

# DITACMS

## **DITA CMS Release 4.2.31: Detailed Release Notes**

# Table of contents

<b>About this document</b>	
<b>Configuration updates (Mandatory)</b>	
Import the new configuration files	4
Update existing configuration files	5
<b>Global changes</b>	
Set a status for documents that are retranslated from source	8
Use the Dita2ReviewPdf transformation scenario with the DITA-OT 2.1.1	12
Run the integrator on the DITA-OT	14
Add taxonomy terms to multiple documents	15
Configure the conditions used in the DITA CMS	16
Define a centralized coloring scheme for highlighting conditional text	19
Working with the eclipseui_template.xml file	21
Using preset list of keys	22
Configure a preset list of keys	22
Add keys to a map from preset list defined in the Content Store	23
Inserting a keyword or external link from keys defined in the map	24
<b>DRM-specific changes</b>	
Enable Clone Version command for closed versions	25

# About this document

This document describes the new features and bug fixes introduced in this build of DITA CMS Release 4.2.

# Configuration updates (Mandatory)

To support the new features in this release, new configuration files must be added to the Content Store and some existing configuration files must be updated.

## Import the new configuration files

---

If you are updating an existing installation to 4.2.31, you need to add four new configuration files to the Content Store to keep your deployment up to date.

The following files must be added to the *system/conf* folder:

- keydefnames.dtd
- keydefnames.xml
- eclipseui\_template.xml

Also, a new folder and file must be added to the *system* folder:  
*system/oxygen\_cfg/oxygen.default.options.xml*.

To facilitate this update, the new configuration files have been packaged in a .zip file and made available on the IXIASOFT Downloads website.

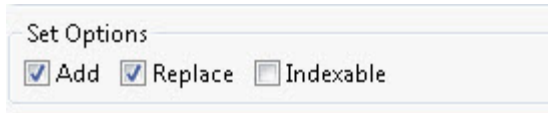
To import the files:

1. **Download the *Build4.2.31\_newfiles.zip* file containing the new configuration files from the following URL: [http://cms.ixiasoft.com/downloads/system\\_config/4.2/](http://cms.ixiasoft.com/downloads/system_config/4.2/)**
2. **Extract the .zip file to a local folder.**
3. **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**
  - a) **Select *Window > Open Perspective > Other***
  - b) **Click TEXTML Administration.**
  - c) **Click OK.**
4. **In the TEXTML Administration view, double-click the server. If your server is not displayed in the view, you must add it to the view.**
5. **When the Connect as dialog opens, type your username and password and click OK.**
6. **Double-click the name of your docbase to open a connection to the Content Store.**
7. **Right-click Repository and click Insert Documents.**
8. **Click Add Folder.**
9. **In the Browse For Folder dialog box, navigate to where you extracted the *Build4.2.31\_newfiles.zip* file and click the *system* folder. Click OK.**

In the **Set As** pane, the path should appear as */system/*.

**10. In the Set Options pane, select the Add and Replace checkboxes and clear Indexable.**

For example:



**11. Click OK.**

The files are imported into the Content Store.

## Update existing configuration files

---

If you are updating an existing installation to 4.2.31, you need to add information to two existing configuration files to keep your deployment up to date.

To update the configuration files:

1. **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**
  - a) **Select *Window > Open Perspective > Other***
  - b) **Click TEXTML Administration.**
  - c) **Click OK.**
2. **In the TEXTML Administration view, double-click the server. If your server is not displayed in the view, you must add it to the view.**
3. **When the Connect as dialog opens, type your username and password and click OK.**
4. **Double-click the name of your docbase to open a connection to the Content Store.**
5. **Expand the Content Store's Repository node and browse to */system/catalogs/* to locate the *catalog-cms-system.xml* file.**
6. **Right-click *catalog-cms-system.xml* and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **In the section `<group xml:base=" ../conf/">` add the following line:**

```
<public publicId="-//ixiasoft.com//cms//config//keydefnames" uri="keydefnames.dtd"/>
```

9. **Save, close, and check in the *catalog-cms-system.xml* file.**
10. **Browse to */system/conf/* to locate the *accessrights.xml* file.**
11. **Right-click *accessrights.xml* and click Check Out.**
12. **Double-click the file to open it in the XML editor area.**

**13. Add new access rights to the *accessrights.xml* file:****a) Locate the following line:**

```
<!-- ***** API METHODS DO NOT MODIFY ***** -->
```

**b) Just before this line, add the following `<method>` code to set access rights for the Edit Variables and Insert Variables actions:**

```
<!-- Possible type front-end / api -->
<method name="Edit Variables" type="front-end" hide-when-disabled="true">
  <!-- determines if a notification email should be sent when this action is taken -->

  <notify enabled="false"/>
  <!-- To be enabled or runnable the condition set must return true -->
  <!-- possible operator are any (One of the condition must match)-->
  <conditionset operator="any">
    <condition>
      <current>
        <type name="map">
          <statuses>
            <status>Authoring:*</status>
          </statuses>
        </type>
      </current>
      <!-- Action user must be in this list -->
      <users>
        <groups>
          <group name="System Administrators"/>
          <group name="Everyone"/>
        </groups>
        <roles/>
      </users>
    </condition>
  </conditionset>
</method>
<!-- Possible type front-end / api -->
<method name="Insert Variables" type="front-end" hide-when-disabled="true">
  <!-- determines if a notification email should be sent when this action is taken -->

  <notify enabled="false"/>
  <!-- To be enabled or runnable the condition set must return true -->
  <!-- possible operator are any (One of the condition must match)-->
  <conditionset operator="any">
    <condition>
      <current>
        <type name="map">
          <statuses>
            <status>Authoring:*</status>
          </statuses>
        </type>
      </current>
      <!-- Action user must be in this list -->
      <users>
        <groups>
          <group name="System Administrators"/>
          <group name="Everyone"/>
        </groups>
        <roles/>
      </users>
    </condition>
  </conditionset>
</method>
```

- c) **If you are not using the roles "System Administrators" or "Everyone", update the roles or groups for all the methods as appropriate for your deployment.**
- d) **In the `<availablemethods>` element, add the following actions:**

```
<action name="Edit Variables"/>
<action name="Insert Variables" />
```

- 14. Save, close, and check in the `accessrights.xml` file.**
- 15. Inform users of the changes and request that they close and reopen their DITA CMS to apply the changes.**

# Global changes

This section contains topics that apply to all deployments of the DITA CMS, including DRM deployments.

## Set a status for documents that are retranslated from source

---

You can now specify the status that documents will be in after a **Retranslate from source** operation, which reverts the content of one or more objects to the original source language.

This feature applies to the Sequential Localization Manager only.

The following list summarizes the steps required to implement this feature:

- Configure the status in the *localizationManagers.xml* file.
- Add the status in the following workflow files:
  - *topic\_status.xml*
  - *map\_status.xml*
  - *image\_status.xml*
- Add the status to the `prepareLocalizationPackage_api` method in the *accessrights.xml* file.

To implement this feature:

1. **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**
  - a) **Select *Window > Open Perspective > Other***
  - b) **Click TEXTML Administration.**
  - c) **Click OK.**
2. **In the TEXTML Administration view, double-click the server. If your server is not displayed in the view, you must add it to the view.**
3. **When the Connect as dialog opens, type your username and password and click OK.**
4. **Double-click the name of your doctype to open a connection to the Content Store.**
5. **Configure the status in the *localizationManagers.xml* file:**
  - a) **Locate the *localizationManagers.xml* file in the repository's */system/conf* collection.**
  - b) **Check out and open the *localizationManagers.xml* file with a text editor.**



- c) In the "com.ixiasoft.cms.controller.localization.sequential.SequentialLocalizationManagerService" manager class, add the `retranslated.state` parameter and set it to the new state.

**Note:** In DRM, this class is called "com.ixiasoft.cms.controller.localization.sequential.SequentialLocalizationManagerServiceWithContainer".

For example:

```
<manager
class="com.ixiasoft.cms.controller.localization.sequential.SequentialLocalizationManagerService"

  name="Sequential Localization manager">
  <config name="substantialChange">false</config>
  <config name="full.context">false</config>
  <config name="context.document.type">Dita2Pdf</config>
  <config name="include.default.images">true</config>
  <config name="in.context.exact.matches">false</config>
  <config name="translatable.attributes">alt,navtitle</config>
  <config name="retranslated.state">retranslated</config>
</manager>
```

- d) Save, close, and check in the `localizationManagers.xml` file.

## 6. Add the status in the `prepareLocalizationPackage_api` method:

- Locate the `accessrights.xml` file in the repository's `/system/conf` collection.
- Check out and open the `accessrights.xml` file with a text editor.
- Look for the line that starts as follows:

```
<method name="prepareLocalizationPackage_api"
```

- d) In the `<statuses>` element, add the following element and specify the new status:

```
<status>Localization:retranslated</status>
```

For example:

**Note:** Your configuration may differ from the example below.

```
<method name="prepareLocalizationPackage_api" type="api">
  <notify enabled="false"/>
  <conditionset operator="any">
    <condition>
      <current>
        <type name="*">
          <statuses>
            <status>Localization:tb translated</status>
            <status>Localization:in translation</status>
            <status>Localization:retranslated</status>
          </statuses>
        </type>
      </current>
```

```

    <users>
      <groups>
        <group name="Everyone"/>
      </groups>
    </users>
  </condition>
</conditionset>
</method>

```

e) **Save, close, and check in the `accessrights.xml` file.**

## 7. Add the new status in the `topic_status.xml` file:

- a) **Locate the `topic_status.xml` file in the repository's `/system/conf` collection.**
- b) **Check out and open the `topic_status.xml` file with a text editor.**
- c) **In the localization cycle, add a `<state>` element for the new retranslated state, as follows:**

```

<state level="0" name="new_state" type="work">
  <lockable>
    <objtypes>
      <type>none</type>
    </objtypes>
  </lockable>
  <nextStates>
    <next>in translation</next>
  </nextStates>
</state>

```

For example:

```

<state level="0" name="retranslated" type="work">
  <lockable>
    <objtypes>
      <type>none</type>
    </objtypes>
  </lockable>
  <nextStates>
    <next>in translation</next>
  </nextStates>
</state>

```

- d) **Make sure that the next state corresponds to the state for topics that are currently being translated.**

For example, in the sample code above, the next state is `in translation`.

e) **Save, close, and check in the `topic_status.xml` file.**

## 8. Add the new status in the `image_status.xml` file:

- a) **Locate the `image_status.xml` file in the repository's `/system/conf` collection.**
- b) **Check out and open the `image_status.xml` file with a text editor.**

- c) **In the localization cycle, add a `<state>` element for the new retranslated state, as follows:**

```
<state level="0" name="new_state" type="work">
  <lockable>
    <objtypes>
      <type>none</type>
    </objtypes>
  </lockable>
  <nextStates>
    <next>in translation</next>
  </nextStates>
</state>
```

For example:

```
<state level="0" name="retranslated" type="work">
  <lockable>
    <objtypes>
      <type>none</type>
    </objtypes>
  </lockable>
  <nextStates>
    <next>in translation</next>
  </nextStates>
</state>
```

- d) **Make sure that the next state corresponds to the state for topics that are currently being translated.**

For example, in the sample code above, the next state is `in translation`.

- e) **Save, close, and check in the `image_status.xml` file.**

9. **Add the new status in the `map_status.xml` file:**

- a) **Locate the `map_status.xml` file in the repository's `/system/conf` collection.**
- b) **Check out and open the `map_status.xml` file with a text editor.**
- c) **In the localization cycle, add a `<state>` element for the new retranslated state, as follows:**

```
<state level="0" name="new_state" type="work">
  <lockable>
    <objtypes>
      <type>none</type>
    </objtypes>
  </lockable>
  <nextStates>
    <next>in translation</next>
  </nextStates>
</state>
```

For example:

```
<state level="0" name="retranslated" type="work">
  <lockable>
    <objtypes>
```

```

        <type>none</type>
    </objtypes>
</lockable>
<nextStates>
    <next>in translation</next>
</nextStates>
</state>

```

- d) **Make sure that the next state corresponds to the state for topics that are currently being translated.**

For example, in the sample code above, the next state is `in translation`.

- e) **Save, close, and check in the `map_status.xml` file.**

10. **Inform users of the changes and request that they close and reopen their DITA CMS to apply the changes.**

## Use the Dita2ReviewPdf transformation scenario with the DITA-OT 2.1.1

---

By default, the Dita2ReviewPdf transformation scenario works with the DITA-OT 1.8.5, but you can also use it with the DITA-OT 2.1.1.

To use the Dita2ReviewPdf transformation scenario with the DITA-OT 2.1.1:

1. **Create a new client conductor file based on the**

`%OutputGenDir%/data/conductor-client20.xml.orig` **template.**

This template ensures that the `outgen.ot.version` and `outgen.ot.release.family` properties are set correctly at the top of the file. For example:

```

<!-- Set the version of the OT that will be used -->
<property name="outgen.ot.version" value="2.1.1"/>
<property name="outgen.ot.release.family" value="2.x"/>

```

For example, create a file called `%OutputGenDir%/data/conductor-acme20.xml`.

2. **Open your client conductor file (for example, `%OutputGenDir%/data/conductor-acme20.xml`).**
3. **Open the DITA CMS `%OutputGenDir%/data/conductor-ixia.xml` conductor file.**  
This file contains the default `dita2reviewpdfwrapper` target.
4. **Copy the contents of the `dita2reviewpdfwrapper` target to your client conductor file.**
5. **Edit the target name to indicate that the target was customized for your company.**

IXIASOFT recommends that you add your company name before the target name. For example:

```
<target name="acme.dita2reviewpdfwrapper" depends="default_dependencies, flatten,
resolve_container_keyref, extract_HiRes_images">
...

```

6. **Save and close your client conductor file.**
7. **Open your client preprocessor file (`%OutputGenDir%/conf/client/preprocessors.xml`).**
8. **Open the DITA CMS `%OutputGenDir%/conf/preprocessor.xml` file.**
9. **Copy the contents of the `dita2reviewpdf` preprocessor from the DITA CMS preprocessor file (i.e., `%OutputGenDir%/conf/preprocessor.xml`) to your client preprocessor file (i.e., `%OutputGenDir%/conf/client/preprocessors.xml`).**
10. **In your client preprocessor file, set the `buildfile` and `target` parameters to point to the new target.**

For example:

```
<preprocessor name="dita2reviewpdf"
class="com.ixiasoft.outputgenerator.preprocessor.AntProcessor">
  <parameters>
    <system>
      <parameter name="buildfile" value="/conductor-acme20.xml"/>
      <parameter name="target" value="acme.dita2reviewpdfwrapper"/>
      <parameter name="clean" value="job_postprocess"/>
      <parameter name="keep.log.files" value="false"/>
      <!-- "keep.log.files" must be false otherwise it will not work -->
    </system>
  </parameters>
</preprocessor>

```

## 11. Run the DITA-OT integrator.

- On Windows: In the `%OutputGenDir%/data/` directory, double-click the `_DITA-OT-2.1.1-integrator.bat` file.
- On Linux: In the `%OutputGenDir%/data/%OT_Dir%/bin` directory, enter the following command:

```
./ant -f ../integrator.xml (Linux)
```

## 12. To test the transformation scenario, restart the Output Generator service.

## Run the integrator on the DITA-OT

---

Any time you make a change to the DITA-OT, you need to run the integrator.

For example, if you add new plugins, make changes to existing plugins, or use a different version of the DITA-OT for your transformation scenarios, you need to run the integrator on the DITA-OT.

IXIASOFT provides two batch files in the *%OutputGenDir%/data/* directory to help you run the integrator, according to the version of the DITA-OT. They are:

- *\_DITA-OT-1.8.5-integrator.bat*: Runs the integrator on the DITA-OT 1.8.5
- *\_DITA-OT-2.1.1-integrator.bat*: Runs the integrator on the DITA-OT 2.1.1

**Note:** If you are using a different version of DITA-OT, you can create your own batch file by making a copy of one of the existing files above and update the DITA-OT version inside the file to point to the correct DITA-OT path.

### To run the integrator on the DITA-OT:

DITA-OT 1.x family:	<ul style="list-style-type: none"> <li>• On Windows: In the <i>%OutputGenDir%/data/</i> directory, double-click the <i>_DITA-OT-1.8.5-integrator.bat</i> file.</li> <li>• On Linux:             <ol style="list-style-type: none"> <li>1. In the <i>%OutputGenDir%/data/%OT_Dir%/</i> directory, run <code>startcmd.sh</code>.</li> <li>2. At the command prompt, enter:                 <pre>ant -f integrator.xml</pre> </li> </ol> </li> </ul> <p>The integration build runs and you should soon see BUILD SUCCESSFUL.</p>
DITA-OT 2.x family:	<ul style="list-style-type: none"> <li>• On Windows: In the <i>%OutputGenDir%/data/</i> directory, double-click the <i>_DITA-OT-2.1.1-integrator.bat</i> file.</li> <li>• On Linux: In the <i>%OutputGenDir%/data/%OT_Dir%/bin</i> directory, enter the following command:             <pre>./ant -f ../integrator.xml (Linux)</pre> </li> </ul> <p>The integration build runs and you should soon see BUILD SUCCESSFUL.</p>

## Add taxonomy terms to multiple documents

---

The **Taxonomy Terms** view allows you to add terms to multiple documents at once.

The documents in the **Documents** pane must be locked before you can make changes to the taxonomy terms. If documents are not locked when you attempt to make a change, a message may appear to request permission to lock the documents. If you click **OK** to lock the documents, the documents are locked regardless of their status.

Once you click **Apply**, the taxonomy changes are applied to the documents. Documents that were in a 'lockable' status such as **Authoring:work** or the equivalent remain locked. Documents with an 'unlockable' status such as **Authoring:done** or the equivalent that were locked by the system are handled differently. When **Apply** is clicked, the document is released, a comment describing the change is automatically added to the Revision History, and the status remains unchanged. If any of the documents cannot be locked, an error message displays and the operation is canceled. To continue, either change the status of the affected documents so that they can be locked or remove them from the **Documents** pane.

Terms are added to the document's `<metadata>` section as keywords or categories, depending on how the system is configured.

To add terms to all the documents in the **Documents** pane:

1. **Open the Taxonomy Terms view.**
2. **Select the documents:**
  - a) **Locate the documents; for example, perform a search, open a map, or display them in a view.**
  - b) **Drag the documents into the Documents pane in the Taxonomy Terms view.**
3. **In the Taxonomy pane, click one or more taxonomy terms.**
4. **Right-click the term(s) and click Add Term(s) to All Documents.**

If a message appears requesting to lock the documents, click **OK**.
5. **Click Apply.**

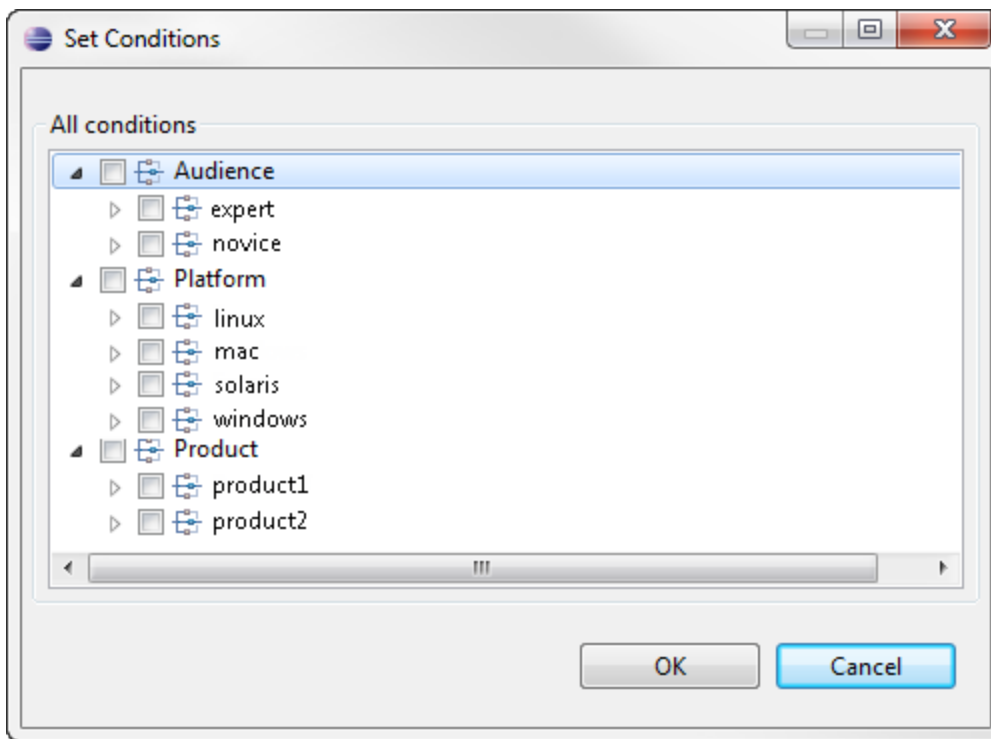
**Important:** Changes are not applied to the documents until you click **Apply**. Do not release documents before clicking **Apply** or the changes will be lost.
6. **Release the documents, as required.**

## Configure the conditions used in the DITA CMS

Conditional attributes let users tailor the same document for different audiences and products. The DITA CMS supports the DITA attributes that let you hide portions of text—or even entire topics—that do not apply to a specific audience, product, platform, etc.

Users can apply conditions on objects in a map or elements in topics, and then specify the conditions to display when they generate their output.

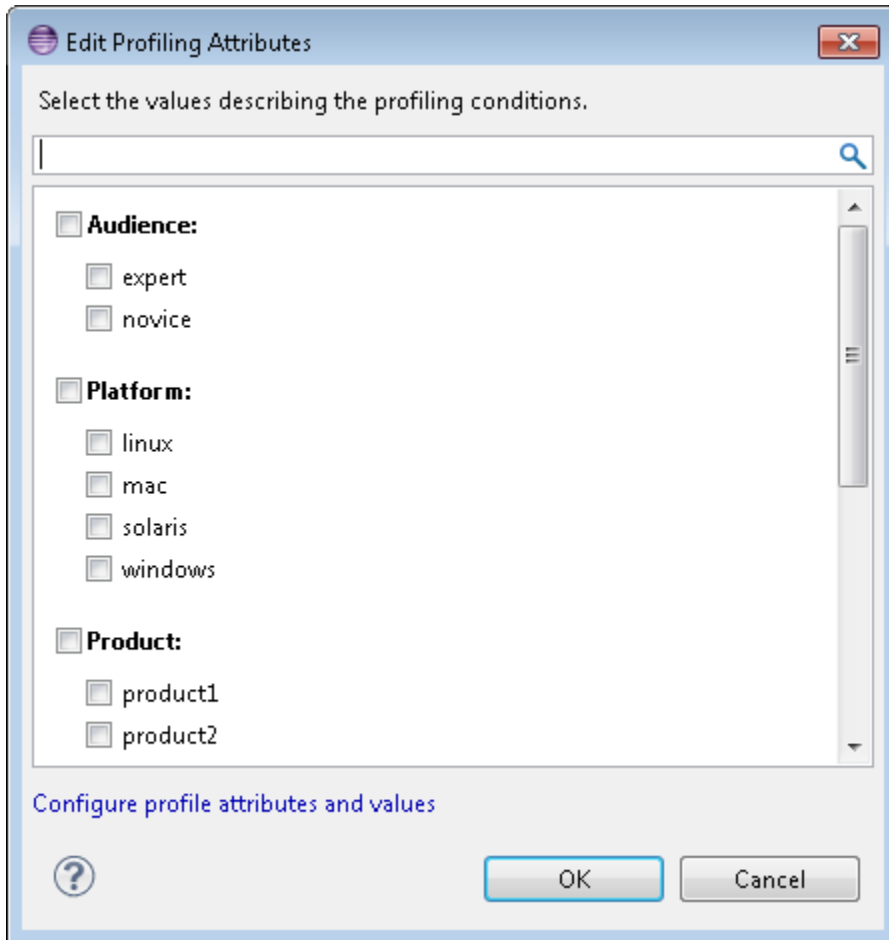
For example, when users click the **Set Conditions** button in the **DITA Map** view, the **Set Conditions** window lists the conditions that are currently configured in the CMS, grouped by attribute, as shown in the following screenshot.



**Figure 1: Example of the Set Conditions dialog box**

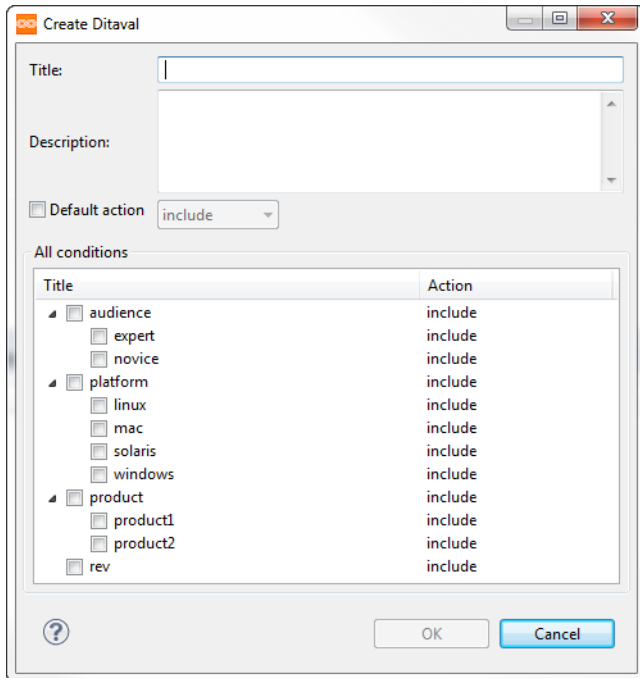
When users right-click highlighted content in a topic in Author mode (in the oXygen XML Editor) and click **Edit Profiling Attributes**, the dialog box lists the available conditions:





**Figure 2: Example of the Edit Profiling Attributes dialog box**

Also, when users create Ditaval files through the DITA CMS, a list of conditions is displayed in the **Create Ditaval** dialog box.



**Figure 3: Example of the Create Ditaval dialog box**

As the DITA CMS system administrator, you need to configure the conditions that will be listed in these dialogs. You configure this information in the *conditionaltext.xml* file.

To configure conditions:

1. **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**
  - a) **Select *Window > Open Perspective > Other***
  - b) **Click TEXTML Administration.**
  - c) **Click OK.**
2. **In the TEXTML Administration view, double-click the server. If your server is not displayed in the view, you must add it to the view.**
3. **When the Connect as dialog opens, type your username and password and click OK.**
4. **Double-click the name of your docbase to open a connection to the Content Store.**
5. **Right-click *conditionaltext.xml* and click Check Out.**
6. **Double-click the file to open it in the XML editor area.**
7. **For each condition to display in the dialogs, create a `<category>` entry, specifying the following information:**

```
<condtextsettings>
  <category name="condition_name" xmlattribute="attribute_name">
```

```

    <allowedvalue>value</allowedvalue>
  </category>
</condtextsettings>

```

Where:

Option	Description
<i>condition_name</i>	Name of the condition as displayed in the dialogs
<i>attribute_name</i>	Name of the XML conditional attribute as defined by DITA 1.2
<i>value</i>	Value that can be set for this condition. Create one <allowedvalue> element per valid value.

For example, to create the dialogs shown above, you would have the following code:

```

<condtextsettings>
  <category name="Audience" xmlattribute="audience">
    <allowedvalue>expert</allowedvalue>
    <allowedvalue>novice</allowedvalue>
  </category>
  <category name="Platform" xmlattribute="platform">
    <allowedvalue>linux</allowedvalue>
    <allowedvalue>mac</allowedvalue>
    <allowedvalue>solaris</allowedvalue>
    <allowedvalue>windows</allowedvalue>
  </category>
  <category name="Product" xmlattribute="product">
    <allowedvalue>product 1</allowedvalue>
    <allowedvalue>product 2</allowedvalue>
  </category>
</condtextsettings>

```

8. **Save, close, and check in the conditionaltext.xml file.**
9. **Inform users of the changes and request that they close and reopen their DITA CMS to apply the changes.**

## Define a centralized coloring scheme for highlighting conditional text


---

You can configure DITA CMS to display the conditional text in different colors and styles in Author mode and in a PDF output generated without selecting a ditaval.

Instead of each user configuring a personalized profiling color scheme for oXygen XML Editor (or Author) in **Window > Preferences**, you can define a scheme in the Content Store which automatically becomes available to DITA CMS Eclipse clients which have never been configured with a profiling color scheme. If users have defined profiling styles for their DITA CMS Eclipse

client (or have had a previously defined style), then the profiling styles defined in the Content Store are ignored by their DITA CMS Eclipse client.

A template for defining the profiling color scheme is available in **system > oxygen\_cfg**. You can build the color scheme using the code in the template; however, it is recommended that you define the scheme in the **Preferences** dialog box and then export it for use in DITA CMS.

If the profiling colors and styles are not visible in **Author** mode, click **Profiling/Conditional Text** toolbar menu  and then click **Show Profiling Colors and Styles**.

To define the color scheme for conditional text:

1. **Open *Window > Preferences*.**
2. **In the left pane, go to *oXygen XML Editor (or oXygen XML Author) > Editor > Edit Modes/Pages > Author > Profiling/Conditional Text > Colors and Styles*.**
3. **In the Colors and Styles pane, import your conditions from a ditaval file or create them as required.**
4. **Click a profiling attribute value and click Edit.**
5. **Define the style as desired and click OK.**
6. **Repeat for each profiling attribute value as required.**
7. **Export the global options for oXygen XML Editor:**
  - a) **In the left pane, click oXygen XML Editor (or oXygen XML Author).**
  - b) **Click Export Global Options**
  - c) **Save to your local disk.**
8. **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**
  - a) **Select *Window > Open Perspective > Other***
  - b) **Click TEXTML Administration.**
  - c) **Click OK.**
9. **In the TEXTML Administration view, double-click the server. If your server is not displayed in the view, you must add it to the view.**
10. **When the Connect as dialog opens, type your username and password and click OK.**
11. **Double-click the name of your docbase to open a connection to the Content Store.**
12. **Expand the Content Store's Repository node and browse to */system/oxygen\_cfg/* to locate the *oxygen.default.options.xml* file.**
13. **Right-click *oxygen.default.options.xml* and click Check Out.**

**14. Double-click the file to open it in the XML editor area.**

If this is the first time you open the file, it contains an example of a defined color scheme for a profiling attribute value.

**15. Open the file containing the exported global options.**

The profiling color scheme is defined inside the `<profilingAttributeStyle-array>` element.

**16. Copy the contents of the `<profilingAttributeStyle-array>` element.****17. In the `oxygen.default.options.xml` file, replace the existing contents by pasting the new contents into the `<profilingAttributeStyle-array>` element.****18. Save, close, and check in the `oxygen.default.options.xml` file.****19. Inform users of the changes and request that they close and reopen their DITA CMS to apply the changes.**

## Working with the `eclipseui_template.xml` file

---

DITA CMS provides a template that you can use to customize your `eclipseui.xml` file.

You can find the latest version of this file in the system configuration package, available at the following URL:

[http://cms.ixiasoft.com/downloads/system\\_config/4.2/](http://cms.ixiasoft.com/downloads/system_config/4.2/)

The system configuration package is provided in a zip file (`dita-cms-config-<build>.zip`) for the latest version of DITA CMS. The `eclipseui_template.xml` file is available at the following location in the zip file:

```
\\config-base\system\conf\eclipseui_template.xml
```

Use this file as an example for customizing the `eclipseui.xml` file for your deployment. The `eclipseui_template.xml` file contains examples for many perspectives and menus available in DITA CMS. You can copy code from the template to your own file and customize it as necessary.

The `eclipseui_template.xml` file is updated regularly, so make sure to download the latest version when customizing the interface for your deployment.

## Using preset list of keys

---

You can now configure a preset list of common or corporately-approved keys (keywords or external links) in the Content Store. Once the list is configured, the keys can be added to map open in the DITA Map view.

### Configure a preset list of keys

Defining a preset list of keys is useful when you want to enforce consistency for the keywords or external links used in a documentation set.

Rather than each user creating their own keys, you can define a list of the commonly-used or company-approved keys that users can access in the **DITA Map** view and insert into their maps.

The list is defined in the *keydefnames.xml* configuration file in *system/conf*. The file contains two types of keys:

- Keywords, which are listed in the following section:

```
<type label="Keyword" name="keyword">
```

- External links, which are listed in the following section:

```
<type label="External links" name="xref">
```

Each key is defined as a `<keydefname>` element in the appropriate section. Each `<keydefname>` element contains attributes defining the name of the key and its default value. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE keydefnames PUBLIC "-//ixiasoft.com//cms//config//keydefnames" "keydefnames.dtd">
<keydefnames>
  <type label="Keyword" name="keyword">
    <keydefname name="company" default-value="IXIASOFT"/>
  </type>
  <type label="External links" name="xref">
    <keydefname name="companyweb" default-value="www.ixiasoft.com"/>
  </type>
</keydefnames>
```

To define the list of keys:

1. **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**
  - a) **Select *Window > Open Perspective > Other***
  - b) **Click TEXTML Administration.**
  - c) **Click OK.**

2. In the TEXTML Administration view, double-click the server. If your server is not displayed in the view, you must add it to the view.
3. When the Connect as dialog opens, type your username and password and click OK.
4. Double-click the name of your docbase to open a connection to the Content Store.
5. Expand the Content Store's Repository node and browse to `/system/conf/` to locate the `keydefnames.xml` file.
6. Right-click `keydefnames.xml` and click Check Out.
7. Double-click the file to open it in the XML editor area.
8. For each new keyword or external link that you want to specify, insert a new line in the appropriate section using the format `<keydefname name="[key name]" default-value="[key value]"/>`.

For example, if you wanted to create a key for the company's name, you would add the following in the section labeled keyword:

```
<keydefname name="company" default-value="IXIASOFT"/>
```

9. Save, close, and check in the `keydefnames.xml` file.
10. Inform users of the changes and request that they close and reopen their DITA CMS to apply the changes.

## Add keys to a map from preset list defined in the Content Store

Although several ways are available for including keys in a map, the DITA CMS Administrator may have defined a list of keywords and external links which you can insert into your map.

To add preset keys to the map :

1. In the DITA Map view, open and lock the map to which you want to add the keys.
2. Right-click the map and click Edit Variables.
3. Select the checkboxes next to the keywords or external links that you want to add.
4. Click OK.

If you cannot see the keys in the **DITA Map** view, make sure the view is configured to display the map elements.



## Inserting a keyword or external link from keys defined in the map

You can insert a key into a topic from those defined in the map.

If a map contains submaps and a key in a submap shares the same name with a key defined in the parent map, the key definition in the parent map prevails. When you click **Insert variable from map**, it inserts a keyword (`<keyword keyref="[value]" />`) in the topic. When you click **Insert external link from map**, it inserts a link (`<xref keyref="[value]" />`) in the topic.

**Restriction:** Glossary keys are not available using this dialog box.

To insert the variable or link:

1. **Place your cursor at the location in your topic where you want to insert the key.**
2. **Do one of the following depending on the type of key you want to insert:**
  - To insert a keyword using oXygen XML Editor, click the  button from the main toolbar.
  - To insert a link using oXygen XML Editor, click the  button from the main toolbar.
  - To insert a keyword using XMetaL XML Editor, right-click and click **Insert variable from map**.
  - To insert a link using XMetaL XML Editor, right-click and click **Insert external link from map**.
3. **In the Variable column, click the variable that you want to insert.**
4. **Click OK.**



# DRM-specific changes

This section contains topics that apply to DRM deployments only.

## Enable Clone Version command for closed versions

---

You can now run the Clone Version command on versions that are closed. This change requires configuration updates.

**Note:** This procedure applies to deployments with the DRM module only.

To enable the Clone Version command for versions that are closed:

1. **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**
  - a) **Select *Window > Open Perspective > Other***
  - b) **Click TEXTML Administration.**
  - c) **Click OK.**
2. **In the TEXTML Administration view, double-click the server. If your server is not displayed in the view, you must add it to the view.**
3. **When the Connect as dialog opens, type your username and password and click OK.**
4. **Double-click the name of your docbase to open a connection to the Content Store.**
5. **Locate the *accessrights.xml* file in the repository's */system/conf* collection.**
6. **Check out and open *accessrights.xml* with a text editor.**
7. **Locate the line that starts with the following:**

```
<method name="CloneDRMVersion"
```

8. **Change the `<status>` element for the version type to `<status>Authoring: *</status>`.**

For example (in bold):

```
<method name="CloneDRMVersion" type="front-end">
...
  <conditionset operator="any">
    <condition>
      <current>
        <type name="version">
          <statuses>
            <status>Authoring: *</status>
          </statuses>
        </type>
      </current>
    </condition>
  </conditionset>
</method>
```

This ensures that the Clone Version command can be applied to versions that are in the Authoring cycle.

9. **Save, close, and check in the *accessrights.xml* file.**
10. **Ask users to close all container and container parts and to restart their DITA CMS Eclipse Client.**

# Index

## A

Ant targets 12  
    Modify in conductor files 12

## C

conditionaltext.xml 16  
Conditions, configure 16

Configure conditions 16

## D

Ditaval values, configure 16

## M

Modify DITA CMS Ant targets 12