require performer sign-off	Supported	Supported	
Configure task description	Supported	Supported	
Configure task priority	Supported	Supported	
Configure task name with run-time variables	Not Supported	Supported	
Configure task instructions with run-time variables	Not Supported	Supported	
Configure activity trigger with events	Supported	Supported	
Configure notify workflow supervisor for trigger and finish with fixed time.	Supported	Supported	

Function	Workflow (Content Server)	BPM (xCP User)	Technical Comments
Configure multiple pre- and post-timer events	Not Supported	Supported	
Configure transition - all connected activities	Supported	Supported	
Configure transition - let performer choose	Supported	Supported	
Configure transition - based on conditions - package attributes	Supported	Supported	
Configure transition - based on conditions - process data	Not Supported	Supported	
Configure activity label, font, image	Supported	Supported	
Configure notification HTML template for state change, delegation, start	Not Supported	Supported	
Configure package visibility, mandatory, auto-launch attributes for entire process.	Not Supported	Supported	Performed step by step in WF; once in BPM.
Data mapper	Not Supported	Supported	Ease of mapping, not available in WF.
Configure package form	Not Supported	Supported	No forms in WF
Data mapper for Process Integrator activity templates	Not Supported	Supported	Ease of mapping, not available in WF.
Flow activity templates	Not Supported	Supported	No reuse of process designs in WF.
Content services activity templates	Not Supported	Supported	No out of the box content related activity templates for WF.
Auto-activity fault handler activities	Not Supported	Supported	No predefined templates for process steps in WF.

Function	Workflow (Content Server)	BPM (xCP User)	Technical Comments
Activity templates for inbound email, FTP, HTTP, and JMS	Not Supported	Supported	No predefined templates for process steps in WF.
Outbound Email HTML message with variable substitution	Not Supported	Supported	No predefined templates for process steps in WF.
Inbound HTTP HTML response with variable substitution	Not Supported	Supported	No predefined templates for process steps in WF.
BAM observation point for package attributes	Not Supported	Supported	No predefined templates for process steps in WF.
New Case from Template activity template	Not Supported	Supported	No predefined templates for case management in WF.
Queue task activity templates	Not Supported	Supported	No predefined templates for process steps in WF.

Chapter 40

Using Intelligent URLs (iURLs)

40.1 Using iURLs with Classic

Intelligent URL (IURL) is a transparent mechanism for completing standard D2 actions using URI action parameters, as if the action was selected from the menu.

- `Locate`: All types of `locate` should work with and without breadcrumbs.
- `Locate by ID`: Locates a document by its object ID. The first path is taken to locate it in the docbase. The following is a sample URL:

  ```
  http://&lt;server_URL>/D2?
  docbase=<docbase>&login=&<userName>&password=<password>&locateId=<r_object_id>
  ```

- `Locate by name`: Locates a document by its object name. The first document located with this name is used. The following is a sample URL:

  ```
  http://<server_URL>/D2?
  docbase=<docbase>&login=<userName>&password=<password>&locateName=<object_name>
  ```

- `Locate by DQL`: Locates a document by qualification. The following is a sample URL:

  ```
  http://<server_URL>/D2?
  docbase=<docbase>&login=<userName>&password=<password>&locateDql=dm_document where
  object_name=’<object_name>’
  ```

  To have the current version of the document, the `i_chronicle_id` is required:

  ```
  http://<server_URL>/D2?
  docbase=<docbase>&login=<userName>&password=<password>&locateDql=dm_sysobject(all)
  where any r_version_label='CURRENT' and i_chronicle_id =(select i_chronicle_id from
  dm_sysobject(all) where r_object_id = '<r_object_id>')
  ```

- `Locate by path`: Locates a folder or a document in a specific folder.

  To locate a folder:

  ```
  http://<server_URL>/D2?
  docbase=<docbase>&login=<userName>&password=<password>&locatePath=/abrun/Documents/
  ```

  To locate a document with an object ID:

  ```
  http://<server_URL>/D2?
  docbase=<docbase>&login=<userName>&password=<password>&locatePath=/abrun/Documents/
  &locateId=0901ffc380038248
  ```

  To locate a document with an object name:

  ```
  http://<server_URL>/D2?
  docbase=<docbase>&login=<userName>&password=<password>&locatePath=/abrun/Documents/
  &locateName=Object_name
  ```

  To locate a specific version:

  ```
  http://<server_URL>/D2?
  docbase=<docbase>&login=<userName>&password=<password>&locateId=0901ffc38003cfa1&ver
  sion=CURRENT
  ```

The following is the workflow launch:

```
http://<server_URL>/D2?
docbase=<docbase>&login=<userName>&password=<password>&&commandEvent=D2_ACTION_WORKFLOW_L
AUNCH&commandParam=showDialogMaxSize==true!!config==FollowUpWF!!
oam_id==080030398008679e#d2!!finishAction==webSocket!!finishURL==wss%3A%2F%2Flocalhost
%3A57388%2FCTF%2F20210903154904688_19764782431519373972!!
finishMsg==successCheckInMessage!!finishFailedMsg==failedCheckInMessage!!
finishCancelledMsg==cancelOperationMessage&locateId=090030388000a37e
```

40.2 Enabling breadcrumbs in doclist widget for iURL

To use iURL functionality with the doclist widget, you must enable breadcrumbs for the widget.

1. In client configuration, click **Widget view** > **Widget**.
2. Select doclist widget.
3. Enable the **Enable BreadCrumb** option.
4. Save the widget.
5. Refresh the cache by clicking **Perform Tools** > **Refresh cache**.

40.3 iURL viewers

Viewer iURLs allow you to configure a direct call to a viewer. The iURL can be configured to include or exclude the client header. The client header appears above the viewer and includes the client name, user profile, and any other options configured for the header.

The following is the syntax or an viewer iURL, including the header:

```
<http/https>://<<smartview-host>/<smartview-app>>/ui/#d2/viewer?node=<r_
object_id>&viewer-name=<viewer-widget-name>
```

Where:

- *<smartview-host>* is the application host.
- *<smartview-app>* is the application name.
- *<r_object_id>* is the `r_object_id` of the document you want to open. The viewer opens the specific version specified by `r_object_id`.
- *<viewer-widget-name>* is the name of the viewer defined in client configuration. For example, if the viewer is named `myBravaCSRWidget` in client configuration, then you would use `myBravaCSRWidget` as this value. Brava Enterprise, Brava CSR, and Intelligent Viewing are the supported viewers.
- `viewer-name` is optional. If not included in the call, the viewer opens with default configuration and name.

Viewer iURL without header

To create a viewer iURL that hides the header, modify the call as follows:

<http/https>://*<<smartview-host>/<smartview-app>>*/ui/**embed**#d2?mode=viewer&node=*<r_object_id>*&viewer-name=*<viewer-widget-name>*

Chapter 41

Utilities Reference

41.1 d2privilegedclient

Purpose

Adds the respective client as a privileged client.

Location

- Classic: `./webapps/D2/utils/d2privilegedclient`
- Smart View: `./webapps/D2–Smartview/utils/d2privilegedclient`
- The client configuration: `./webapps/D2–Config/utils/d2privilegedclient`

Syntax

```
d2privilegedclient.cmd  -d <repository> -u <install owner> -p <password>
```

Note: Syntax applies to Windows only. For Linux, use the .sh file.

Parameters

The following parameters are used in the syntax.

Name	Description
-d	Docbase
-u	Installation owner
-p	Password