

# DITACMS

## What's New in DITA CMS 4.1



# Table of contents

<b>Chapter 1: New DITA CMS 4.1 features</b>	
<b>Using the Build Manifest feature</b>	<b>6</b>
Understanding the Build Manifest feature	6
Create a Build Manifest	7
Set default values for ditaval files, languages, and user parameters	12
Add and configure output types	13
Output a Build Manifest	16
Search for a Build Manifest	18
<b>DRM libraries</b>	<b>19</b>
Understanding DRM libraries	19
Use libraries in your documents	21
<b>Using the new Concurrent Localization method</b>	<b>25</b>
<b>Importing content as read-only</b>	<b>26</b>
Import external maps as read-only	27
Import external topics as read-only	31
Import external images as read-only	35
Import multiple resolution images as read-only content	38
<b>Import Ditaval files</b>	<b>42</b>
<b>Resolve all the topics in a map</b>	<b>45</b>

<b>Using the root-catalog.xml file to configure catalogs</b>	<b>46</b>
Add the root-catalog.xml catalog to the oXygen editor	46
<b>Miscellaneous improvements to the Search view</b>	<b>49</b>
<b>Other enhancements</b>	<b>50</b>
<b>Chapter 2: New DITA CMS 4.1 Administration features</b>	
<b>Create a map template that generates topics</b>	<b>52</b>
<b>Optimize map display</b>	<b>52</b>

# 1

## New DITA CMS 4.1 features

### Topics:

- **Using the Build Manifest feature**
- **DRM libraries**
- **Using the new Concurrent Localization method**
- **Importing content as read-only**
- **Import Ditaval files**
- **Resolve all the topics in a map**
- **Using the root-catalog.xml file to configure catalogs**
- **Miscellaneous improvements to the Search view**
- **Other enhancements**

This section describes the new features introduced in this release of the DITA CMS.

## Using the Build Manifest feature

---

This section describes how to use the Build Manifest feature.

### Understanding the Build Manifest feature

A *Build Manifest* is a DITA CMS object that allows you to configure a list of different outputs for a map.

For each map, you can specify the following information:

- Types of output to generate (PDF, HTML, xHTML, etc.)
- Languages to output; you can specify default languages for all the output types or specify them per output type
- Ditaval file to use; you can specify a default ditaval file for all the output types or specify it per output type
- User parameters that must be applied; these are the user parameters that are defined in the transformation scenario for the output type. You can specify default parameters for all the output types or specify them per output type.

#### Sample use case

Let's take the *DITA CMS User Guide* as an example. This document is generated in three different output types:

- xHTML
- PDF
- Eclipse help

There are two flavors of this document—oXygen and XMetaL—so two different outputs must be generated for each output type.

For the purpose of this example, we'll also assume that the *User Guide* is available in 3 languages:

- English
- French
- Japanese

This means that a total of 18 outputs (3 output types \* 2 flavors \* 3 languages) need to be generated for the *User Guide*. With the Build Manifest, instead of having to call the Generate Output command 18 times, you generate the output of a single object—the Build Manifest object—which generates these 18 outputs.

## Build Manifest workflow

Just like any object, the Build Manifest goes through a DITA CMS workflow. When you create a Build Manifest object, it's in **Authoring:work** (or the equivalent in your workflow). The object must be locked before you can work on it, and you release it when the work is completed.

You can also create a snapshot of a Build Manifest object.

## Using the NightBuild

IXIASOFT provides a script that you can run to process multiple Output Generator jobs in batch without having to use the DITA CMS Eclipse client. You can call the NightBuild script directly on the command-line or create a batch file that will store the parameter values you want to use for the script.

Using the NightBuild with the Build Manifest can be very useful. You use the DITA CMS Eclipse Client to create the Build Manifest for a map, and then you output the Build Manifest from the NightBuild script.

For more information about the NightBuild script, contact IXIASOFT Support.

## To create a Build Manifest for a map

1. You create a Build Manifest object.
2. If applicable, you set default values for ditaval files, languages, and user parameters.
3. You create and configure all the output types into which you want to generate the map.
4. You output the Build Manifest with the Output Generator, which generates all the outputs configured in the Build Manifest.

This chapter provides these procedures.

## Create a Build Manifest

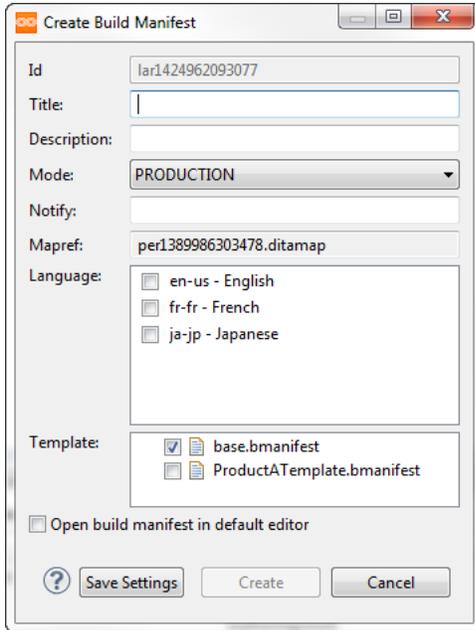
The first step is to create the Build Manifest object.

This procedure will use the *DITA CMS User Guide* as a use case for creating a Build Manifest.

To create a Build Manifest for a map:

- 1. In a view that lists the map, right-click the map and select Create Build Manifest.**

The Create Build Manifest window is displayed:



The following table describes the fields in the Build Manifest window:

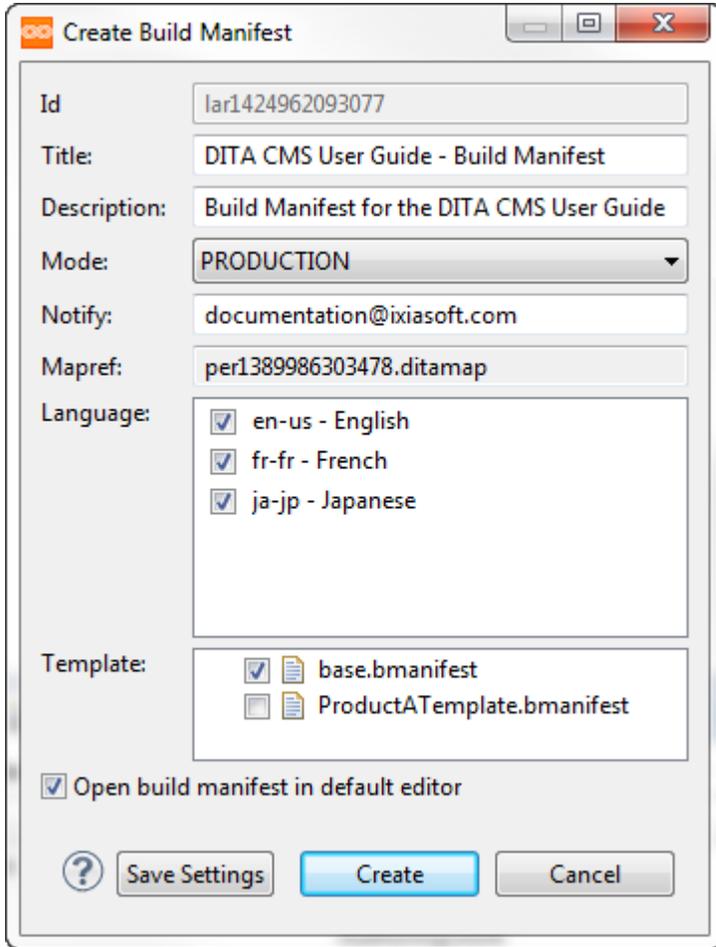
**Table 1: Build Manifest window**

Field	Description
Id	Id of the Build Manifest object. This field is generated automatically and cannot be modified.
Title	Name of the Build Manifest.
Description	Description of the Build Manifest object. The description will be added to the <code>&lt;shortdesc&gt;</code> element of the object.
Mode	Flag that can be used by the NightBuild script in a query to determine whether to generate the Build Manifest. For example, if you have a script that runs every night to generate the output for the maps in your Content Store, you can write a query that uses this field as follows: <ul style="list-style-type: none"> <li>• <b>Mode=Debug</b>: The NightBuild script does not generate output for the Build Manifests that are in Debug mode. This lets you debug your Build Manifest using the DITA CMS interface without impacting the night build.</li> </ul>

Field	Description
	<ul style="list-style-type: none"><li>• <b>Mode=Production:</b> The NightBuild script generates output for the Build Manifests that are in Production mode.</li></ul>
Notify	Email address of the person to notify when the build is completed. This does not have to be a DITA CMS user. <b>Note:</b> For this feature, you must configure the notification message as well as the SMTP information in the Output Generator. See <i>Installing and Configuring the Output Generator</i> for more information.
Mapref	ID of the map to generate.
Language	List of default languages into which to generate the outputs.
Template	Template to use to create this Build Manifest object.

## 2. Enter the fields as appropriate for your Build Manifest.

For example, the following diagram shows the window for the *DITA CMS User Guide*:



**Create Build Manifest**

Id:

Title:

Description:

Mode:

Notify:

Mapref:

Language:

- en-us - English
- fr-fr - French
- ja-jp - Japanese

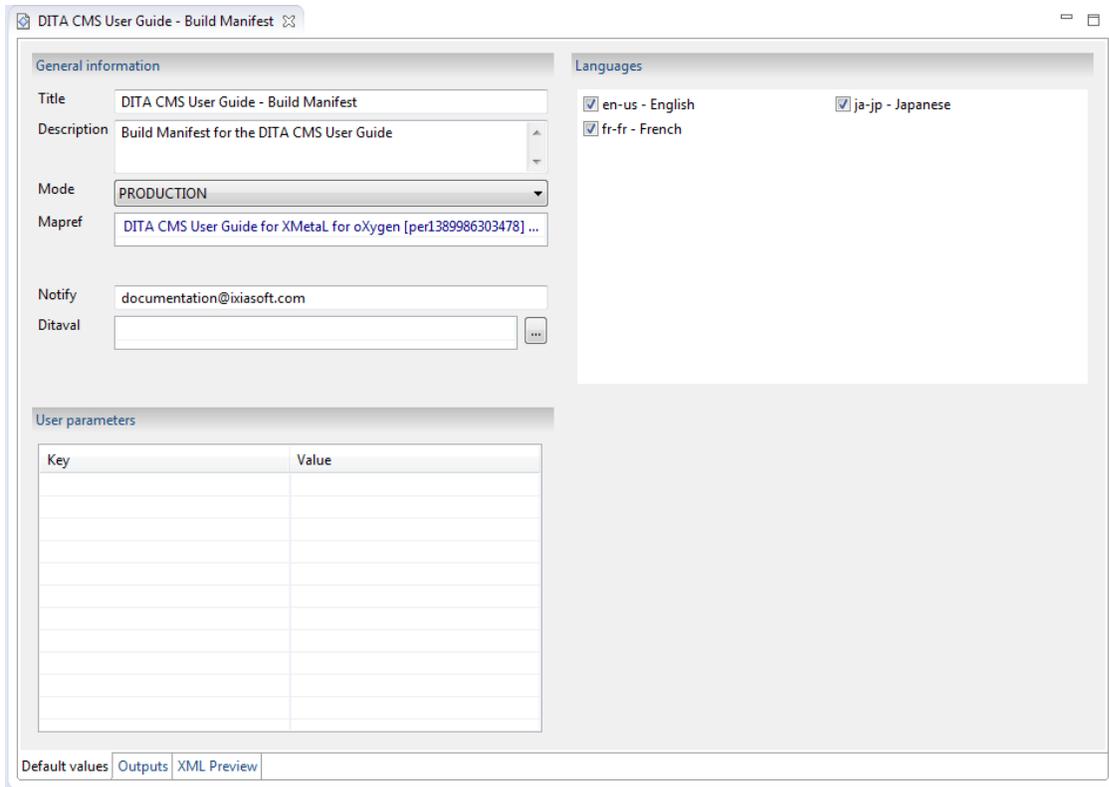
Template:

-  base.bmanifest
-  ProductATemplate.bmanifest

Open build manifest in default editor

3. To open the Build Manifest after you created it and configure the default values and output types, select **Open build manifest in default editor**.
4. **Click Create.**

The Build Manifest object is created. If you selected the **Open build manifest in default editor**, the Build Manifest object is opened in the DITA CMS, as shown below, and the values you entered in the Create Build Manifest window are displayed:



The Build Manifest window has the following tabs:

- **Default values:** Specifies general information about the Build Manifest. It also lets you configure default values for ditaval files, languages, and user parameters. The default values are used for all the output types defined in the Build Manifest, unless you override them for a specific output type.
- **Outputs:** Specifies the different types of output into which to transform the source map. You can override the default ditaval, language, and user parameter values for an output type in this tab.
- **XML Preview:** Provides the XML preview of the Build Manifest object. You cannot modify the XML source.

The next step is to set default values for ditaval files, languages, and user parameters.

## Set default values for ditaval files, languages, and user parameters

You can set a ditaval file, languages, and user parameters that will apply to all the output types defined in the Build Manifest.

When you set a ditaval file, languages, or user parameters in the Default values tab, they will be used for all outputs that do not have ditaval, languages, or user parameters values defined.

When you define values for a specific output type, the default values are ignored for that output type.

To define default values for the output types:

**1. In the Build Manifest window, select the Default values tab.**

**2. To set a default ditaval file:**

a) **In the Ditaval field, click the ... button.**

The Ditaval dialog is displayed.

b) **Select the ditaval file to use and click OK.**

**3. To set default languages, select them in the Languages area.**

**4. To set default user parameters:**

a) **In the User parameters area, right-click the area and select Add.**

The User parameters window is displayed.

b) **In the Key field, enter the name of the user parameter.**

**Note:** The name specified in the Key must match \*exactly\* the name of the user parameter as defined in the *preprocessors.xml* file of the Output Generator; for example, `ixia.args.rellinks`.

c) **In the Value field, enter the value for this user parameter.**

d) **Click OK.**

The parameter is added to the window.

e) **Repeat for every default user parameters that must be set.**

**5. Save the Build Manifest.**

The next step is to add and configure output types.

## Add and configure output types

You add and configure output types in the **Outputs** tab of the Build Manifest window.

For each output type, you define the following information in the **Output type information** area:

- **Type:** Name of the output type from the Output Generator.
  - Note:** You need to be connected to a running Output Generator before you can define output types. The Build Manifest feature retrieves the list of available output types from the running Output Generator.
- **Status:** Specifies whether the output type is enabled or not. This can be useful if you want to configure an output type without generating it right away. When set to **DISABLED**, the output type will not be generated when you output the Build Manifest.
- **Description:** Description of the output type.

You can also override the following values for the output type:

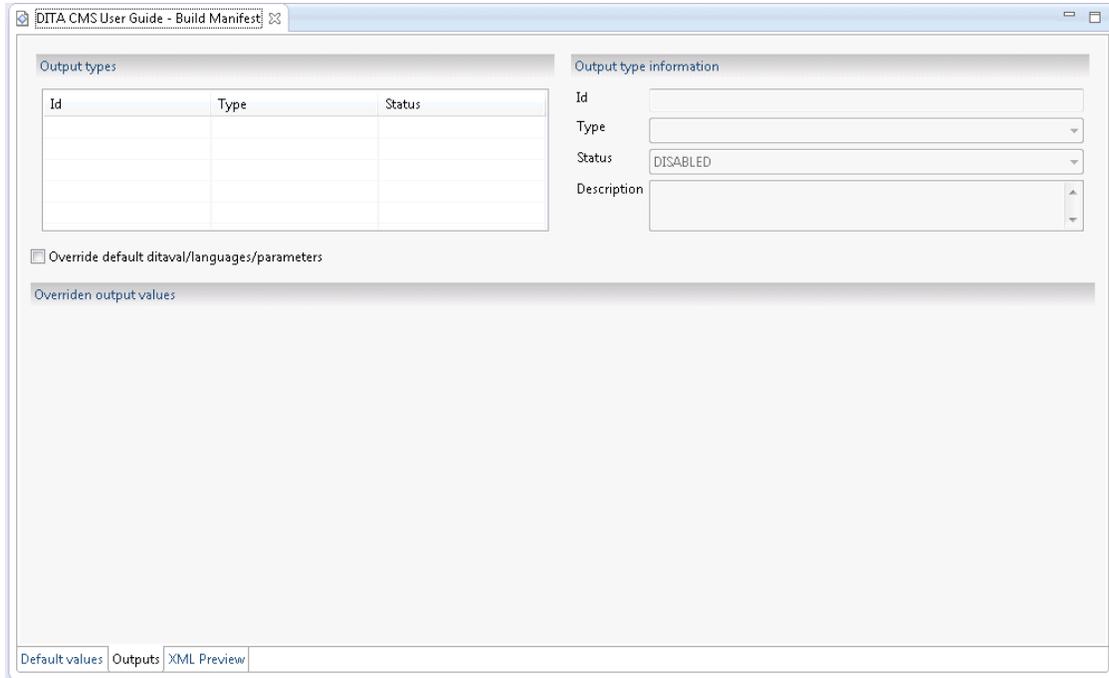
- **Ditaval:** Ditaval file that will be used for this output type. If you specify a ditaval file for the output type, the default ditaval file specified in the **Default values** tab (if any) will be ignored.
- **Languages:** Languages that will be used for this output type. If you specify a language for the output type, the default language(s) specified in the **Default values** tab (if any) will be ignored.
- **User parameters:** User parameters that will be used for this output type. These are the user parameters that are defined in the transformation scenario for the output type. If you specify parameters for the output type, the default parameters specified in the **Default values** tab (if any) will be ignored. The name specified in the Key must match *exactly* the name of the user parameter as defined in the *preprocessors.xml* file of the Output Generator.

**Important:** User parameters are not cumulative. For example, if you set parameter values at the Default level and then set other parameter values at the Output type levels, only the parameters set at the Output type level will be used.

To add output types:

1. **In the Build Manifest window, click the Outputs tab.**

The Output types tab is displayed:



Depending on the Build Manifest template that you are using, one or more output types may already be defined.

**2. To add a new output type, right-click in the Output types area and select Add.**

A new output type is added in the Output types area. By default, the output type is enabled, meaning that it will get generated.

**3. Select the new output type and specify the information for this type in the Output type information area, as follows:**

**Option**                      **Description**

**Type**                              Select the output type

**Note:** If the Type field is displayed in red and you cannot select an output type, this means that there is no Output Generator running. You must connect to a running Output Generator before you can configure an output type.

**Status**                              Select one of the following:

- To generate this output type when you generate the build manifest, select **Enabled**.
- Otherwise, select **Disabled**.

**Description**                      Enter a description for this output type.

4. **To configure a ditaval file, user parameters, or languages specifically for this output type, select the output type and click Override default ditaval/languages/parameters.**

The **Overridden output values** area appears.

5. **To override a ditaval file:**

- a) **In the Ditaval field, click the ... button.**

The Ditaval dialog is displayed.

- b) **Select the ditaval file to use and click OK.**

6. **To override languages, select the languages in the Languages field.**

7. **To override user parameters:**

- a) **In the User parameters area, right-click the area and select Add.**

The User parameters window is displayed.

- b) **In the Key field, enter the name of the user parameter.**

**Note:** The name specified in the Key must match *exactly* the name of the user parameter as defined in the *preprocessors.xml* file of the Output Generator; for example, `ixia.args.rellinks`.

- c) **In the Value field, enter the value for this user parameter.**

- d) **Click OK.**

The parameter is added to the window.

- e) **Repeat for every default user parameters that must be set.**

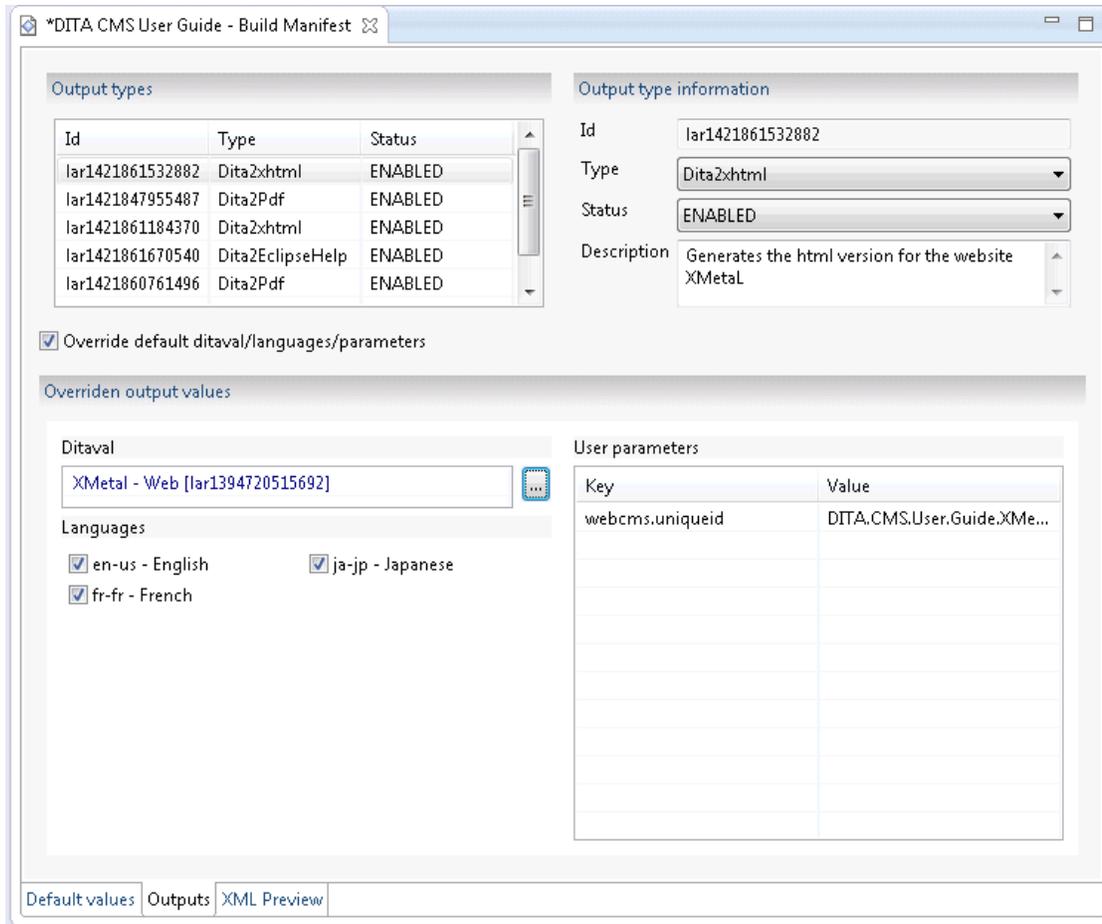
8. **Repeat this procedure for every output type to generate.**

9. **Save the Build Manifest.**

The following diagram shows the **Output type** tab for the *DITA CMS User Guide*. Six output types are created for this document:

- 2 Dita2PDF (one oXygen and one XMetaL)
- 2 Dita2xhtml (one oXygen and one XMetaL)
- 2 Dita2EclipseHelp (one oXygen and one XMetaL)

This is necessary since the oXygen and XMetaL versions require different ditaval files and user parameters.



The final step is to output the Build Manifest.

## Output a Build Manifest

To output a Build Manifest, you use the Build Manifest output generator type.

**Note:** You must release the Build Manifest to get the most recent content.

This procedure describes how to output a Build Manifest using the Generate Output command in the DITA CMS Eclipse Client. You can also use the NightBuild script to generate the output. For more information, contact IXIASOFT Support.

Generating a Build Manifest can take a long time, depending on the size of your maps and the number of output types/languages configured. Therefore, IXIASOFT recommends that you configure an output directory and select the Auto-Save option so that the Build Manifest outputs are saved automatically when the operation is completed. See the section "Configure import and export directories" in the *DITA CMS User Guide* for more information.

To output a build manifest in the DITA CMS Eclipse Client:

## 1. Right-click the Build Manifest to output and select Generate Output.

The Generate Output window is displayed.

## 2. Select the BuildManifest output type and click Create.

The outputs are generated for this map.

**Note:** This operation may take hours, depending on the number of output type/languages and map sizes.

When the output is completed, it is saved in a file called:

```
<build_manifest_title>.BuildManifest.<authoring_language>.zip
```

For example:

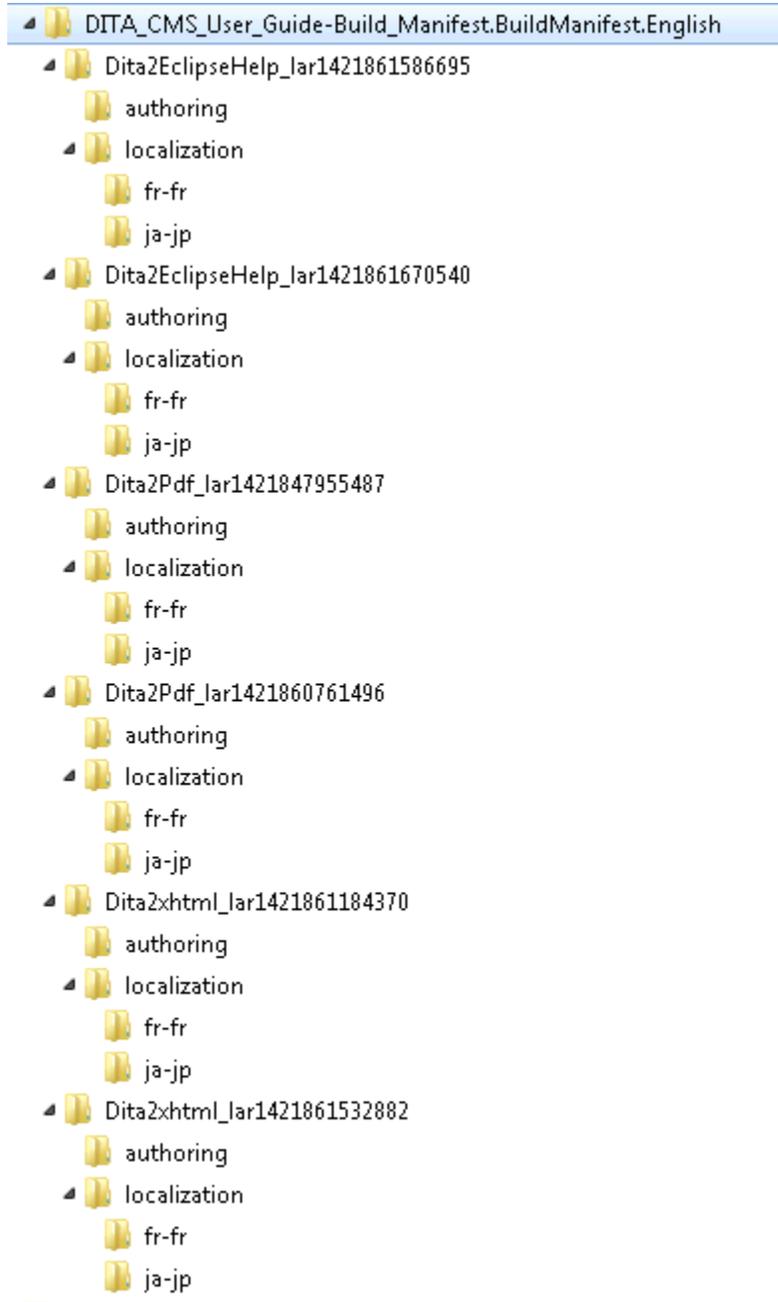
```
DITA_CMS_User_Guide-Build_Manifest.BuildManifest.English.zip
```

The zip file includes the following:

- The Build Manifest file in XML format (for example, *mte1425048934103.bmanifest*)
- A log file for the output operation (for example, *mte1425048934103.build.html*)
- Directories for all the output types, as follows:

```
<output_type_1>
  authoring --> Contains output 1 for the source language
  localization
    <language1> --> Contains the output for the localized language 1
    ...
    <languagex> --> Contains the output for the localized language x
  ...
<output_type_x>
  authoring --> Contains output X for the source language
  localization
    <language1> --> Contains the output for the localized language 1
    ...
    <languagex> --> Contains the output for the localized language x
```

For example, the 18 outputs for the *DITA CMS User Guide* might be packaged as follows:



## Search for a Build Manifest

Build Manifests are categorized as "other" document types in the DITA CMS Search view.

To search for a Build Manifest:

- 1. In the Cycles panel of the Search view, select Authoring.**
- 2. In the Document Types panel, select Others.**

3. Click the drop-down arrow to the right of **Others**.
4. Deselect all types and keep only **build-manifest**.
5. Click the **Search** button.

## DRM libraries

---

This section describes the new DRM library features introduced in DITA CMS 4.1.

### Understanding DRM libraries

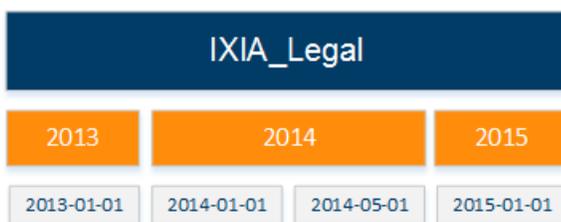
A *library* is a collection of DITA CMS objects that can be shared between different product versions.

Libraries can be created to store legal statements, trademarks, variables, error messages, list of terms, etc. Libraries can be shared between different product versions. They have their own release cycle, which can be different from the release cycles of the products.

**Note:** DRM libraries are very much like the libraries used in computer programs.

Consider a library that stores the legal statements used for all the products in your company. It includes logos, trademarks, copyrights, etc. The legal statements are updated every year to specify a new year in the copyright statement. For this use case you could create a Legal statement library that will be used by all your products. When you update the legal statements, they are updated automatically in all the products that use this library.

Just like products, libraries include releases and versions. The following diagram shows how libraries could be implemented for the IXIASOFT legal statements:



There are three releases for the IXIA\_Legal library: 2013, 2014, and 2015. In the 2013 release, one version was created, called 2013-01-01, for the 2013 copyright statements. In the 2014 releases, two versions were created: one called 2014-01-01 to update the copyright statements to 2014 and one called 2014-05-01 when IXIASOFT updated its branding. A new release was created in 2015, called 2015-01-01, to update the copyright statements to the current year.

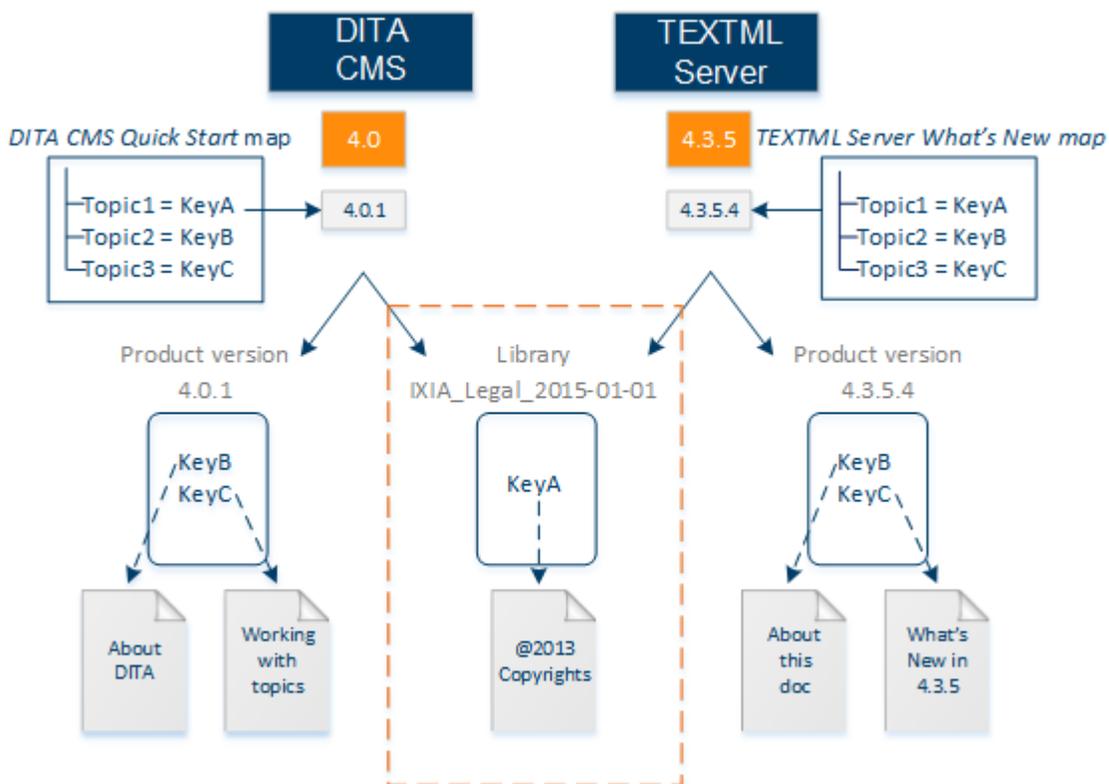
## Working with libraries

You can make libraries available for use with product versions by adding the libraries to the product versions. The DITA CMS then adds the library to the version and the map can reference any object or element in the library.

For example, consider the IXIA\_Legal library. This library is used by the DITA CMS and TEXTML Server products. Every map in the DITA CMS and TEXTML Server products uses the same copyright topic from the library.

The following diagram shows how this is implemented in the DRM. In this example, two maps in two different products use the IXIA\_Legal library:

- The *DITA CMS Quick Start* map for the 4.0.1 version references the IXIA\_Legal\_2015-01-01 library.
- The *TEXTML Server What's New* map for the 4.3.5.4 version also references the IXIA\_Legal\_2015-01-01 library.



As these maps get released on their own release schedules, the version of the library stays the same, so all maps continue to reference the 2013 copyrights until the library is updated.

## Use libraries in your documents

Libraries contain topics that can be shared between different product in the DRM. To use libraries, you first create the initial library structure, the same way that you create a product structure.

1. Create a new library
2. Create a new release for the library
3. Create a new version for the library

Once the library version is created, you add it to the product version. All the documents in that product version can then reference any objects in the library version.

When you need to update the library version—for example, if you need to update a logo or change the copyrights from 2013 to 2014—the process is as follows:

1. Create a new release for the library, if appropriate.
2. In that release, create a new version of the library based on the existing version. This ensures that the keys remain the same between versions.
3. Branch and modify the library objects that you need to update.
4. Update the product versions to use the new version of the library.

These procedures are provided in this section.

## Create a new library

Libraries contain topics that can be shared between different products in the DRM.

Libraries have the file extension *.library* but they are XML files and so they can be most easily edited in your preferred XML editor. You should add a default editor to *.library* files in your Eclipse preferences so that the libraries that you want to edit are opened automatically in your preferred editor.

To create a new library:

1. **In the Dynamic Release Management view, click the Create a new product/library button in the view toolbar.**

The **Create Product or Library** dialog box is displayed.

2. **Enter the library name.**
3. **In the type field, select Library.**
4. **Specify the language for the library in the Language field.**
5. **Enter a description for the library.**

## 6. Optionally:

- a) **Select Open object in default editor to lock the library and open it in your editor for *.library* files.**

Select this option if you need to add more information, such as aliases, to the library.

- b) **Click the Save Settings button to save the Language and Open object in default editor settings for the next time that you create a library in the Create Product or Library dialog box.**

## 7. Click OK.

The library is created and displayed in the **Dynamic Release Management** view. If you selected **Open object in default editor**, then the library is opened in your preferred library editor for you to edit.

## Create a new release

Releases are contained by products or libraries and in turn contain and classify versions.

Releases have the file extension *.release* but they are XML files and so they can be most easily edited in your preferred XML editor. You should add a default editor to *.release* files in your Eclipse preferences so that the releases that you want to edit are opened automatically in your preferred editor.

To create a new release:

1. **In the Dynamic Release Management view, right-click the product or library to which you want to add a release and select New Release.**

The **Create Release** dialog box is displayed.

2. **Give the release name and/or number and a description.**
3. **Specify the language for the release in the Language field.**
4. **Optionally:**

- a) **Select Open object in default editor to lock the release and open it in your editor for *.release* files.**

Select this option if you need to add more information, such as alias release names, to the release.

- b) **Click the Save Settings button to save the Language and Open object in default editor settings for the next time that you create a release in the Create Release dialog box.**

5. **Click OK.**

The release is created and displayed in the **Dynamic Release Management** view under the product or library on which you right-clicked. If you selected **Open object in default editor**, then the release is opened in your preferred release editor for you to edit.

## Create a new version

A version contains the actual documentation objects.

Content is always created for a version. A version must exist before content can be added to it, and the same content object can be used in more than one version.

**Note:** This procedure is useful for creating the first version in a release. If you are creating a subsequent version in a release and want to reuse the same content as the previous version, you should use the Clone a version procedure.

Versions have the file extension *.version* but they are XML files and so they can be most easily edited in your preferred XML editor. You should add a default editor to *.version* files in your Eclipse preferences so that the versions that you want to edit are opened automatically in your preferred editor.

To create a new version:

1. **In the **Dynamic Release Management** view, right-click the release to which you want to add a version and select **New Version**.**

The **Create Version** dialog box is displayed.

2. **Give the version name and a description.**
3. **Specify the language for the version in the Language field.**
4. **Optionally:**
  - a) **Select **Open object in default editor** to lock the version and open it in your editor for *.version* files.**

Select this option if you need to add more information, such as alias version names, to the version.
  - b) **Click the **Save Settings** button to save the Language and **Open object in default editor** settings for the next time that you create a version in the **Create Version** dialog box.**
5. **Click **OK**.**

The version is created and displayed in the **Dynamic Release Management** view under the appropriate release. If you selected **Open object in default editor**, then the version is opened in your preferred version editor.

## Add a library to a product version

To use library objects in a product version, you first add the library to the version.

When a library is added to a version, all documents in that version can reference any object or element in the library.

To add a library to a version:

1. **Right-click the version to which you want to add a library and select *Library > Add Library*.**

The **Select Versions** dialog box is displayed.

2. **In the **All Products/Libraries** pane, select the libraries for which you would like to display the releases and versions.**

Use the filter at the bottom of the pane to filter the libraries in the list by library name.

The libraries, releases, and versions are displayed in the **Selected Products/Libraries, Releases, and Versions** pane.

3. **To exclude libraries, releases, and versions according to the exclusion filters, click **Apply global exclusion filter settings**.**

4. **In the **Selected Products/Libraries, Releases, and Versions** pane, select the library versions to add.**

Use the filter at the bottom of the pane to filter the libraries in the list by release name.

The **Primary Version** drop-down list is populated with the list of selected versions.

5. **Click **OK**.**

**The library is added to the product version. You can now use any library object or element in the product version.**

## Use library elements

Once you have added a library to a version, you can reference any element or object in that library from your version documents.

Adding a library to a version means that any key used in that library is now resolvable by the version.

For example, suppose that you want to use elements from the IXIA\_Legal library in a document. You have added the IXIA\_Legal library version (for example, IXIA\_Legal\_2015\_01\_01) to the version for your map (for example, DITA\_CMS\_4.0.4).

To reference an object from the library, you can:

1. Use the Dynamic Release Management panel in the Search view to display all the objects that are in the IXIA\_Legal\_2015\_01\_01 library version.
2. Add the objects displayed in the Search Results to the map or topic, as appropriate.

## Update a library version

When you need to update the library version—for example, if you need to update a logo or change the copyrights from 2013 to 2014—the process is as follows:

1. Create a new release for the library, if appropriate.
2. In that release, create a new version of the library based on the existing version. This ensures that the keys remain the same between versions.
3. Branch and modify the library objects that you need to update.

The final step is to update the product versions to use the new version of the library, as follows:

1. **In the Dynamic Release Management view, right-click the library version that you want to update and select *Library > Update Library*.**

The **Update Library** dialog box is displayed.

2. **Select the check box for the library that will replace the current library and click OK.**

Use the filter at the bottom of the pane to filter the libraries in the list by library name.

All the documents in the product version now reference the updated elements in the library.

## Using the new Concurrent Localization method

---

The DITA CMS 3.4/4.0 Concurrent Localization method was completely revamped in DITA CMS 4.1.

**Note:** If you are using the DITA CMS 3.4/4.0 version of the concurrent localization method and want to upgrade to the new version, please contact IXIASOFT Support.

It now includes the following features:

- Ability to generate the localization kit in XLIFF and DITA formats
- Very flexible auto-translation process, which can be configured to ignore changes that you consider minor such as extra spaces, punctuation, `<draft-comment>` elements, etc.
- A new status in the workflow that indicates whether a document was completely auto-translated by the system
- A new report that provides detailed feedback on the localization process (coming soon)

- An improved localization kit that provides more content, such as a document that includes all the source text and images to provide context for the translation

For more information about the concurrent localization method, see the *DITA CMS User Guide* and *DITA CMS System Administrator's Guide*.

**Note:** To differentiate between the two versions of the concurrent localization model, the original version was renamed *node-based localization method* in DITA CMS 4.1. The term *concurrent localization method* is used to refer to the improved version introduced in 4.1. The node-based localization method will be deprecated in the next major DITA CMS release. If you are currently using the node-based localization method, IXIASOFT recommends that you upgrade to the new concurrent localization method. Please contact IXIASOFT Support for more information.

## Importing content as read-only

---

You can import external documents into the DITA CMS and mark them as read-only.

This allows you to include content that is not developed by your company and should not be modified, such as:

- Third-party documentation
- DITA content automatically generated from engineering databases
- Etc.

Once imported, you can use read-only content in your documentation. For example, you could add a parts catalog provided by a third-party as an appendix to one of your document.

You can also reference content that was added as read-only. You can use any existing key available in the read-only content.

When you import content as read-only, it is added to the DITA CMS Content Store but it follows a different workflow than your documentation. The documents are put in the **Authoring:readonly** status. You cannot change the status of the documents and you cannot update the content.

The following sections provide the procedures to import external maps, topics, and images as read-only content in the Content Store.

## Import external maps as read-only

The Import Maps wizard lets you add one or more maps as read-only content.

When you import a map, this map and its dependencies (for example, the map, topics, and images it refers to) are imported into the repository and assigned the **Authoring:readonly** status. If one of the map dependencies cannot be found (for example, one of its topic is missing), the import is cancelled and an error message is displayed.

When you import a map, its file name is added to the object's metadata in the `originalFileName` index and the path relative to its containing map is added to the object's metadata in the `OriginalRelativePath` index.

You can specify the following options when importing maps:

Option	Description
<b>Update existing documents</b>	<p>When you re-import an external document, the existing document and its children are always updated. This option is always enabled for external documents.</p> <p>When it imports a document, the DITA CMS determines if this document already exists by checking:</p> <ol style="list-style-type: none"> <li>1. That a document exists in the repository with the same name as the original file name of the imported document.</li> <li>2. If a document with the same name exists, and the <b>Use relative path along with file name to match documents to update</b> option is selected, the DITA CMS also checks that the base path of the imported document also matches the base path of the document in the repository.</li> </ol> <p>If a document already exists, the existing document is updated with the contents of the imported document.</p>
<b>Default language</b>	<p>All documents in the DITA CMS must have a designated language. If the documents you are importing do not have a language attribute, the system will assign them the language you select from this list. If the document you are importing has dependencies, this language will also be assigned to these dependencies.</p>

Option	Description
	<p>You can also set the <b>Override unknown languages with default</b> option. If you select this option, when you import documents that reference a language that's not configured in your system, the system will replace these values with the default language specified.</p>
<p><b>Images</b></p>	<p>If the maps you are importing reference images, you need to specify the type and format of the images.</p> <ul style="list-style-type: none"> <li>• <b>Type</b> is an informative label that is written into the image metadata. Default values are <b>Line Art</b>, <b>Screen Capture</b>, and <b>Equation</b>.</li> </ul> <p>If an image is labeled <b>Line Art</b> or <b>Screen Capture</b>, the system will automatically designate it as Needing Translation. Images labeled <b>Equation</b> are not designated as needing translation; if these images are part of a map that's sent for localization, the system will automatically promote them to the review state and will not include them in the image localization kit.</p> <p>You can set the translation designation for specific images as required once the image has been imported, and the image type may be changed once the image is in the repository.</p> <ul style="list-style-type: none"> <li>• <b>Format</b> may determine the output format of an image, depending on your system setup.</li> </ul>
<p><b>Labels</b></p>	<p>You can assign one or more labels to a document and its dependencies. Labels are added as metadata and improve searchability.</p>
<p><b>Import options</b></p> <p><b>MAP IDs: Retain root map IDs</b></p>	<p>If this option is not selected, the map is assigned a new ID by the system. This will be used as both the file name and as the map root ID. The imported map's root ID is overwritten.</p> <p>If this option is selected, then the value of the imported map ID remains unchanged. If no other map already has this file name, then this ID will also be used as the document file name. If there is a conflict with an existing map file name, the system will assign the imported map a unique file name, which will be different from the map ID.</p>

Option	Description
<b>Topicrefs: Remove @navtitle and @type from &lt;topicref&gt; containing @href</b>	<p>If this option is selected, then <code>navtitle</code> and <code>type</code> attributes are removed from <code>&lt;topicref&gt;</code>s containing an <code>href</code> attribute when the map is imported.</p> <p>When you are authoring DITA documents, there are several ways that you can specify topic attributes such as <code>type</code> and <code>title</code>: at the map level and at topic level. DITA CMS does not maintain map attributes automatically. If you keep these in the map, then you'll need to update them manually if, for example, you change the title of a topic.</p>
<b>Images: Images have unique names</b>	<p>If this option is selected and there are two (or more) images with the same file name, then the system imports just one of them (randomly). If you leave this check box empty, then all images are imported, they keep their duplicate titles, and the system assigns unique file names.</p>
<b>Resources: Resources have unique names</b>	<p>If this option is selected and there are two (or more) resources with the same file name, then the system imports just one of them (randomly). If you leave this check box empty, then all resources are imported, keep their duplicate titles, and the system assigns file names.</p>
<b>Import Base Path:</b>	<p>Specifies the base path to use when replacing existing documents.</p>

To import one or more maps and its dependencies:

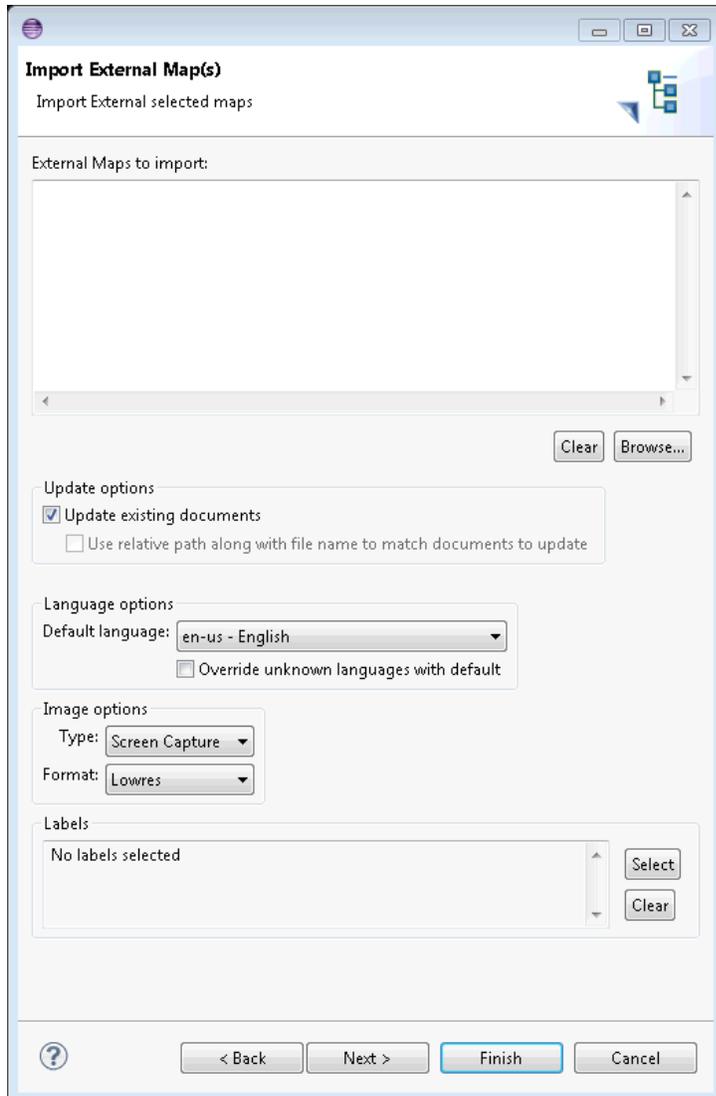
- 1. Select *File > Import...***

The **Import** dialog appears.

- 2. Open the DITA CMS folder and select **Import Maps (EXTERNAL)**.**

- 3. Click **Next**.**

The **Import Map(s)** dialog appears.



**4. Click Browse.**

The **Open** dialog appears.

**5. Navigate to the \*.ditamap file(s), select them, and click Open.**

The selected maps appear in the **Maps to import** list.

**6. To replace existing documents with the contents of the imported documents, select **Update existing documents**. To also use the relative path to match the documents to update, select **Use relative path along with file name to match documents to update**.**

**7. Select the **Default language**.**

**8. If your maps reference images, select a **Type and Format** from the drop-down lists in the **Images** area.**

9. (Optional) To assign a label, click **Select** and add one or more labels. Click **OK** when done.

#### 10. Click Next.

The **Import Options** dialog appears. Select any of the check boxes that apply:

- **MAP IDs: Retain root map IDs**
- **Topicrefs: Remove @navtitle and @type attributes from <topicref> containing @href**
- **Images: Images have unique names**
- **Resources: Resources have unique names**
- **Import Base Path**

#### 11. Click Finish.

## Import external topics as read-only

The Import Topics wizard lets you add one or more topics as read-only content.

When you import a topic, this topic and its dependencies (for example, the topics and images it refers to) are imported into the repository and assigned the **Authoring:readonly** status. If one of the topic dependencies cannot be found (for example, one of its images is missing), the import is cancelled and an error message is displayed.

When you import a topic, its file name is added to the object's metadata in the `originalFileName` index and the path relative to its containing map is added to the object's metadata in the `OriginalRelativePath` index.

You can specify the following options when importing topics:

Option	Description
<b>Update existing documents</b>	<p>When you re-import an external document, the existing document and its children are always updated. This option is always enabled for external documents.</p> <p>When it imports a document, the DITA CMS determines if this document already exists by checking:</p> <ol style="list-style-type: none"> <li>1. That a document exists in the repository with the same name as the original file name of the imported document.</li> <li>2. If a document with the same name exists, and the <b>Use relative path along with file name to match documents to update</b> option is selected, the DITA CMS also checks that the base path of the</li> </ol>

Option	Description
	<p>imported document also matches the base path of the document in the repository.</p> <p>If a document already exists, the existing document is updated with the contents of the imported document.</p>
<b>Default language</b>	<p>All documents in the DITA CMS must have a designated language. If the documents you are importing do not have a language attribute, the system will assign them the language you select from this list. If the document you are importing has dependencies, this language will also be assigned to these dependencies.</p> <p>You can also set the <b>Override unknown languages with default</b> option. If you select this option, when you import documents that reference a language that's not configured in your system, the system will replace these values with the default language specified.</p>
<b>Images</b>	<p>If the topics you are importing reference images, you need to specify the type and format of the images.</p> <ul style="list-style-type: none"><li>• <b>Type</b> is an informative label that is written into the image metadata. Default values are <b>Line Art</b>, <b>Screen Capture</b>, and <b>Equation</b>.  If an image is labeled <b>Line Art</b> or <b>Screen Capture</b>, the system will automatically designate it as Needing Translation. Images labeled <b>Equation</b> are not designated as needing translation; if these images are part of a map that's sent for localization, the system will automatically promote them to the review state and will not include them in the image localization kit.  You can set the translation designation for specific images as required once the image has been imported, and the image type may be changed once the image is in the repository.</li></ul> <ul style="list-style-type: none"><li>• <b>Format</b> may determine the output format of an image, depending on your system setup.</li></ul>
<b>Labels</b>	<p>You can assign one or more labels to a document and its dependencies. Labels are added as metadata and improve searchability.</p>

Option	Description
<b>Import options</b>	
<b>Images: Images have unique names</b>	If this option is selected and there are two (or more) images with the same file name, then the system imports just one of them (randomly). If you leave this check box empty, then all images are imported, they keep their duplicate titles, and the system assigns unique file names.
<b>Resources: Resources have unique names</b>	If this option is selected and there are two (or more) resources with the same file name, then the system imports just one of them (randomly). If you leave this check box empty, then all resources are imported, keep their duplicate titles, and the system assigns file names.
<b>Import Base Path:</b>	Specifies the base path to use when replacing existing documents.

To import one or more topics and its dependencies:

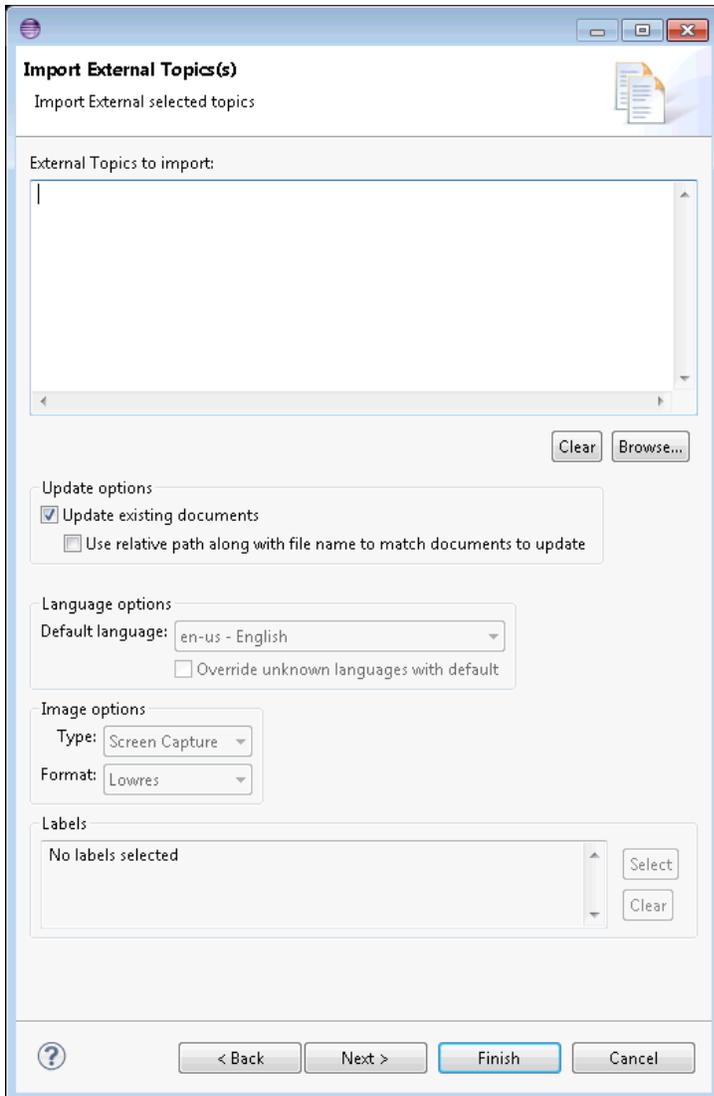
1. **Select *File > Import...***

The **Import** dialog appears.

2. **Open the DITA CMS folder and select **Import Topics (EXTERNAL)**.**

3. **Click **Next**.**

The **Import External Topic(s)** dialog appears.



**4. Click Browse.**

The **Open** dialog appears.

**5. Navigate to the topics, select them, and click Open.**

The selected topics appear in the **Topics to import** list.

**6. To replace existing documents with the contents of the imported documents, select **Update existing documents**. To also use the relative path to match the documents to update, select **Use relative path along with file name to match documents to update**.**

**7. Select the **Default language**.**

**8. If your topics reference images, select a **Type and Format** from the drop-down lists in the Image area.**

9. (Optional) To assign a label, click **Select** and add one or more labels. Click **OK** when done.

10. Click **Next**.

The **Import Options** dialog appears. Select any of the check boxes that apply:

- **Images: Images have unique names**
- **Resources: Resources have unique names**
- **Import Base Path**

11. Click **Finish**.

## Import external images as read-only

The Import Images wizard offers you a way to import images as read-only content.

When you import an image, its file name is added to the object's metadata in the `originalFileName` index and the path relative to its containing map is added to the object's metadata in the `OriginalRelativePath` index.

Images are assigned the **Authoring:readonly** status.

You can specify the following options when importing images:

Option	Description
<b>Update existing documents</b>	<p>When you re-import an external document, the existing document and its children are always updated. This option is always enabled for external documents.</p> <p>When it imports a document, the DITA CMS determines if this document already exists by checking:</p> <ol style="list-style-type: none"> <li>1. That a document exists in the repository with the same name as the original file name of the imported document.</li> <li>2. If a document with the same name exists, and the <b>Use relative path along with file name to match documents to update</b> option is selected, the DITA CMS also checks that the base path of the imported document also matches the base path of the document in the repository.</li> </ol> <p>If a document already exists, the existing document is updated with the contents of the imported document.</p>

Option	Description
<b>Default language</b>	<p>All documents in the DITA CMS must have a designated language. If the documents you are importing do not have a language attribute, the system will assign them the language you select from this list. If the document you are importing has dependencies, this language will also be assigned to these dependencies.</p> <p>You can also set the <b>Override unknown languages with default</b> option. If you select this option, when you import documents that reference a language that's not configured in your system, the system will replace these values with the default language specified.</p>
<b>Images</b>	<p>You need to specify the type and format of the images you are importing.</p> <ul style="list-style-type: none"><li data-bbox="509 846 1468 1209">• <b>Type</b> is an informative label that is written into the image metadata. Default values are <b>Line Art</b>, <b>Screen Capture</b>, and <b>Equation</b>.  If an image is labeled <b>Line Art</b> or <b>Screen Capture</b>, the system will automatically designate it as Needing Translation. Images labeled <b>Equation</b> are not designated as needing translation; if these images are part of a map that's sent for localization, the system will automatically promote them to the review state and will not include them in the image localization kit.  You can set the translation designation for specific images as required once the image has been imported, and the image type may be changed once the image is in the repository.</li><li data-bbox="509 1398 1468 1476">• <b>Format</b> may determine the output format of an image, depending on your system setup.</li></ul>
<b>Labels</b>	<p>You can assign one or more labels to a document and its dependencies. Labels are added as metadata and improve searchability.</p>
<b>Import options</b>	<p><b>Images:</b> <b>Images have unique names</b></p> <p>If this option is selected and there are two (or more) images with the same file name, then the system imports just one of them (randomly).</p>

Option	Description
<b>Import Base Path:</b>	If you leave this check box empty, then all images are imported, they keep their duplicate titles, and the system assigns unique file names.  Specifies the base path to use when replacing existing documents.

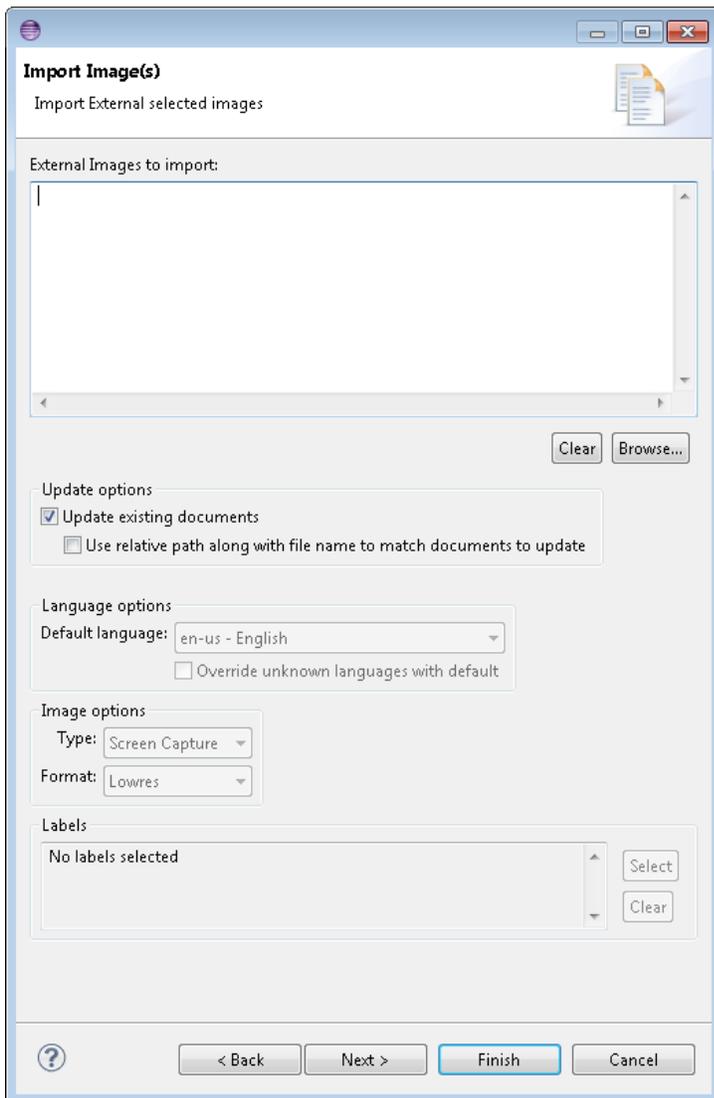
### 1. Select **File > Import...**

The **Import** dialog appears.

### 2. Open the DITA CMS folder and select **Import Images (EXTERNAL)**.

### 3. Click **Next**.

The **Import Image(s)** dialog appears.



#### 4. Click Browse.

The **Open** dialog appears.

#### 5. Navigate to the images, select them, and click Open.

The selected images appear in the **Images to import** list.

#### 6. To replace existing documents with the contents of the imported documents, select **Update existing documents**. To also use the relative path to match the documents to update, select **Use relative path along with file name to match documents to update**.

#### 7. Select the **Default language**.

#### 8. Select a **Type and Format** of the images you are importing.

You can change the values for individual images later on, if you want.

#### 9. (Optional) To assign a label, click **Select and add one or more labels**. Click **OK** when done.

#### 10. Click Next.

The **Import Options** dialog appears. Select any of the check boxes that apply:

- **Images: Images have unique names**
- **Import Base Path**

You can also enter a description for the image.

#### 11. Click Finish.

## Import multiple resolution images as read-only content

The Import Images With Multiple Format wizard lets you import images and their different resolutions as read-only content.

When you publish in different media, you most likely use different image resolutions for each output. You might, for example, want to use a low resolution interlaced JPEG with output that's designed for the web and have a high-resolution version for PDFs.

This procedure lets you simultaneously import all the different formats that you use for each image and automatically assign them to the different output formats. You can, at the same time, specify the default format and whether the images will need to be translated.

Images are assigned the **Authoring:readonly** status.

### Pre-requisites

- Each image resolution must be saved in a different directory. It is recommended to create an import directory with a subdirectory for each type of resolution; for example:

```
C:\images_import\Medres
C:\images_import\PDF
C:\images_import\Web
C:\images_import\Hires
...
```

- When multi-resolution images are imported into the system, they are assigned a collective ID and stored in a zip file with the extension *.image*. To successfully assign the different resolutions to the correct DITA CMS image object, all the different resolutions of an image must share the same file name. For example, you might have a file called *bike.jpg* that you'd use for web output, and another called *bike.png* that you'd use for glossy brochures.

You can specify the following options when importing images:

Option	Description
<b>Default language</b>	<p>All documents in the DITA CMS must have a designated language. If the documents you are importing do not have a language attribute, the system will assign them the language you select from this list. If the document you are importing has dependencies, this language will also be assigned to these dependencies.</p> <p>You can also set the <b>Override unknown languages with default</b> option. If you select this option, when you import documents that reference a language that's not configured in your system, the system will replace these values with the default language specified.</p>
<b>Type and Default format</b>	<p>You need to specify the type and default format of the images you are importing.</p> <ul style="list-style-type: none"> <li>• <b>Type</b> is an informative label that is written into the image metadata. Default values are <b>Line Art</b>, <b>Screen Capture</b>, and <b>Equation</b>.</li> </ul> <p>If an image is labeled <b>Line Art</b> or <b>Screen Capture</b>, the system will automatically designate it as Needing Translation. Images labeled <b>Equation</b> are not designated as needing translation; if these images are part of a map that's sent for localization, the system will automatically promote them to the review state and will not include them in the image localization kit.</p>

Option	Description
	<p>You can set the translation designation for specific images as required once the image has been imported, and the image type may be changed once the image is in the repository.</p> <ul style="list-style-type: none"><li>• <b>Format</b> may determine the output format of an image, depending on your system setup.</li></ul>

### To import multiple resolution images:

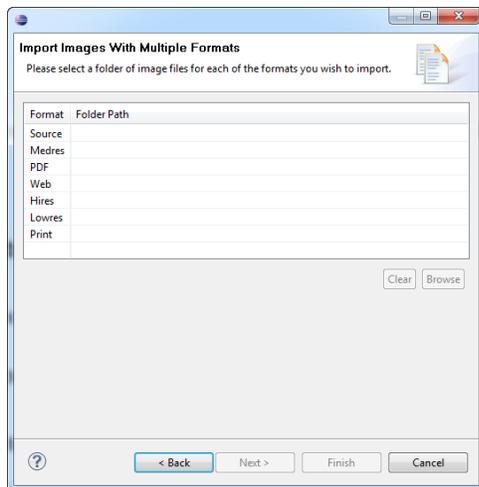
1. **Select *File > Import...***

The **Import** dialog appears.

2. **Open the DITA CMS folder and select *Import Images With Multiple Formats (EXTERNAL)*.**

3. **Click *Next*.**

The **Images With Multiple Formats** dialog appears.

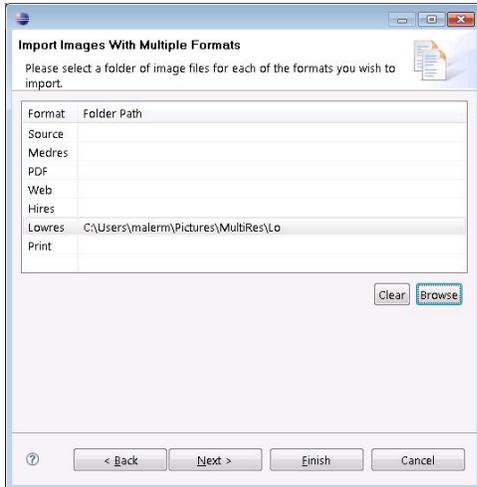


4. **Select the image **Format** to import and click **Browse**.**

The **Browse For Folder** dialog appears.

5. **Navigate to the directory that contains the images for this format and click **OK**.**

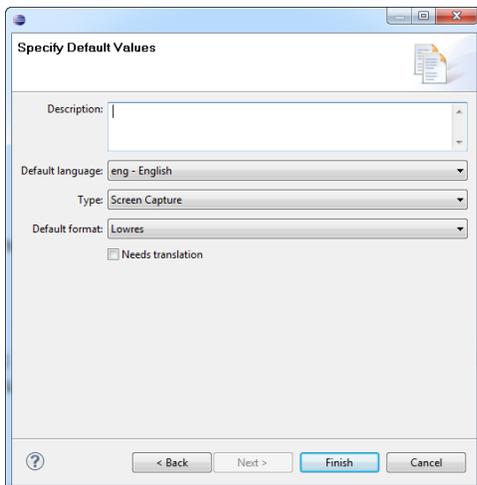
The selected directory appears in the **Folder Path** list.



6. Repeat steps 4-5 as required for all the formats that you are using.

7. (Optional) Click Next.

The **Specify Default Values** dialog appears.



The values you specify will be used for all the resolutions of all the images. You can change the values for individual images later on, if you want.

8. Enter a Description.

9. Select the **Default language**.

10. Select a **Type**.

11. Select the **Default format**.

This may determine the output format that an image will be used for, depending on your system setup.

12. Select or deselect **Needs Translation**, as required.

Certain types of graphics may automatically enable this check box. This lets you override the default.

You can change the values for individual images later on, if you want.

### 13. Click Finish.

## Import Ditaval files

---

You can import Ditaval files in the DITA CMS.

These files must follow the standard Ditaval format defined in the Oasis standard. See the following link for more information:

<http://docs.oasis-open.org/dita/v1.2/os/spec/common/about-ditaval.html>

When you import a Ditaval file, its file name is added to the object's metadata in the `originalFileName` index and the path relative to its containing map is added to the object's metadata in the `OriginalRelativePath` index.

**Note:** Any document you import is automatically assigned to you.

### Specifying a Ditaval title and description

You can specify the Ditaval title and description inside processing instructions before importing the Ditaval file. This ensures that a meaningful title and description are associated with this Ditaval file.

To specify the title and description, add the following processing instructions to your Ditaval file before importing it:

- `ixiasoft.ditaval.title`
- `ixiasoft.ditaval.description`

### For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<?ixiasoft.ditaval.title Internal User Guide for Linux ?>
<?ixiasoft.ditaval.description Conditions for publishing the User Guide for Linux to an
internal audience ?><val>
  <prop action="include" att="product"/>
  <prop action="include" att="audience"/>
  <prop action="include" att="rev"/>
  <prop action="include" att="platform"/>
  <prop action="exclude" att="audience" val="External"/>
  <prop action="include" att="audience" val="Internal"/>
  <prop action="include" att="platform" val="Linux"/>
</val>
```

### Import options

You can specify the following options when importing Ditaval files:

Option	Description
<b>Update existing documents</b>	<p>When it imports a document, the DITA CMS determines if this document already exists by checking:</p> <ol style="list-style-type: none"> <li>1. That a document exists in the repository with the same name as the original file name of the imported document.</li> <li>2. If a document with the same name exists, and the <b>Use relative path along with file name to match documents to update</b> option is selected, the DITA CMS also checks that the base path of the imported document also matches the base path of the document in the repository.</li> </ol> <p>If a document already exists and the <b>Update existing documents</b> option is selected, the existing document is updated with the contents of the imported document. If this option is not selected, a new document is created, with a new file name.</p> <p>Note that, to be updated, the document:</p> <ul style="list-style-type: none"> <li>• Must be in a lockable state (for example Authoring:Draft)</li> <li>• Must not be already locked by another user</li> </ul>

### Import options

<b>Import Base Path:</b>	Specifies the base path to use when replacing existing documents.
--------------------------	-------------------------------------------------------------------

To import one or more Ditaval files:

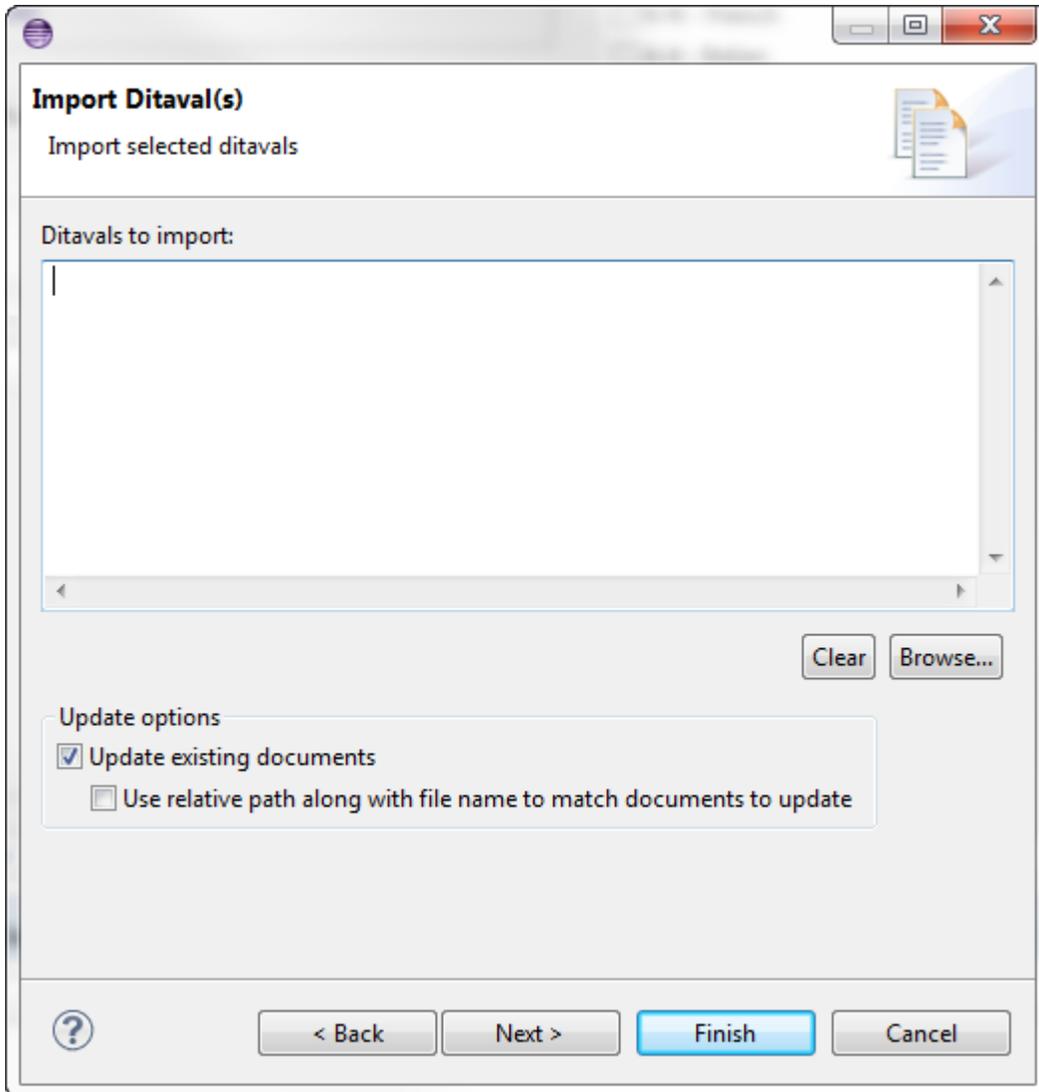
1. **Select *File > Import...***

The **Import** dialog appears.

2. **Open the DITA CMS folder and select Import Ditaval.**

3. **Click Next.**

The **Import Ditaval(s)** dialog appears.



4. **Click Browse.**

The **Open** dialog appears.

5. **Navigate to the folder that contains the Ditaval file(s), select the file(s), and click Open.**

The selected Ditaval file(s) appear in the **Ditavals to import** list.

6. **To replace existing documents with the contents of the imported documents, select **Update existing documents**. To also use the relative path to match the documents to update, select **Use relative path along with file name to match documents to update**.**

7. **Click Next.**

The **Import Options** dialog appears. Enter the **Import Base Path** if applicable.

8. **Click Finish.**

The DitaVal file is imported is imported.

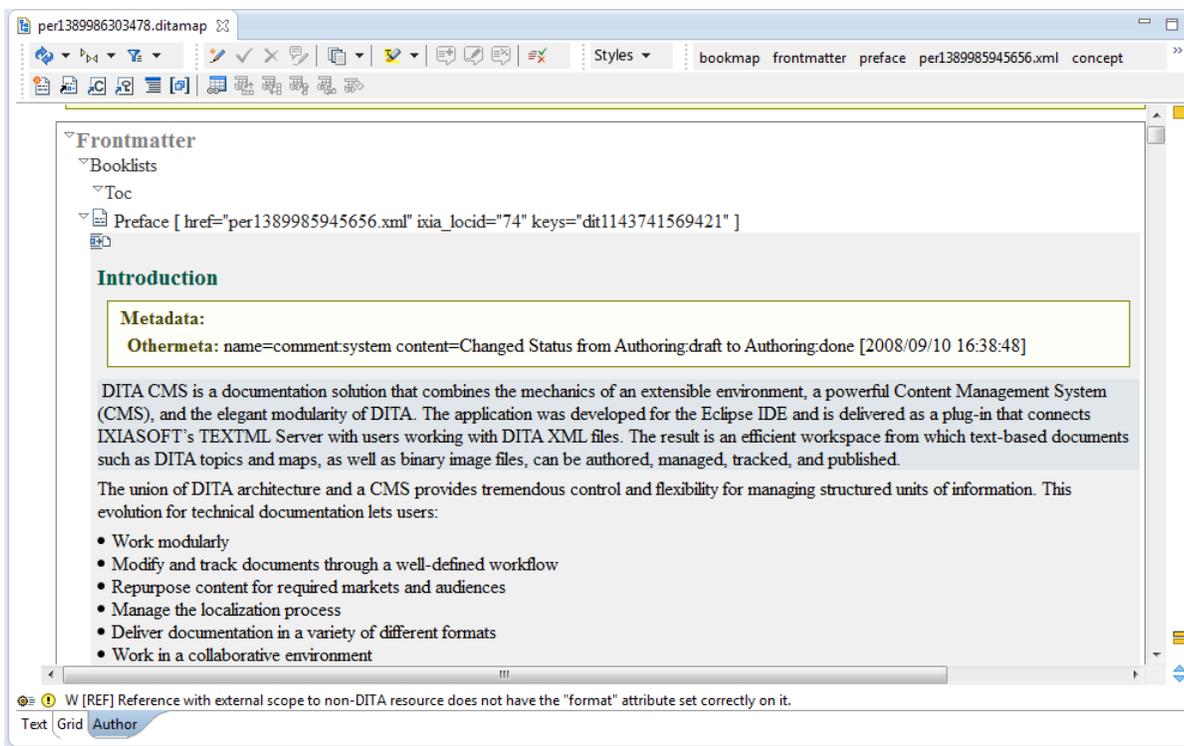
## Resolve all the topics in a map

You can now resolve all the topics when opening a map.

**Note:** This feature applies to oXygen only.

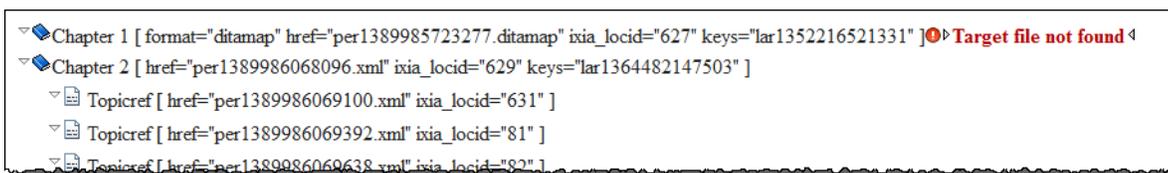
This option can be useful to see the content of the whole map without having to generate the output of the map.

When you select this option, you can see the contents of the map in the Author view of the map. For example:



**Note:** You can see the content of the map, but you cannot edit it. You must open each object separately before you can edit it.

This option also allows you to open the map in oXygen and not see error messages such as the following:



To resolve all the keys in a map:

1. **Open a view that lists the map.**
2. **Right-click the map and select *Open With > XML Editor with resolved topics*.**

The DITA CMS resolved all the keys in the map and opens the map. Depending on the size of the map, this can take a long time.

## Using the *root-catalog.xml* file to configure catalogs

---

Specifying catalogs is now easier in DITA CMS 4.1.

The DITA CMS uses catalogs to locate the DTDs that define the DITA elements and objects used in your system. In DITA CMS 4.0, the *master-catalog.xml* catalog was added to the oXygen editor. This catalog was specific to a user's current server and Content Store, so users had to configure the catalogs again if they changed server, Content Store, username, etc.

DITA CMS 4.1 introduces a new catalog to simplify this process. This catalog, *root-catalog.xml*, is stored outside of the user's repository and dynamically points to the *master-catalog.xml* catalog in the Content Store currently used.

The advantage to this method is that you need to configure the path to the *root-catalog.xml* only once, when you set up the DITA CMS Eclipse Client.

## Add the *root-catalog.xml* catalog to the oXygen editor

For the DITA CMS to locate the DTDs for the DITA objects and elements in your documentation, you must add the *root-catalog.xml* file to the oXygen editor.

The *root-catalog.xml* dynamically points to the *master-catalog.xml* catalog file, which contains the list of all the catalogs used by the DITA CMS. If you add a new catalog to the system, make sure to add it to the *master-catalog.xml* catalog file as described in the *Integrating DTDs into the DITA CMS* document.

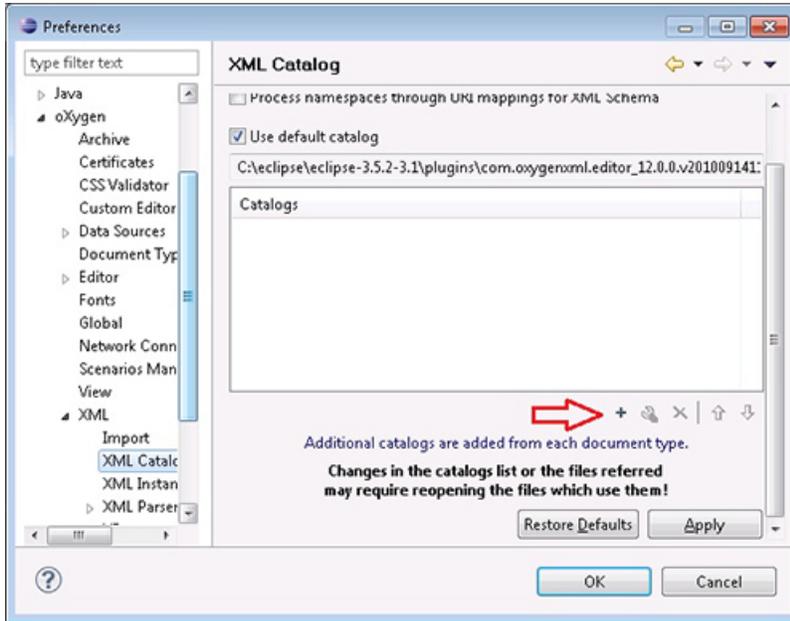
1. **From the Window menu, select Preferences.**

The **Preferences** dialog opens.

2. **Select *oXygen > XML > XML Catalog*.**

The **XML Catalog** pane appears.

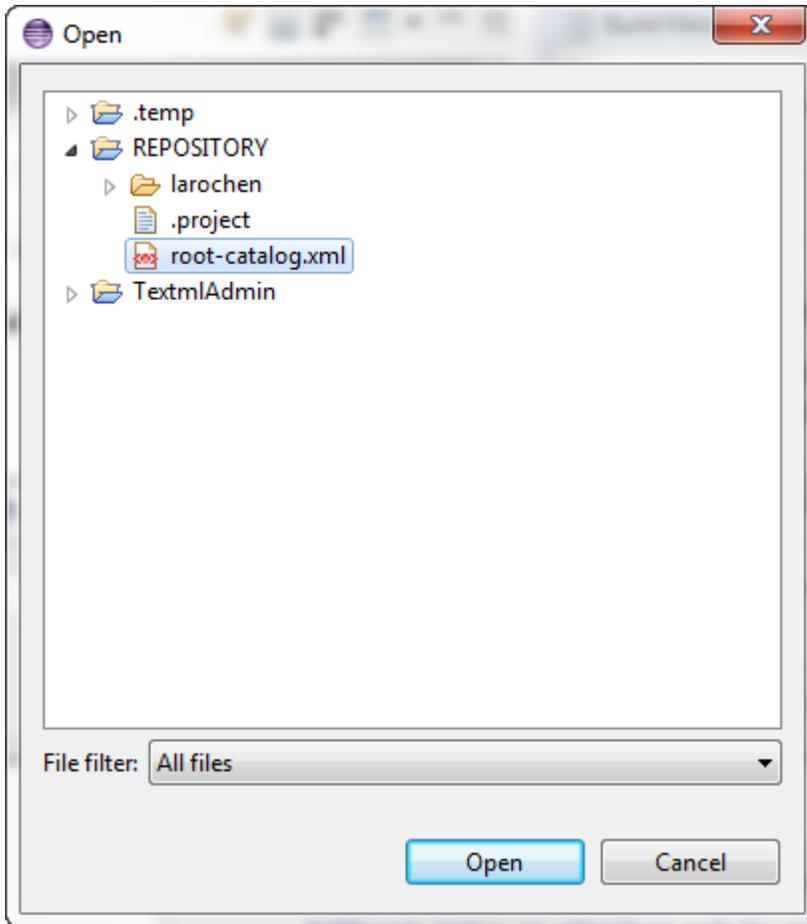
3. **Click the New (+) button, as shown below.**



The **Choose Catalog** dialog appears.

4. In the **Choose Catalog** dialog, click the **Browse** button's selection arrow  and then select **Browse workspace**.

The **Open** dialog appears.



**5. Expand the REPOSITORY node and then select the *root-catalog.xml* file.**

**Note:** If you do not see this file:

- Click **Cancel** twice to return to the **Preferences** dialog.
- In the **Preferences** dialog, select **General > Workspace**.
- Select **Refresh using native hooks or polling** and click **Apply**.
- Repeat this procedure from step 2. You should now see the file.

**6. Click Open.**

The **Choose Catalog** dialog reappears, with your selection in it.

**7. Click OK.**

The **Preferences** dialog is displayed again and the *root-catalog.xml* catalog is listed in the Catalogs list.

**8. In the Preferences dialog, click Apply and then OK.**

The DITA CMS catalogs are now available to the oXygen editor.

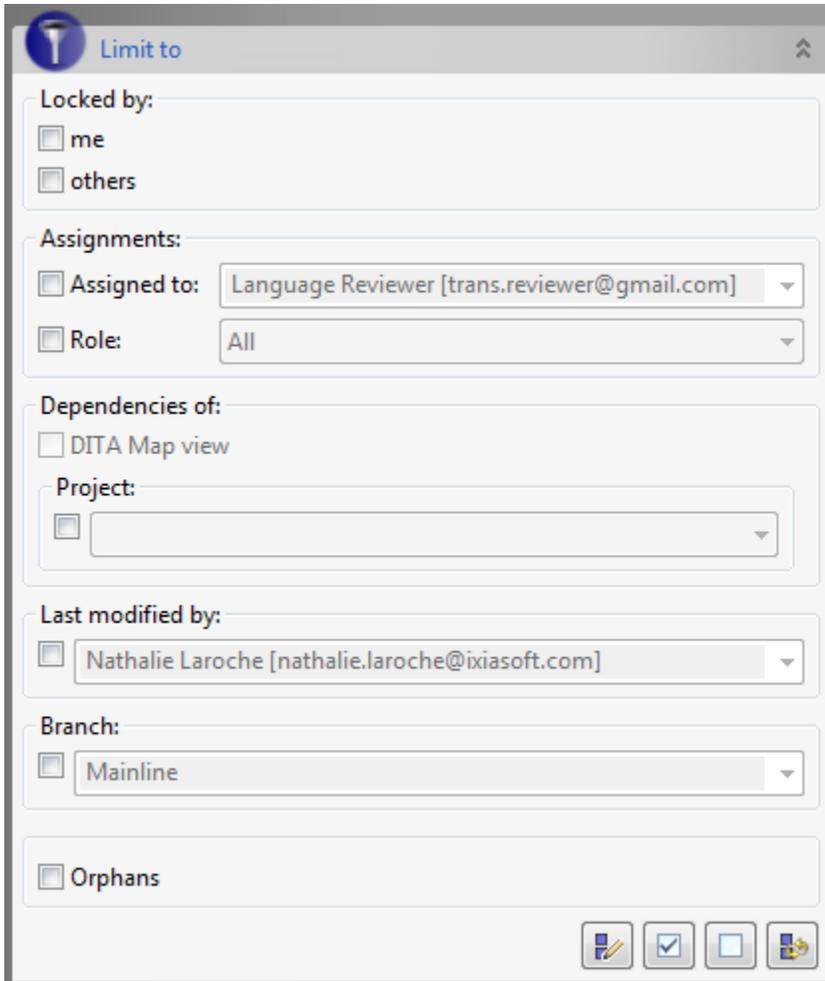
## Miscellaneous improvements to the Search view

---

The DITA CMS 4.1 includes miscellaneous improvements to the Search view.

### Limit to panel

In particular, the Limit to panel was updated to allow users to search objects that are assigned to a specific user or role, as shown below:



To search documents assigned to a specific user:

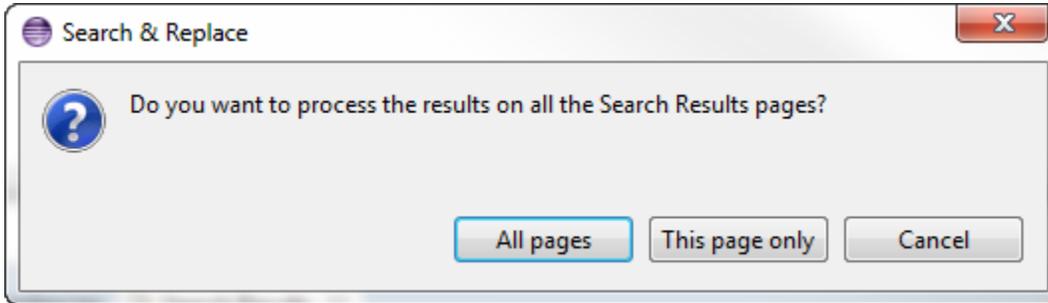
In the **Limit to** panel, select **Assigned to** and then select the name of the user in the drop-down list. You can also enter text directly in the **Assigned to** field to search for a specific user name.

To search documents assigned to a specific role:

In the **Limit to** panel, select **Role** and then select the role in the drop-down list.

## Search and Replace function in the Search Results

If search results are provided over many pages, the Search and Replace operation now asks users whether they want to search the current page or all the pages:



## Other enhancements

---

This section lists other enhancements made to the DITA CMS in 4.1:

- Setting access rights for the DITA CMS actions is now easier and more efficient. See the *DITA CMS System Administrator's Guide* for more information.
- Some paths in the IXIASOFT DTDs have been updated for optimization. The updated DTD files are imported in your configuration when you upgrade to the DITA CMS 4.1 as described in the *DITA CMS 4.1 Upgrade Guide*.

# 2

## New DITA CMS 4.1 Administration features

### Topics:

- **Create a map template that generates topics**
- **Optimize map display**

This section describes the new features for CMS Administrators.

## Create a map template that generates topics

---

You can create a map template with topics that will be generated automatically when users create a map based on this template.

By default, if you add topics to a map template, they will be added as stubs to the map: empty positions are created in the map and act as place holders, but the actual topics are not generated.

To indicate that a topic should be generated when a map is created from a template, you add the `xtrc="generate"` attribute to the topic. For example, consider the following map template:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE map PUBLIC "-//IXIA//DTD IXIA DITA Map//EN" "IxiaMap.dtd">
<map id="mte1408546420716" xml:lang="en-us">
  <title>Template Map</title>
  <topicref navtitle="About this document"
    type="Oxygen/concept.xml"
    xtrc="generate" />
  <topicref navtitle="Requirements"
    type="Oxygen/concept.xml"/>
</map>
```

When a map is generated from this template, the "About this document" topic will be generated automatically, while only a topic stub will be created for the "Requirements" topic.

## Optimize map display

---

By default, when opening a map in the DITA Map view, the DITA CMS loads all the children of the map.

This operation can take a long time for large maps. To improve the speed at which the DITA CMS opens a map, you can configure the objects that do not need to be loaded automatically when a map is open.

For example, you could configure the DITA CMS so that submaps are not automatically loaded when you open a map. They get loaded only when you open them from the DITA Map view.

To configure the objects that should not be loaded:

1. **Open the TEXTML Administration view.**
2. **Connect to your server and Content Store.**
3. **In the repository's `/system/conf` collection, check out the `display.xml` file and open it with a text editor.**
4. **Locate the `<maps>` element.**

For example:

```
...
<maps>
  <references>
    <unexpanded class="- map/map container/container "/>
    <unexpanded class="- map/map container/container containerpart/containerpart "/>
  </references>
  <container class="- map/map container/container ">
    <subcontainer xpath="//*[contains(@class, ' mapgroup-d/mapref ')]" attrib="href"/>
  </container>
</maps>
```

By default, the DITA Map view does not load the containers, which are used in the Dynamic Release Management module.

**5. To configure an object so that it doesn't get loaded, add its class to the `<references>` element.**

For example, to prevent submaps from being loaded when you open a map, add the map class to the `<references>` element, as follows:

```
...
<references>
  <unexpanded class="- map/map "/>
```

**6. Save, close, and check in the `display.xml` file.**

**7. Synchronize the configuration.**

