Upgrading the DITA CMS Core DRM from 4.4 to 4.5

Last revised: December 22, 2017

Table of contents

Before you start

This document describes how to upgrade your DITA CMS 4.4 deployment to the latest DITA CMS Core 4.5 release.

Before you start the upgrade, you must be running the latest build of the DITA CMS Core version 4.4 and that you have applied all the version 4.4 configuration changes. If you have not completed all the configuration changes for version 4.4, please follow all the instructions in **Preparing your deployment for the upgrade** on page 11.

Preparation before the upgrade

Beginning with the DITA CMS Core 4.5 release, the legacy Web Collaborative Reviewer and Web Author products are not longer supported products in the deployment. As a result, some additional steps must be performed before the Core components are upgraded. Please see **Summary of upgrade path** on page 12 for more information.

Upgrading the Core components

To upgrade the DITA CMS, you need to upgrade all the components.

Table 1: DITA CMS Core components

Step	Component	Procedure
1.	Content Store	**Upgrade your Content Store to current release** on page 91
2.	DITA CMS Eclipse Client	**Upgrade the DITA CMS Eclipse Client** on page 101
3.	Output Generator	**Install a new Output Generator and upgrade your configuration** on page 103
4.	Scheduler	**Upgrade the Scheduler** on page 113

Upgrading to DITA 1.3

If you want to upgrade to DITA 1.3, please see **Upgrading a Deployment to DITA 1.3 Guide**

Copyright notice

1 Preparing your deployment for the upgrade

Topics:

With the end of support for the legacy DITA CMS products, Web Author and Web Collaborative Reviewer, some additional steps are required to upgrade your deployment to the current DITA CMS Core release version.

Summary of upgrade path

To help you understand what you need to do to prepare for the upgrade, identify the starting state of your deployment to obtain the list of tasks you must complete before starting the upgrade.

You must evaluate each of the following sections to identify the effort required to perform the upgrade.

Part 1: Remove legacy products

Beginning with the DITA CMS Core 4.5 release, the legacy Web Collaborative Reviewer and Web Author products are not longer supported products in the deployment.

Starting state of your deployment	Tasks required BEFORE performing the upgrade to Core 4.5
If your deployment contains the legacy products Web Author or Web Collaborative Reviewer	• Uninstall Tomcat 6. • Uninstall GlassFish 3.2. • Remove the Web Collaborative Reviewer plugins/dropins from the DITA CMS Eclipse Client instances. • Remove the Web Author configuration files from the Content Store. These instructions are provided in **Removing the legacy products** on page 15.

Part 2: Prepare your Content Store

Your Content Store must be upgraded to the last release before performing the upgrade to this release. If your Content Store is not upgraded to Core 4.4.46, you must apply all the configuration changes required for each release between the build you have in your deployment and Core 4.4.46.

Starting state of your Core components	Tasks required BEFORE performing the upgrade to Core 4.5
If your Content Store is at Core 4.2.31 or lower	Please contact IXIASOFT Services for assistance.

Starting state of your Core components	Tasks required BEFORE performing the upgrade to Core 4.5
If your Content Store is at one of the versions from Core 4.2.33 to 4.2.53	1. Backup your Content Store. 2. Follow each of the instructions to update your Content Store configuration for each release from your existing version to Core 4.2.53. See **Updating a Core 4.2.X to 4.2.53** on page 22. 3. Follow the instructions to upgrade your Content Store from Core 4.2.53 to Core 4.3.61. See **Upgrade Core 4.2.53 to latest 4.3 release** on page 48. 4. Follow the instructions to upgrade your Content Store from Core 4.3.61 to Core 4.4.46. See **Upgrade Core 4.3.61 to latest 4.4 release** on page 74. 5. Proceed with the upgrade to Core 4.5. See **Upgrade your Content Store to current release** on page 91.
If your Content Store is at one of the Core 4.3.X versions	1. Backup your Content Store. 2. Follow each of the instructions to update your Content Store configuration for each release from your existing version to Core 4.3.61. See **Updating a Core 4.3.X to 4.3.61** on page 36. 3. Follow the instructions to upgrade your Content Store from Core 4.3.61 to Core 4.4.46. See **Upgrade Core 4.3.61 to latest 4.4 release** on page 74. 4. Proceed with the upgrade to Core 4.5. See **Upgrade your Content Store to current release** on page 91.
If your Content Store is at one of the Core 4.4.X versions	1. Backup your Content Store.

Starting state of your Core components	Tasks required BEFORE performing the upgrade to Core 4.5
	2. No configuration updates are required. Proceed with the upgrade to Core 4.5. See **Upgrade your Content Store to current release** on page 91.

Note: If you are not sure which starting state is relevant to you, please contact IXIASOFT Support team.

Part 3: Prepare your DITA CMS Web component

The new DITA CMS Web component requires a change to your IXIASOFT license model. Please contact IXIASOFT Sales to find out how your organization can update its license to use DITA CMS Web.

For the full instructions for how to upgrade your existing DITA CMS Web component, see *Upgrading the DITA CMS Web from 4.5 to 4.6 Guide* on the DITA CMS Web page on **http://www.ixiasoft.com/en/products/dita-cms/documentation**.

Table 2: Web upgrade path

Starting state of your Web component	Tasks required BEFORE performing the upgrade to Web 4.6
If the webplatform folder in your Content Store only contains the *web.platform.config.xml* file	1. Contact IXIASOFT Sales team to obtain your license for DITA CMS Web. 2. Import the webroles files. 3. Remove the Xeditor configuration from the Content Store. 4. Ensure that the properties.txt file is present in the location where you installed DITA CMS Web. 5. Once you have the license, proceed with the upgrade to Web 4.6.
If the webplatform folder in your Content Store contains the *web.platform.config.xml*, *webroles.xml*, and *webroles.dtd* files	1. Contact IXIASOFT Sales team to obtain your license for DITA CMS Web.

Starting state of your Web component	Tasks required BEFORE performing the upgrade to Web 4.6
	2. Once you have the license, proceed with the upgrade to Web 4.6.

Removing the legacy products

Since the legacy Web Collaborative Reviewer and Web Author products are not longer supported products in the deployment, they must be removed.

The following provides an overview of the tasks that are required before you begin the upgrade.

1. Uninstall the Oracle GlassFish Server
2. Uninstall Apache Tomcat 6.0
3. Remove the legacy Collaborative Reviewer plugin or dropin from your instances of the DITA CMS Eclipse Client
4. Remove the legacy Web Author configuration from the Content Store

Uninstall Oracle GlassFish Server

To uninstall the server:

1. Stop the CMS Application Server:

- On Windows: Click the Start button and select ***All Programs*** > ***Oracle GlassFish Server*** > ***Stop Application Server***.
- On Linux: Open a command prompt and type the following command:

```
/sbin/service glassfish.cmsappserver stop
```

2. Remove the program:

- On Windows: Open a command prompt using **Run as administrator** and type:

```
sc delete cmsappserver-glassfish
```

- On Linux: Open a command prompt and type:

```
chkconfig --del cmsappserver.glassfish
rm -f /etc/init.d/cmsappserver.glassfish
```

3. Delete the installation folder:

- On Windows: Delete the Oracle GlassFish installation folder; for example, *C:\glassfish3*.
- On Linux: In the command prompt, type:

```
rm -rf /opt/ixiasoft/glassfish
```

Uninstall Apache Tomcat 6.0 Application Server

To uninstall the server:

1. Stop the Tomcat service:

- On Windows: On the machine where the product is installed, click ***Start*** > ***Control Panel*** > ***Administrative Tools*** and double-click **Services**. In the **Services** window, click `Apache Tomcat` in the **Name** column. In the left pane, click the **Stop** link.
- On Linux: Go to the %TomcatDir%/bin/ directory and type the following command:

```
[root@t19c6264 bin]# ./shutdown.sh
```

2. Remove the program:

- On Windows: On the machine where the product is installed, click ***Start*** > ***Control Panel*** and open the **Uninstall or change program** dialog box. Right-click **Apache Software Foundation Tomcat 6 (remove only)** and click **Uninstall**.
- On Linux: Open a command prompt and type:

```
chkconfig --del tomcat6
rm -f /etc/init.d/tomcat6
```

3. Delete the installation folder:

- On Windows: Delete the Apache Tomcat installation folder; for example: `c:\Program Files\Apache Software Foundation`
- On Linux: In the command prompt, type:

```
rm -rf /opt/ixiasoft/apache-tomcat6.0.xx
```

Remove the Collaborative Reviewer plugin when installed through an update site

If the Collaborative Reviewer plugin is installed in the DITA CMS Eclipse Client through an update site, you need to remove the update site from the list available from each instance of the DITA CMS Eclipse Client.

To remove the update site:

1. **Open the DITA CMS Eclipse Client.**
2. **In the main menu, click *Help > About Eclipse SDK*.**
3. **Click the Installation Details.**
4. **Click the Installed Software tab.**
5. **Click Collaborative Reviewer and click the Uninstall button.**

Remove the Collaborative Reviewer plugin when installed as a dropin

If the Collaborative Reviewer plugin is installed in the DITA CMS Eclipse Client as a dropin, you need to delete the dropin **from each instance** of the DITA CMS Eclipse Client.

To delete the dropin:

1. **Close the DITA CMS Eclipse Client.**
2. **Delete the Collaborative Reviewer dropin; for example:**
   C:\eclipses\cms43_admin_x64\dropin\ com.ixiasoft.eclipse.collaborative.reviewer-4.x.xxxx

Remove the Web Author configuration from the Content Store

To delete the Web Author configuration:

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 the *system* folder to locate the *cms.webauthor* folder.**
6. **Create a backup of the folder. If you customized the frameworks, the content of the folder might be useful for DITA CMS Web. To create a backup:**
   a) **Right-click the *cms.webauthor* folder and click Export.**
   b) **In the Destination folder, type the location where you want to save the folder or click Browse.**
   c) **Select the Content checkbox.**
   d) **Click OK.**
7. **Right-click the *cms.webauthor* folder and click Delete.**

Prepare your Content Store for the upgrade

If your Content Store is not upgraded to Core 4.4.46, you must apply all the configuration changes required for each release between the build you have in your deployment and Core 4.4.46.

To help you understand what you need to do to prepare for the upgrade, identify the starting state of your Core components in the following table to obtain the list of tasks you must complete before starting the upgrade.

Starting state of your Core components	Tasks required BEFORE performing the upgrade to Core 4.5
If your Content Store is at Core 4.2.31 or lower	Please contact IXIASOFT Services for assistance.
If your Content Store is at one of the versions from Core 4.2.33 to 4.2.53	**1.** Backup your Content Store. **2.** Follow each of the instructions to update your Content Store configuration for each release from your existing version to Core 4.2.53. See **Updating a Core 4.2.X to 4.2.53** on page 22. **3.** Follow the instructions to upgrade your Content Store from Core 4.2.53 to Core 4.3.61. See **Upgrade Core 4.2.53 to latest 4.3 release** on page 48.

Starting state of your Core components	Tasks required BEFORE performing the upgrade to Core 4.5
	4. Follow the instructions to upgrade your Content Store from Core 4.3.61 to Core 4.4.46. See **Upgrade Core 4.3.61 to latest 4.4 release** on page 74. 5. Proceed with the upgrade to Core 4.5. See **Upgrade your Content Store to current release** on page 91.
If your Content Store is at one of the Core 4.3.X versions	1. Backup your Content Store. 2. Follow each of the instructions to update your Content Store configuration for each release from your existing version to Core 4.3.61. See **Updating a Core 4.3.X to 4.3.61** on page 36. 3. Follow the instructions to upgrade your Content Store from Core 4.3.61 to Core 4.4.46. See **Upgrade Core 4.3.61 to latest 4.4 release** on page 74. 4. Proceed with the upgrade to Core 4.5. See **Upgrade your Content Store to current release** on page 91.
If your Content Store is at one of the Core 4.4.X versions	1. Backup your Content Store. 2. No configuration updates are required. Proceed with the upgrade to Core 4.5. See **Upgrade your Content Store to current release** on page 91.

Back up your Content Store

If you haven't already done so, make a backup copy of your Content Store.

Perform an on-demand backup of the docbase

If you need to manually trigger a backup of the docbase, you can launch the backup from the TEXTML Server interface.

> **Note:** Content Store data is stored in a TEXTML Server docbase. To back up a Content Store, you back up the TEXTML Server docbase that stores the Content Store data.

To trigger a backup:

1. **On the computer where the TEXTML Server is installed, create a directory for the backups.**
2. **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.**
3. **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.**
4. **When the Connect as dialog opens, type your username and password and click OK.**
5. **Double-click the name of your docbase to open a connection to the Content Store.**
6. **Right-click the docbase and select Backup.**

   The **Start backup** window opens:

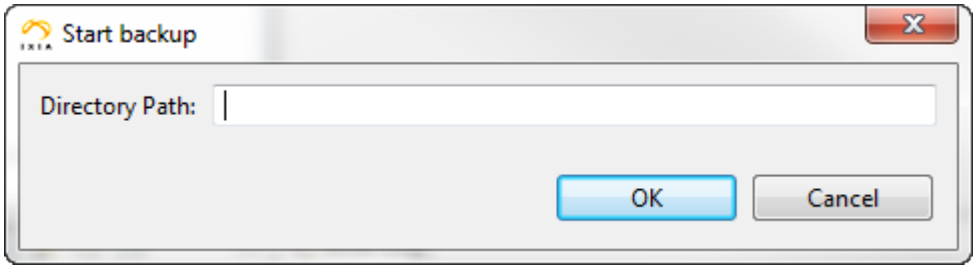


Figure 1: Backing up a docbase

Important note:
BEWARE: If the destination folder is not empty, TEXTML Server deletes the contents of the folder before starting the backup.

7. **In the Directory Path box, type the full path to the folder you created for the docbase backups.**

   For example (replace `[version]` by the release version) :

   ```
   C:\docbase_backups\[version]\docbaseA
   ```

   ```
   /docbase_backups/[version]/docbaseA
   ```

   Note: The TEXTML Server service or daemon ***must*** have write privileges to the top-level folder storing the backups; for example, the *docbase_backups* directory.

8. **Click OK.**

   The backup is initiated. To verify, look in the **Properties** view under ***General*** > ***Status***. When the backup is initiated, the value for **Status** changes from **Ready** to **Ready Backing-up**.

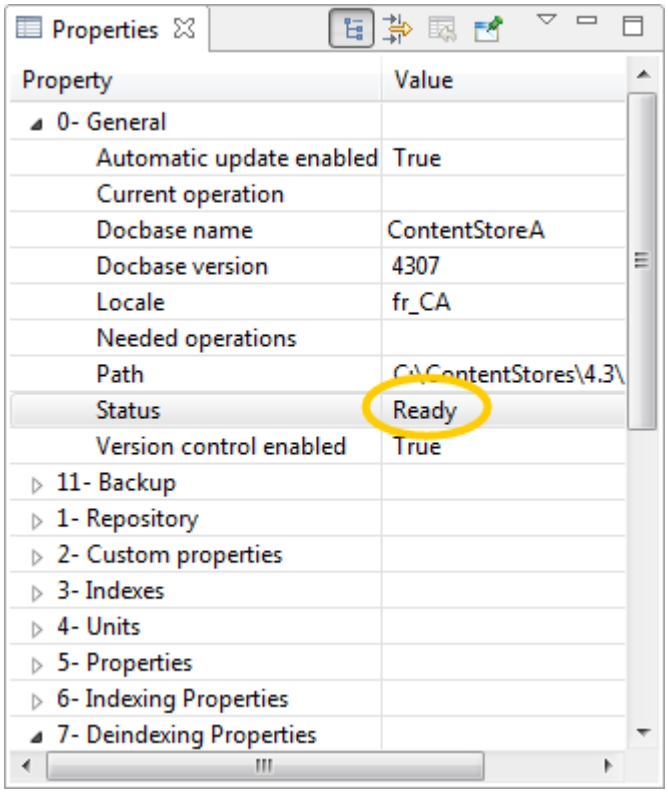


Figure 2: Example of Properties view:

9. **Wait while the backup occurs. Depending on the size of the docbase and your server activity, this could take considerable time.**

   Note: While the backup is occurring, a temporary file is created (*backuperror.tmp*) under the backup folder. If no errors occur during backup, this file is deleted and a new file called *backup.info* is created when the backup is complete.

10. To confirm that the backup was successful, look in your target directory:

- If the *backup.info* file is present, then the backup was successful.
- It the backup is completed but the *backuperror.tmp* is present, then an error occurred and the backup is corrupted.

If an error occurs, or to get more information about the backup operation, look at the logs. They are available in the log directory.

On Windows:

```
%program_data%\IxiaSoft\TextmlServer[TEXTML Server version]\Log
```

For example:

```
C:\ProgramData\IxiaSoft\TextmlServer[TEXTML Server version]\Log
```

On Linux:

```
<textml_install_dir>/<instance_name>/log/
```

For example:

```
/opt/ixiasoft/textmlserver/ts01/log/
```

Updating a Core 4.2.X to 4.2.53

This section contains the instructions to apply all the configuration changes required for each release between the 4.2.X build you have in your deployment to Core 4.2.53.

Apply build-specific configuration updates

When a new DITA CMS build is released, it sometimes requires changes to the configuration.

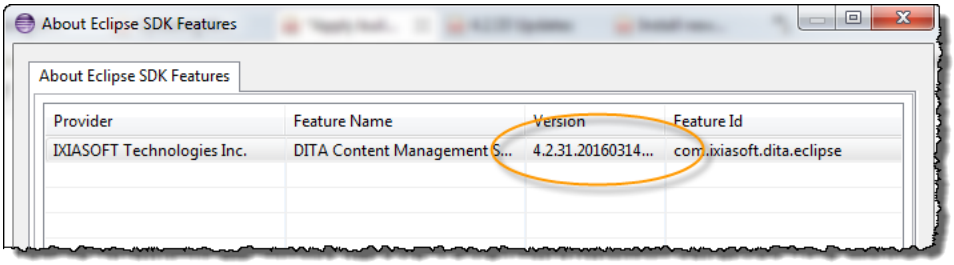
The following table lists the configuration updates currently available:

Table 3: Configuration updates

Build Number	Configuration updates to apply
4.2.53	• **Update the Index Definition document** on page 24
4.2.45	• **Replace the project_message.xsl template in Scheduler** on page 25

Build Number	Configuration updates to apply
	• **Update the schedule.xml file** on page 27 • **Edit the snapshot_status.xml configuration file** on page 27
4.2.38	• **Updating the accessrights.dtd to include new statuschange method** on page 28
4.2.34	• **Upgrade custom Build Manifest templates** on page 30 • **DRM only: Update the Index Definition document** on page 31
4.2.33	• **Download and import configuration files** on page 32 • **Update existing configuration files** on page 33 • **DRM-only: Update the Index Definition document** on page 35
4.2.25 to 4.2.31	Please contact IXIASOFT Services team for assistence.

1. **Determine your current build number as follows:**
   a) **In the DITA CMS Eclipse Client, click *Help > About Eclipse SDK* from the menu bar.**
   The About Eclipse SDK dialog is displayed.
   b) **Click the IXIASOFT icon ( ).**
   The About Eclipse SDK features dialog displays the version number of the DITA CMS. For example, the following diagram shows that DITA CMS 4.2.31 is installed.



2. **Apply all the configuration changes for the builds that were released after your current build version.**
   For example, if you're at version 4.2.38 of the DITA CMS and you're upgrading to version 4.2.53, then you need to apply the configuration changes for 4.2.45 and 4.2.53.

4.2.53 updates

Perform the following procedures to update your Content Store to 4.2.53.

In this build, you have the following changes to make (all deployments):

- **Update the Index Definition document** on page 24

For deployments using DRM, you have the following additional changes to make:

- No changes required.

Update the Index Definition document

If you are updating an existing installation to 4.2.53, you need to update the Index Definition document to keep your deployment up to date.

> **Note:** IXIASOFT recommends that you change the Index Definition document during off hours only. This operation will require a reindexing of the modified indexes, which will reduce the performance of the TEXTML Server.

When adding new indexes to the index definition, IXIASOFT recommends that you add them in alphabetical order.

To update the Index Definition document:

1. **Open a TEXTML Server Console Java.**
2. **Connect to the server and Content Store to upgrade.**
3. **Expand the Content Store node to display the Index Definition.**
4. **Right-click Index Definition and select Lock.**
5. **Open the Index Definition document in an XML editor.**
6. **Search for the `<index NAME="rootClass">` element and add the attribute `SYNC="True"` to it.**

   For example:

```
<index NAME="rootClass" SYNC="True">
 <!-- System index required by the DITA CMS -->
 <stringindex KEEPEXTRACTEDVALUES="True">
  <elements>
   <element DEPTH="INFINITE" XPATH="/*/@class"/>
   <element DEPTH="INFINITE" XPATH="/dita/*/@class"/>
  </elements>
 </stringindex>
</index>
```

7. **Under `<indexes>` add the following `<index>` element:**

```
<index NAME="reusable_fulltext">
   <!-- System index required by the DITA CMS -->
   <admindescription>Fulltext index on the documents</admindescription>
   <wordindex>
      <elements>
         <element DEPTH="INFINITE" XPATH="//rcbody/*"/>
      </elements>
   </wordindex>
</index>
```

8. **Right-click the Index Definition document and select Check In.**

4.2.45 updates

Perform the following procedures to update your Content Store to 4.2.45.

In this build, you have the following changes to make (all deployments):

- **Replace the project_message.xsl template in Scheduler** on page 25
- **Update the schedule.xml file** on page 27
- **Edit the snapshot_status.xml configuration file** on page 27

For deployments using DRM, you have the following additional changes to make:

- No changes required.

Replace the project_message.xsl template in Scheduler

You only need to replace the *project_message.xsl* template if you have installed Scheduler in your deployment.

To replace the project_message.xsl template:

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/scheduler/template* **to locate the** *project_message.xsl* **file.**

6. **Right-click** *project_message.xsl* **and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **Delete the contents of the file.**
9. **Copy the following text and paste it in the file.**

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
 <xsl:output method="html" version="1.0" encoding="UTF-16" indent="yes"/>
 <xsl:template match="/message">
  <html>
   <head>
    <style type="text/css">
     body, th, td {
         font-family: Verdana, Arial, Sans-serif;
      font-size: 10pt;
     }
     .title {
      color: blue;
         font-size: 14pt;
      padding: 20px 0px 20px 0px;
     }
     .infoSending{

     }
    </style>
   </head>
   <body>
    <div class="infoSending"> Message received from: <strong><xsl:value-of
select="//author"/></strong>
    on <xsl:value-of select="//date"/></div>
   <br/>
    <div class="title">
     <xsl:value-of select="//title"/>
    </div>
    <br/>
    <xsl:value-of select="//body"/>
   </body>
  </html>
 </xsl:template>
</xsl:stylesheet>
```

10. **Save the file.**
11. **Save, close, and check in the file.**
12. **Inform users of the changes.**

    The changes will be applied automatically once users close and then reopen their DITA CMS. Users can also apply the changes without restarting their DITA CMS by clicking ***DITA CMS*** > ***Synchronize Configuration***.

Update the schedule.xml file

A new job was added to the *schedule.empy.xml* file. You must add this job to your *schedule.xml* file.

> **Note:** You only need to perform this task if you have installed Scheduler in your deployment.

To update the *schedule.xml* file:

1. **Open the** *%SchedulerDir%/conf/schedule.xml* **file.**
2. **Look for the closing `</jobs>` and `</postmaster>` tags at the end of the file.**
3. **Just before the closing `</jobs>` tag, add the following code:**

```
<!-- Take the project Messages in the scheduler/outbox, send them as mail to the team
members and move them in
 the scheduler/sentitems-->
<job useClass="com.ixiasoft.cms.postman.jobs.ProjectMessageJob" enable="true" >
   <schedule>
      <when>*/5 * * * *</when><!-- this will run every 5 minutes -->
   </schedule>
   <configuration>
      <property name="transform.xsl" value="project_message.xsl"/>
   </configuration>
</job>
```

4. **Save and close the file.**

Edit the snapshot_status.xml configuration file

A small change is required in the *snapshot_status.xml* to enable snapshots of a map in localization to have the initial state from the Localization cycle rather than the Authoring cycle.

To edit the snapshot_status.xml configuration file:

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** *snapshot_status.xml* **file.**

6. **Right-click** *snapshot_status.xml* **and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **Locate `collection="/content/localization/"` and replace it with `collection="/content/snapshots/".`**

   For example:

```
<cycle collection="/content/snapshots/" description="" initial="false" lastworkcycle="false"
 level="0" name="Localization" type="localization">
```

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

4.2.38 updates

Perform the following procedures to update your Content Store to 4.2.38.

In this build, you have the following changes to make (all deployments):

- **Updating the accessrights.dtd to include new statuschange method** on page 28

For deployments using DRM, you have the following additional changes to make:

- No changes required.

Updating the accessrights.dtd to include new statuschange method

If you are updating an existing installation to 4.2.38, you must edit the accessrights.dtd configuration file to keep your deployment up to date. You must add the statuschange method to the accessrights.dtd to support the new feature which allows administrators to restrict the visibility of status options by group.

To update the file:

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** *accessrights.dtd* **file.**
6. **Right-click** *accessrights.dtd* **and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **Locate the following section:**

```
<!ELEMENT method (notify, conditionset)                                    >
<!ATTLIST method
name CDATA #REQUIRED
alt  CDATA #IMPLIED
type (front-end | api | outputtype) #REQUIRED
hide-when-disabled (true | false ) "false"
multiselect-disabled (true | false ) "false"
use-specialization (true | false) "false"                          >
```

9. **Add `statuschange` to the type attribute as shown below:**

```
<!ELEMENT method (notify, conditionset)                                    >
<!ATTLIST method
name CDATA #REQUIRED
alt  CDATA #IMPLIED
type (front-end | api | outputtype | statuschange) #REQUIRED
hide-when-disabled (true | false ) "false"
multiselect-disabled (true | false ) "false"
use-specialization (true | false) "false"                          >
```

10. **Save, close, and check in the file.**
11. **Inform users of the changes and request that they close and reopen their DITA CMS to apply the changes.**

4.2.34 updates

Perform the following procedures to update your Content Store to 4.2.34.

In this build, you have the following changes to make (all deployments):

- **Upgrade custom Build Manifest templates** on page 30

For deployments using DRM, you have the following additional changes to make:

- **DRM only: Update the Index Definition document** on page 31

Upgrade custom Build Manifest templates

If you created custom Build Manifest templates in DITA CMS version 4.2.31 or earlier, you need to add the `<locstatusend/>` element to your templates.

This element ensures that only the localized content that is in the Localization:done status (or the equivalent in your deployment) is produced when you output a Build Manifest in multiple languages.

> **Note:** You do not need to perform this procedure on the standard DITA CMS template (that is, */system/templates/bmanifests/build-manifest.bmanifest*). This template was updated automatically when you applied configuration changes in 4.2.31.

If the `<locstatusend/>` element is not added to your templates, the following error appears when creating a new Build Manifest based on the custom template:

```
The content of element type "globals" must match
"(shortdesc,mode,locstatusend,mapref,languages,rootcontext,ditaval,properties,notify?).
```

> **Note:** Existing Build Manifests will get updated automatically when you next modify them. You do not need to update your existing Build Manifests. Only the Build Manifest templates must be updated.

To add the `<locstatusend/>` element to your existing Build Manifest templates:

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. **Navigate to the repository's** */system/templates/bmanifests* **collection.**
6. **For each custom Build Manifest template that you created:**
   a) **Right-click the template and select Check Out.**
   b) **Open the template with a text editor.**
   c) **Locate the `<mapref>` element.**

d) **Just before the <mapref> element, add the following line:**

```
  <locstatusend/>
```

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE build-manifest PUBLIC "-//IXIA//DTD DITA CMS Build Manifest//EN"
"build-manifest.dtd">
<build-manifest xml:lang="en-US" id="id">
 <title>Build Manifest Title</title>
 <globals>
  <shortdesc>Build Manifest Description</shortdesc>
  <mode otherprops="PRODUCTION"/>
  <locstatusend/>
  <mapref></mapref>
  <languages>
   ...
```

Note: Leave the <locstatusend/> element empty in the template. When you release a Build Manifest, the appropriate final localization status (for example, Localization:done) is extracted from your configuration and added to the Build Manifest.

e) **Save, close, and check in the template file.**

f) **Repeat this step for all your custom Build Manifest templates.**

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

DRM only

This section describes configuration changes that apply to DRM deployments only.

DRM only: Update the Index Definition document

If you are updating an existing DRM installation to 4.2.34, you need to update the Index Definition document to keep your deployment up to date.

Note: IXIASOFT recommends that you change the Index Definition document during off hours only. This operation will require a reindexing of the modified indexes, which will reduce the performance of the TEXTML Server.

To update the Index Definition document:

1. **Open a TEXTML Server Console Java.**
2. **Connect to the server and Content Store to update.**
3. **Expand the Content Store node to display the Index Definition.**
4. **Make a backup of the Index Definition document:**

a) **Right-click Index Definition and select Export.**
b) **In the Save As dialog, specify the directory where to save the backup and click Save.**

5. **Right-click Index Definition and select Lock.**
6. **Open the Index Definition document in an XML editor.**
7. **Under <summaries>\<summary NAME="fullsummary">, add the following <field> elements:**

```
<field NAME="CreationDate" TYPE="Property" VALUE="All"/>
<field NAME="CreationTime" TYPE="Property" VALUE="All"/>
```

8. **Save the Index Definition document.**
9. **Right-click the Index Definition document and select Check In.**

4.2.33 updates

Perform the following procedures to update your Content Store to 4.2.33.

In this build, you have the following changes to make (all deployments):

- **Download and import configuration files** on page 32
- **Update existing configuration files** on page 33

For deployments using DRM, you have the following additional changes to make:

- **DRM-only: Update the Index Definition document** on page 35

Download and import configuration files

To update an existing installation to 4.2.33, you need to add new configuration files to the Content Store and update existing ones to keep your deployment up to date.

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

- *templateparameters.dtd*
- *templateparameters.xml*

Also, the following files must be updated:

- *system/catalogs/catalog-cms-system.xml*
- *system/plugins/com.ixiasoft.dita.dtd/dtd/build-manifest.mod*
- *system/templates/bmanifests/build-manifest.bmanifest*

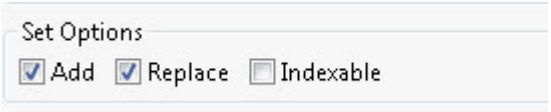
To facilitate this update, these files have been packaged in a .zip file and made available on the IXIASOFT Downloads website.

To import the files:

1. **Download the** *Build_4.2.33_newfiles.zip* **file, which contains the configuration files, from the following URL: 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** *Build_4.2.33_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

To update an existing installation to 4.2.33, you need to add information to the 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/conf* **to locate the** *localizationManagers.xml* **file.**
6. **Right-click** *localizationManagers.xml* **and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **In the Sequential Localization Manager section under `SequentialLocalizationManagerService` class add the following lines:**

```
<!-- <config name="import.translations.skip.review">true</config> -->
<!-- <config
name="translation.autotranslation.fulltranslation.status">Localization:translated</config>
 -->
```

For example:

```
<!-- Sequential Localization Manager, with auto-translation disabled -->
<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="import.translations.skip.review">true</config>-->
 <!-- <config
name="translation.autotranslation.fulltranslation.status">Localization:translated</config>
 -->
 <!-- <config name="retranslated.state"></config> -->
</manager>

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

 name="Sequential Image Localization manager"
 type="image">
</manager>
```

9. **Save, close, and check in the** *localizationManagers.xml* **file.**

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

DRM only

This section describes configuration changes that apply to DRM deployments only.

DRM-only: Update the Index Definition document

To update an existing DRM installation to 4.2.33, you need to update the Index Definition document to keep your deployment up to date.

> **Note:** IXIASOFT recommends that you change the Index Definition document during off hours only. This operation will require a reindexing of the modified indexes, which will reduce the performance of the TEXTML Server.

To update the Index Definition document:

1. **Open a TEXTML Server Console Java.**
2. **Connect to the server and Content Store to update.**
3. **Expand the Content Store node to display the Index Definition.**
4. **Make a backup of the Index Definition document:**
   a) **Right-click Index Definition and select Export.**
   b) **In the Save As dialog, specify the directory where to save the backup and click Save.**
5. **Right-click Index Definition and select Lock.**
6. **Open the Index Definition document in an XML editor.**
7. **Look for the `keyref-contentDependencies` index.**

   For example:

```
<index CUSTOMPROPERTY="True" NAME="keyref-contentDependencies" SYNC="True">
   <!-- System index required by the DITA CMS -->
   <stringindex KEEPEXTRACTEDVALUES="True">
      <elements>
         <element DEPTH="0" XPATH="//reference[@contentDependency='true'][@keyref or
@conkeyref]/@srcRef"/>
         <!-- Patch for 4.0 and 4.1 legacy content -->
        <element DEPTH="0" XPATH="//reference[@contentDependency='false'][contains(@class,
 ' map/topicref ')][@keyref]/@srcRef"/>
        <element DEPTH="0" XPATH="//reference[@contentDependency='false'][contains(@class,
 ' topic/image ')][@keyref]/@srcRef"/>
         <element DEPTH="0" XPATH="for $e in
//reference[@contentDependency='false'][@conkeyref] return substring-before($e/@srcRef,
'/')"/>
      </elements>
   </stringindex>
</index>
```

8. **Replace the two lines in bold above with the following:**

```
<element DEPTH="0"
XPATH="/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][contains(@class,
 ' map/topicref ')][@keyref]/@srcRef"/>
<element DEPTH="0"
XPATH="/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][contains(@class,
 ' topic/image ')][@keyref]/@srcRef"/>
```

The updated index will look as follows:

```
<index CUSTOMPROPERTY="True" NAME="keyref-contentDependencies" SYNC="True">
   <!-- System index required by the DITA CMS -->
   <stringindex KEEPEXTRACTEDVALUES="True">
      <elements>
         <element DEPTH="0" XPATH="//reference[@contentDependency='true'][@keyref or
@conkeyref]/@srcRef"/>
         <!-- Patch for 4.0 and 4.1 legacy content -->
         <element DEPTH="0"
XPATH="/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][contains(@class,
 ' map/topicref ')][@keyref]/@srcRef"/>
         <element DEPTH="0"
XPATH="/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][contains(@class,
 ' topic/image ')][@keyref]/@srcRef"/>
         <element DEPTH="0" XPATH="for $e in
//reference[@contentDependency='false'][@conkeyref] return substring-before($e/@srcRef,
'/')"/>
      </elements>
   </stringindex>
</index>
```

9. **Save the Index Definition document.**
10. **Right-click the Index Definition document and select Check In.**

Updating a Core 4.3.X to 4.3.61

This section contains the instructions to apply all the configuration changes required for each release between the 4.3.X build you have in your deployment to Core 4.3.61.

Apply build-specific configuration updates

When a new DITA CMS build is released, it sometimes requires changes to the configuration.

The following table lists the configuration updates currently available:

Table 4: Configuration updates

Build Number	Configuration updates to apply
4.3.61	**Update the Index Definition document** on page 38 **Edit the users.dtd file** on page 39

Build Number	Configuration updates to apply
4.3.57	**Download and import configuration files** on page 40
4.3.47	**Download and import configuration files** on page 42 **Edit the resources.xml file** on page 44 **Edit the advancedSearchIndexes.xml file (DRM only)** on page 45 **Recommended updates** on page 46
4.3.33	**Correct the doctype declaration in the pivotlanguages.xml file** on page 47
4.3.31	Initial 4.3 release

1. **Determine your current build number as follows:**
   a) **In the DITA CMS Eclipse Client, click *Help > About Eclipse SDK* from the menu bar.**
      The About Eclipse SDK dialog is displayed.
   b) **Click the Installation Details button.**
   c) **Click the Plug-ins tab.**
   d) **Double-click the Provider column header to sort the names and look for *IXIASOFT Technologies Inc.*.**
   e) **In the Plug-in Name column, look for *DITA Content Management System*.**
   f) **In the same row, look in the Version column, which lists the installed version number.**
2. **Apply all the configuration changes for the builds that were released after your current build version.**
   For example, if you were at version 1.1.0 and you wanted to update to version 1.1.5, then you would need to apply the configuration changes from each version from 1.1.1 to 1.1.5.

4.3.61 updates

Perform the following procedures to update your Content Store to 4.3.61.

In this build, you have the following changes to make (all deployments):

- **Update the Index Definition document** on page 38
- **Edit the users.dtd file** on page 39

For deployments using DRM, you have the following additional changes to make:

- None.

Update the Index Definition document

This procedure describes how to modify the Index Definition document to update your Content Store to the current build.

> **Note:** IXIASOFT recommends that you change the Index Definition document during off hours only. This operation will require a reindexing of the modified indexes, which will reduce the performance of the TEXTML Server.

When adding new indexes to the index definition, IXIASOFT recommends that you add them in alphabetical order.

To update the Index Definition document:

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 right-click the** *Index Definition* **file and select Lock.**
6. **Double-click the file to open it in the XML editor area.**
7. **Under <indexes>, locate the following <index> element:**

```
<index NAME="type" SYNC="True">
<!-- Index referenced by the Advanced Search on attributes -->
 <stringindex KEEPEXTRACTEDVALUES="False">
  <elements>
   <element DEPTH="INFINITE" XPATH="//@type"/>
  </elements>
 </stringindex>
</index>
```

8. **In the index, delete `SYNC="True"`. For example, it should look as follows:**

```
<index NAME="type">
<!-- Index referenced by the Advanced Search on attributes -->
 <stringindex KEEPEXTRACTEDVALUES="False">
```

```
  <elements>
   <element DEPTH="INFINITE" XPATH="//@type"/>
  </elements>
 </stringindex>
</index>
```

9. **Locate the following <index> element:**

```
<index NAME="wordCount" SYNC="True">
 <numericindex KEEPEXTRACTEDVALUES="True">
  <integerindexproperties/>
  <elements>
   <element DEPTH="0" XPATH="string-length(normalize-space(string(/))) -
string-length(replace(normalize-space(string(/)), ' ', ''))" />
  </elements>
 </numericindex>
</index>
```

10. **In the index, delete `SYNC="True"`. For example, it should look as follows:**

```
<index NAME="wordCount">
 <numericindex KEEPEXTRACTEDVALUES="True">
  <integerindexproperties/>
  <elements>
   <element DEPTH="0" XPATH="string-length(normalize-space(string(/))) -
string-length(replace(normalize-space(string(/)), ' ', ''))" />
  </elements>
 </numericindex>
</index>
```

11. **Save the Index Definition document.**
12. **Right-click the Index Definition document and select Check In.**

Edit the users.dtd file

A couple of small changes are required in the *users.dtd* file to prevent text from being added in the `<role>` and `<group>` elements.

To edit the *users.dtd* file:

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** *users.dtd* **file.**
6. **Right-click the file and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **Locate the element `<!ELEMENT role (#PCDATA)` and replace `(#PCDATA)` with `EMPTY`.**
   For example: `<!ELEMENT role EMPTY`
9. **Locate the element `<!ELEMENT group (#PCDATA)` and replace `(#PCDATA)` with `EMPTY`.**
   For example: `<!ELEMENT group EMPTY`
10. **Save, close, and check in the file.**

4.3.57 updates

Perform the following procedures to update your Content Store to 4.3.57.

In this build, you have the following changes to make (all deployments):

- **Download and import configuration files** on page 40

For deployments using DRM, you have the following additional changes to make:

- None.

Download and import configuration files

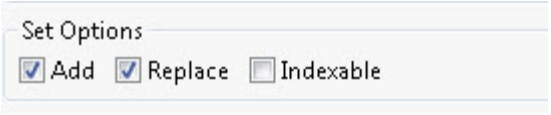
The update package contains the files necessary to update the Preview view to allow it to generate previews for topics containing either DITA 1.2 or DITA 1.3 elements.

1. **Go to the following URL:**
   http://cms.ixiasoft.com/downloads/system_config/4.3/
2. **Download the** *Build_4.3.57_updates.zip* **file to your computer.**
3. **Extract the .zip file to your local disk.**
4. **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.**
5. **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.**

6. **When the Connect as dialog opens, type your username and password and click OK.**
7. **Double-click the name of your docbase to open a connection to the Content Store.**
8. **If you customized your Preview view, make a backup of your *html* folder and some files in the *resource* folder.**
   a) **Expand the *Repository* node and browse to */system/xsl/html*.**
   b) **Right-click the *html* folder and click Rename.**
   c) **In the New Collection Name box, type a new name for the folder so it will not be overwritten when you import the new files. For example, *html_backup*.**
   d) **Click OK.**
   e) **Browse to the */system/xsl/resource* folder.**
   f) **Right-click the *commonltr.css* file and click Rename.**
   g) **In the New Document Name box, type a new name for the file so it will not be overwritten when you import the new files. For example, *commonltr.backup.css*.**
   h) **Click OK.**
   i) **Repeat the previous steps to rename the *commonrtl.css* and *messages.xml* files.**
9. **Right-click Repository node and click Insert Documents.**
10. **Click Add Folder.**
11. **In the Browse For Folder dialog box, browse to the location where you extracted the configuration files and select the *system* folder (for example, *C:\temp\Build_4.3.57_updates\config-base\system*). Click OK.**

    In the **Set As** pane, the path should appear as */system/*.
12. **In the Set Options pane, select the Add and Replace checkboxes and clear Indexable.**

    For example:

13. **Click OK.**

    The existing *html* folder is replaced.
14. **If you had customized your Preview view, use your backup copy to re-implement your changes to the Preview view. The Preview view is now based on the xhtml plugin of the DITA OT 2.3.1.**

4.3.47 updates

Perform the following procedures to update your Content Store to 4.3.47.

In this build, you have the following changes to make (all deployments):

- **Download and import configuration files** on page 42
- **Edit the resources.xml file** on page 44
- **Recommended updates** on page 46

For deployments using DRM, you have the following additional changes to make:

- **Edit the advancedSearchIndexes.xml file (DRM only)** on page 45

Download and import configuration files

To update an existing installation to 4.3.47, you need to update configuration files to keep your deployment up to date.

The following files must be updated:

- *system/conf/conditionaltext.dtd*
- *system/conf/languages.dtd*
- *system/plugins/com.ixiasoft.dita13.dtd/IxiaDitabase.dtd* (applies only to DITA 1.3 deployments)
- *system/plugins/com.ixiasoft.dita13.dtd/IxiaMap-drm.dtd* (applies only to DITA 1.3 deployments)

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

To import the files:

1. **Go to the following URL:**

   http://cms.ixiasoft.com/downloads/system_config/4.3/
2. **Download the** *Build_4.3.47_updates.zip* **file to your computer.**
3. **Extract the .zip file to a local working directory.**

   For example:

   ```
   C:\temp\
   ```

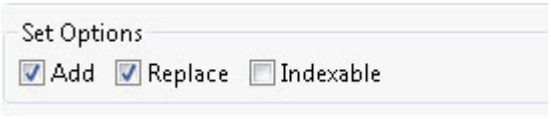
   The following folders are available:

   - `config-base`: Contains the files that must be updated in all deployments
   - `config-dita13`: Contains the files that must be updated in DITA 1.3 deployments only

4. **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.**
5. **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.**
6. **When the Connect as dialog opens, type your username and password and click OK.**
7. **Double-click the name of your docbase to open a connection to the Content Store.**
8. **Right-click Repository and click Insert Documents.**
9. **Click Add Folder.**
10. **In the Browse For Folder dialog box, browse to the** *config-base/system* **folder in the update directory (for example,** *C:\temp\Build_4.3.47_updates\config-base\system***) and click the** *system* **folder. Click OK.**

    In the **Set As** pane, the path should appear as */system/*.
11. **In the Set Options pane, select the Add and Replace checkboxes and clear Indexable.**

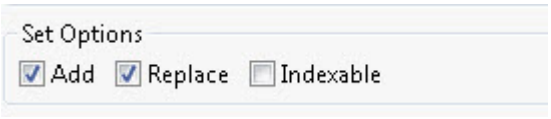
    For example:

12. **Click OK.**
13. **(For DITA 1.3 deployments only!) In the Browse For Folder dialog box, browse to the** *config-dita13/system* **folder in the update directory (for example,** *C:\temp\Build_4.3.47_updates\config-dita13\system***) and click the** *system* **folder. Click OK.**

    In the **Set As** pane, the path should appear as */system/*.
14. **In the Set Options pane, select the Add and Replace checkboxes and clear Indexable.**

    For example:

15. **Click OK.**

    The files are imported into the Content Store.

Edit the resources.xml file

If you are updating to 4.3.47, you need to add a new type to the *resources.xml* file.

When a topic containing references to non-DITA files such as PDF files or videos was imported, the non-DITA files were imported as resource objects of an undefined type, which resulted in them not being searchable from the Search view. To import non-DITA files as searchable resources, the *resources.xml* file must be edited to add a new type called "unknown".

Also, the ditaval-file type must be removed (if present), because ditaval files are not resources.

To edit the *resources.xml* file:

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** *resources.xml* **file.**
6. **Right-click the** *resources.xml* **file and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **Add the following line to the file:**

```
<type description="">unknown</type>
```

For example, the file should look as follows:

```
<resource>
 <resourcetypes>
  <type description="">pdf-graphics</type>
  <type description="">pdf-legal</type>
  <type description="">css</type>
  <type description="">html</type>
  <!-- Default type used by the import process -->
       <type description="">unknown</type>
 </resourcetypes>
</resource>
```

9. **If the `<type description="">ditaval-file</type>` line is present, remove it from the file.**
10. **Save, close, and check in the file.**
11. **Inform users of the changes.**

    The changes will be applied automatically once users close and then reopen their DITA CMS. Users can also apply the changes without restarting their DITA CMS by clicking ***DITA CMS*** > ***Synchronize Configuration***.

Edit the advancedSearchIndexes.xml file (DRM only)

If you are updating to 4.3.47, you need to add a new search index to the *advancedSearchIndexes* file.

This search index allows you to search with the filename in the **Search** view, in the **Search in** drop-down list.

To add the index:

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** *advancedSearchIndexes* **file.**
6. **Right-click the** *advancedSearchIndexes* **file and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **Add the following index to the file:**

```
<index display="Filename" listable="false" name="Name" type="property"
 views="ALL"/>
```

9. **Save, close, and check in the file.**
10. **Inform users of the changes.**

The changes will be applied automatically once users close and then reopen their DITA CMS. Users can also apply the changes without restarting their DITA CMS by clicking ***DITA CMS*** > ***Synchronize Configuration***.

Recommended updates

This section provides configuration updates that are recommended but not required.

In the access rights configuration, the methods of type `outputtype` (for example, Dita2Pdf, Dita2htmlhelp, Dita2EclipseHelp, Dita2xhtml, etc.) should be updated to replace the `<type name="*">` element with the specific object types to which they can be applied. This ensures that Build Manifests can be generated using the BuildManifest output type only (and not the Dita2Pdf output type, for example).

To fix this issue:

1. **Expand the Content Store's `Repository` node and browse to** */system/conf* **folder.**
2. **Right-click the accessrights.xml file and select Check Out.**
3. **Open the accessrights.xml file in an XML editor.**
4. **Look for methods with the following type: `type="outputtype".`**

   For example:

```
<method name="Dita2Pdf" type="outputtype">
 <notify enabled="false"/>
 <conditionset operator="any">
  <condition>
   <current>
    <type name="*">
     <statuses>
      <status>Authoring:*</status>
      <status>Localization:*</status>
      <status>Published:*</status>
     </statuses>
    </type>
    ...
```

5. **Replace the `<type name="*"> ... </type>` element with the following:**

```
<type name="snapshot">
 <statuses>
  <status>Authoring:*</status>
  <status>Localization:*</status>
 </statuses>
</type>
<type name="map">
 <statuses>
  <status>Authoring:*</status>
  <status>Localization:*</status>
  <status>Published:*</status>
 </statuses>
</type>
```

```
<type name="topic">
 <statuses>
  <status>Authoring:*</status>
  <status>Localization:*</status>
  <status>Published:*</status>
 </statuses>
</type>
<type name="image">
 <statuses>
  <status>Authoring:*</status>
  <status>Localization:*</status>
  <status>Published:*</status>
 </statuses>
</type>
```

This ensures that the output type is applied to snapshots, maps, topics, and images only.

6. **Repeat for all methods with the `type="outputtype"` type.**
7. **Save and check in the *accessrights.xml* file.**
8. **Inform users of the changes.**

   The changes will be applied automatically once users close and then reopen their DITA CMS. Users can also apply the changes without restarting their DITA CMS by clicking ***DITA CMS*** > ***Synchronize Configuration***.

4.3.33 updates

Perform the following procedures to update your Content Store to 4.3.33.

In this build, you have the following changes to make (all deployments):

- **Correct the doctype declaration in the pivotlanguages.xml file** on page 47

For deployments using DRM, you have the following additional changes to make:

- No changes required.

Correct the doctype declaration in the pivotlanguages.xml file

A small change is required in the *pivotlanguages.xml* file to correct the doctype declaration.

To edit the pivotlanguages.xml configuration file:

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** *pivotlanguages.xml* **file.**
6. **Right-click** *pivotlanguages.xml* **and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **Locate the doctype declaration.**

   For example:

```
<!DOCTYPE comments PUBLIC "-//ixiasoft.com//cms//config//pivotLanguages"
"pivotLanguages.dtd">
```

9. **Replace the word "*comments*" with "*pivotlanguages*".**

   For example:

```
<!DOCTYPE pivotlanguages PUBLIC "-//ixiasoft.com//cms//config//pivotLanguages"
"pivotLanguages.dtd">
```

10. **Save, close, and check in the file.**
11. **Inform users of the changes.**

    The changes will be applied automatically once users close and then reopen their DITA CMS. Users can also apply the changes without restarting their DITA CMS by clicking ***DITA CMS*** > ***Synchronize Configuration***.

Upgrade Core 4.2.53 to latest 4.3 release

This section contains the instructions to apply all the configuration changes required to upgrade your Core 4.2.53 Content Store to the latest build in the 4.3 release.

Download the DITA CMS Core 4.3 upgrade package

The DITA CMS 4.3 upgrade package contains files required to upgrade your Content Store.

To download the Core upgrade package:

1. **Go to the following URL:**

   http://cms.ixiasoft.com/downloads/4.2_to_4.3_upgrade/

2. **Copy the** *DITA_CMS_4.3_upgrade_package-<date>.zip* **file to your computer.**
3. **Unzip the upgrade package to a working directory.**

   For example:

```
C:\temp\
```

> **Note:** Throughout this document, the `%UpgradePackageDir%` expression is used to represent the upgrade package directory.

The upgrade package contains the following directories:

- `base-upgrade`: Contains the files required to upgrade to DITA CMS 4.3
- `drm-upgrade`: Contains the files required to upgrade the Dynamic Release Management module to 4.3
- `output-gen-upgrade`: Contains the files required to upgrade the Output Generator

Import and configure the Core 4.3 configuration files

This procedure describes how to import and configure the upgrade's configuration files for DITA CMS Core 4.3.

The configuration changes below are required to support the new 4.3 features and updates, such as:

- Subject schemes
- Incremental localization
- Pivot language localization
- Modifications to the *linguisticReview.xml* DTD
- Multi-level libraries
- Rename version

To upgrade your Content Store:

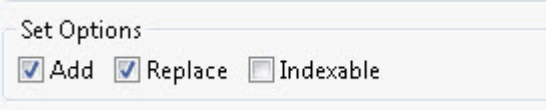
1. **Open a TEXTML Server Console Java.**
2. **Connect to the server and Content Store to upgrade.**
3. **Make a backup of the** *linguisticReview.xml* **file:**
   a) **Expand the Content Store's `Repository` node and browse to the** */system/conf* **folder.**
   b) **Right-click the** *linguisticReview.xml* **file and select Rename.**
   c) **In the New Document Name box, enter `linguisticReview.cms42.xml`.**
   d) **Click OK.**

4. **If you customized your Preview view, make a backup of your *html* folder and some files in the *resource* folder. The Preview view is now based on the xhtml plugin of the DITA OT 2.3.1.**
   a) **Expand the *Repository* node and browse to */system/xsl/html*.**
   b) **Right-click the *html* folder and click Rename.**
   c) **In the New Collection Name box, type a new name for the folder so it will not be overwritten when you import the new files. For example, *html_backup*.**
   d) **Click OK.**
   e) **Browse to the */system/xsl/resource* folder.**
   f) **Right-click the *commonltr.css* file and click Rename.**
   g) **In the New Document Name box, type a new name for the file so it will not be overwritten when you import the new files. For example, *commonltr.backup.css*.**
   h) **Click OK.**
   i) **Repeat the previous steps to rename the *commonrtl.css* and *messages.xml* files.**
5. **Insert system files:**
   a) **Right-click the *Repository* node and select Insert Documents.**

      The **Insert Documents Dialog** window appears.
   b) **Click Add Folder and browse to the *base-upgrade**system* folder in the upgrade package directory (for example, *C:\temp\DITA_CMS_4.3_upgrade_package\base-upgrade\system*).**
   c) **Select the `system` folder and click OK.**

      The system folder is listed in the **Insert Documents Dialog** window.
   d) **Select the Add and Replace options in the Set Options panel and leave Indexable unchecked.**

      For example:

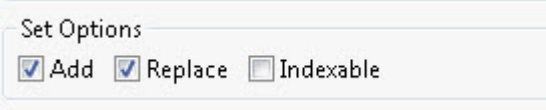
   e) **Click OK to import the system files.**
   f) **Right-click the *Repository* node and select Insert Documents.**

      The **Insert Documents Dialog** window appears.
   g) **Click Add Folder and browse to the *drm-upgrade**system* folder in the upgrade package directory (for example, *C:\temp\DITA_CMS_4.3_upgrade_package\drm-upgrade\system*).**
   h) **Select the `system` folder and click OK.**

The system folder is listed in the **Insert Documents Dialog** window.

i) **Select the Add and Replace options in the Set Options panel and leave Indexable unchecked.**

For example:



j) **Click OK to import the system files.**

6. **Add the `ValidateLayerTrigger` to the *triggers.xml* file:**
   a) **In the */system/conf* folder, right-click the *triggers.xml* file and select Check Out.**
   b) **Open the file.**
   c) **Add the following `<trigger>` element to the file, in the `<triggers>` element:**

```
<!-- trigger for DRM layer value check -->
<trigger class="com.ixiasoft.cms.triggers.drm.ValidateLayerTrigger" name="Check Library
 Layer Value" objtype="product" apply-to="Release" schedule="before"/>
```

   d) **Save and check in the file.**

7. **If you are using the linguistic review feature, add the classes of the elements you do not want translated in the updated *linguisticReview.xml* file:**
   a) **Expand the Content Store's `Repository` node and browse to the */system/conf* folder.**
   b) **Open the *linguisticReview.cms42.xml* file in an XML editor.**
   c) **Right-click the *linguisticReview.xml* file and select Check Out.**
   d) **Open the *linguisticReview.xml* file in an XML editor.**
   e) **Add the `name` attribute values from *linguisticReview.cms42.xml* to the *linguisticReview.xml* file.**

      Note: The element name was renamed from `<className>` to `<class>`, so be careful when copying and pasting the `name` attribute values between the two files.

   f) **Save and check in the *linguisticReview.xml* file.**

8. **Add new icons to the *equivalence.xml* file:**
   a) **In the */system/conf* folder, right-click the *equivalence.xml* file and select Check Out.**
   b) **Open the file.**
   c) **In the map type, look for the `<object type="subjectScheme"/>` element:**

   For example:

```
<equivalence type="map" standardSearch="true" groupName="Maps">
  <object type="map"/>
```

```
  <object type="bookmap" icon="/system/conf/icons/bookmap-icon.png"/>
  <object type="subjectScheme"/>

</equivalence>
```

d) **Add the icon attribute as follows:**

```
<equivalence type="map" standardSearch="true" groupName="Maps">
 <object type="map"/>
 <object type="bookmap" icon="/system/conf/icons/bookmap-icon.png"/>
 <object type="subjectScheme" icon="/system/conf/icons/subjscheme.png"/>
</equivalence>
```

e) **In the topic type, look for the `<object type="glossentry"/>` element:**

For example:

```
<equivalence type="topic" standardSearch="true" groupName="Topics">
   <object type="topic"/>
   ...
   <object type="glossentry"/>
   ...
</equivalence>
```

f) **Add the icon attribute as follows:**

```
<equivalence type="topic" standardSearch="true" groupName="Topics">
   <object type="topic"/>
   ...
   <object type="glossentry" icon="/system/conf/icons/glossentry.png"/>
   ...
</equivalence>
```

g) **Save and check in the file.**

9. **Integrate the Subject Scheme DTD into the *IxiaMap.dtd*:**

a) **In the */system/plugins/com.ixiasoft.dita.dtd/dtd* folder, right-click the *IxiaMap.dtd* file and select Check Out.**

b) **Open the file.**

c) **Look for the `MAP ENTITY DECLARATIONS` section.**

For example:

```
<!-- ============================================================= -->
<!--                    MAP ENTITY DECLARATIONS                    -->
<!-- ============================================================= -->

<!ENTITY % bookmap-dec
  PUBLIC "-//OASIS//ENTITIES DITA 1.2 BookMap//EN"
         "../../../dtd/bookmap/dtd/bookmap.ent"
>%bookmap-dec;
```

d) **At the end of this section, add the following code:**

```
<!ENTITY % subjectScheme-dec
  PUBLIC "-//OASIS//ENTITIES DITA 1.2 Subject Scheme Map//EN"
         "../../../dtd/subjectScheme/dtd/subjectScheme.ent"
>%subjectScheme-dec;
```

e) **Look for the `DOMAINS ATTRIBUTE OVERRIDE` section.**

For example (the actual code may vary):

```
<!-- ============================================================= -->
<!--                    DOMAINS ATTRIBUTE OVERRIDE                 -->
<!-- ============================================================= -->
<!--                    Must be declared ahead of the DTDs, which
                        puts @domains first in order               -->
<!ENTITY included-domains
                        "&bookmap-att;
                         &delay-d-att;
                         &mapgroup-d-att;
                         ...
```

f) **After the `&mapgroup-d-att;` code, add the following code:**

```
 &subjectScheme-att;
```

For example:

```
<!-- ============================================================= -->
<!--                    DOMAINS ATTRIBUTE OVERRIDE                 -->
<!-- ============================================================= -->
<!--                    Must be declared ahead of the DTDs, which
                        puts @domains first in order               -->
<!ENTITY included-domains
                        "&bookmap-att;
                         &delay-d-att;
                         &mapgroup-d-att;
                         &subjectScheme-att;
                        ...
```

g) **Look for the `MAP ELEMENT INTEGRATION` section.**

For example:

```
<!-- ============================================================= -->
<!--                    MAP ELEMENT INTEGRATION                    -->
<!-- ============================================================= -->

<!ENTITY % map-type
  PUBLIC "-//OASIS//ELEMENTS DITA 1.2 Map//EN"
         "../../../dtd/base/dtd/map.mod"
>%map-type;

<!ENTITY % bookmap-type
  PUBLIC "-//OASIS//ELEMENTS DITA 1.2 BookMap//EN"
         "../../../dtd/bookmap/dtd/bookmap.mod"
>%bookmap-type;
```

h) **At the end of this section, add the following code:**

```
<!ENTITY % subjectScheme-type
  PUBLIC "-//OASIS//ELEMENTS DITA 1.2 Subject Scheme Map//EN"
         "../../../dtd/subjectScheme/dtd/subjectScheme.mod"
>%subjectScheme-type;
```

i) **Save and check in the file.**

10. **Integrate the Subject Scheme DTD into the** *IxiaMap-releasemanagement.dtd***:**

a) **In the** */system/plugins/com.ixiasoft.dita.dtd/dtd* **folder, right-click the** *IxiaMap-releasemanagement.dtd* **file and select Check Out.**

b) **Open the file.**

c) **Look for the `MAP ENTITY DECLARATIONS` section.**

For example:

```
<!-- ============================================================= -->
<!--                    MAP ENTITY DECLARATIONS                    -->
<!-- ============================================================= -->

<!ENTITY % bookmap-dec
  PUBLIC "-//OASIS//ENTITIES DITA 1.2 BookMap//EN"
         "../../../dtd/bookmap/dtd/bookmap.ent"
>%bookmap-dec;
```

d) **At the end of this section, add the following code:**

```
<!ENTITY % subjectScheme-dec
  PUBLIC "-//OASIS//ENTITIES DITA Subject Scheme Map//EN"
         "../../../dtd/subjectScheme/dtd/subjectScheme.ent"
>%subjectScheme-dec;
```

e) **Look for the `DOMAINS ATTRIBUTE OVERRIDE` section.**

For example (the actual code may vary):

```
<!-- ============================================================= -->
<!--                    DOMAINS ATTRIBUTE OVERRIDE                 -->
<!-- ============================================================= -->
<!--                    Must be declared ahead of the DTDs, which
                        puts @domains first in order               -->
<!ENTITY included-domains
                        "&bookmap-att;
                         &delay-d-att;
                         &mapgroup-d-att;
                         ...
```

f) **After the `&mapgroup-d-att;` code, add the following code:**

```
 &subjectScheme-att;
```

For example:

```
<!-- ============================================================= -->
<!--                    DOMAINS ATTRIBUTE OVERRIDE                 -->
<!-- ============================================================= -->
<!--                    Must be declared ahead of the DTDs, which
                        puts @domains first in order               -->
<!ENTITY included-domains
                        "&bookmap-att;
                         &delay-d-att;
                         &mapgroup-d-att;
                         &subjectScheme-att;
                        ...
```

g) **Look for the `MAP ELEMENT INTEGRATION` section.**

For example:

```
<!-- ============================================================= -->
<!--                    MAP ELEMENT INTEGRATION                    -->
<!-- ============================================================= -->

<!ENTITY % map-type
  PUBLIC "-//OASIS//ELEMENTS DITA 1.2 Map//EN"
         "../../../dtd/base/dtd/map.mod"
>%map-type;

<!ENTITY % bookmap-type
  PUBLIC "-//OASIS//ELEMENTS DITA 1.2 BookMap//EN"
         "../../../dtd/bookmap/dtd/bookmap.mod"
>%bookmap-type;
```

h) **At the end of this section, add the following code:**

```
<!ENTITY % subjectScheme-type
  PUBLIC "-//OASIS//ELEMENTS DITA 1.3 Subject Scheme Map//EN"
         "../../../dtd/subjectScheme/dtd/subjectScheme.mod"
>%subjectScheme-type;
```

i) **Save and check in the file.**

11. **In your own shell DTDs, perform the previous steps on your map DTDs to integrate the SubjectScheme DTD.**

    This step is essential, otherwise you will not be able to use subject schemes in your specialized maps.

    Note: Contact IXIASOFT Support if you need help with this step.

12. **Remove the old system ID for the SubjectScheme DTD:**

    a) **Browse to the** */system/conf* **folder.**

    b) **Right-click the** *systemid.xml* **file and select Check Out.**

    c) **Open the** *systemid.xml* **file.**

d) **Look for the Subject Scheme reference:**

```
<reference public="-//OASIS//DTD DITA Subject Scheme Map//EN"
system="subjectScheme.dtd"/>
```

e) **Remove this line from the file.**

f) **Save and check in the** *systemid.xml* **file.**

13. **If you had not customized the** *eclipseui.xml* **file for your deployment, use the new version provided in DITA CMS 4.3:**

    The default file was updated so that the initial UI is simplified. If you already customized your *eclipseui.xml* file, you can skip this step.

    a) **Expand the Content Store's `Repository` node and browse to the** */system/conf* **folder.**

    b) **Right-click the** *eclipseui.xml* **file and select Rename.**

    c) **In the New Document Name box, enter `eclipseui.cms42.xml`.**

    d) **Click OK.**

    e) **Right-click the** *eclipseui.new_cms43_version.xml* **file and select Rename.**

    f) **In the New Document Name box, enter `eclipseui.xml`.**

    g) **Click OK.**

14. **Replace the `type name="*"` elements with specific object types in the** *accessrights.xml* **file for all front-end methods:**

    These changes are required to fix errors that occur when an object does not have one of the cycles specified and users select the method in the interface. For example, consider the following code for the "Assign to" method:

```
<method name="Assign to" 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>
    <!-- Object on which action is taken -->
    <current>
     <type name="*">
      <statuses>
       <status>Localization:*</status>
      </statuses>
     </type>
    </current>
```

    This code indicates that the Assign to method can be applied to all the object types in the Localization cycle. However, the Localization cycle does not exist for objects of type `build-manifest` or `snapshot`, so the code above will create errors when users try to run the "Assign to" command in the user interface.

Note: This step applies only for front-end methods, that is, methods with the **type="front-end"** attribute; for example:

```
<method name="Assign to" type="front-end" hide-when-disabled="true">
  <!-- determines if a notification email should be sent when this action is taken -->
  <notify enabled="false"/>
```

To prevent these issues:

a) **Expand the Content Store's `Repository` node and browse to** */system/conf* **folder.**

b) **Right-click the accessrights.xml file and select Check Out.**

c) **Open the accessrights.xml file in an XML editor.**

d) **Locate the first `<type name="*">` element.**

For example:

```
<method name="Assign to" 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>
    <!-- Object on which action is taken -->
    <current>
     <type name="*">
      <statuses>
       <status>Localization:*</status>
      </statuses>
     </type>
    </current>
```

e) **If the method is of `type="front-end"` (as shown above in bold), duplicate the `<type name="*">` element for each object type to which this action applies and replace the `"*"` value with the object type.**

For example, if the action applies to maps, topics, images, and resources, replace the `<condition>` code above with the following:

```
<condition>
   <!-- Object on which action is taken -->
   <current>
      <type name="map">
         <statuses>
           <status>Localization:*</status>
         </statuses>
      </type>
      <type name="topic">
         <statuses>
            <status>Localization:*</status>
         </statuses>
      </type>
      <type name="image">
         <statuses>
```

```
            <status>Localization:*</status>
         </statuses>
      </type>
      <type name="resource">
         <statuses>
           <status>Localization:*</status>
         </statuses>
      </type>
   </current>
   <!-- Action user must be in this list -->
   <users>
   ...
</condition>
```

f) **Repeat for all `<type name="*">` elements of front-end methods in the file.**

g) **Save the** *accessrights.xml* **file.**

15. **Update the access rights for the Edit Variables method to ensure it is available to the appropriate users at the appropriate status only:**

a) **In the** *accessrights.xml* **file, look for the Edit Variables method.**

It should look as follows (the actual code may differ according to your deployment):

```
<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>
```

b) **Change the `<status>` element to `Authoring:work` (or the equivalent for your deployment).**

For example:

```
<type name="map">
   <statuses>
      <status>Authoring:work</status>
```

```
   </statuses>
</type>
```

c) **Set the `<users>` element to the groups and roles that can run this command.**

For example:

```
<users>
   <roles>
      <role name="Information Architect"/>
      <role name="Writer"/>
   </roles>
   <groups>
      <group name="System Administrators"/>
   </groups>
</users>
```

d) **Save the *accessrights.xml* file.**

16. **Update the access rights for the Insert Variables method to ensure it is available to the appropriate users at the appropriate status only:**

a) **In the *accessrights.xml* file, look for the Insert Variables method.**

It should look as follows (the actual code may differ according to your deployment):

```
<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>
```

b) **Change the `<status>` element to `Authoring:work` (or the equivalent for your deployment).**

For example:

```
<type name="map">
   <statuses>
```

```
      <status>Authoring:work</status>
   </statuses>
</type>
```

c) **Set the `<users>` element to the groups and roles that can run this command.**

For example:

```
<users>
   <roles>
      <role name="Information Architect"/>
      <role name="Writer"/>
   </roles>
   <groups>
      <group name="System Administrators"/>
   </groups>
</users>
```

d) **Save and check in the *accessrights.xml* file.**

17. Add the Filename to the *display.xml* file:

This will allow you to see the filename of objects in the DITA CMS views.

a) **In the */system/conf* folder, right-click the *display.xml* file and select Check Out.**

b) **Open the file.**

c) **Add the following `<key>` element to the file, in a `<columns>` element:**

Note: Add the key in alphabetical order of label.

```
<key halign="LEFT" label="Filename" name="Name" sortOrder="ASC" sortType="ALPHA"
type="Property" visibility="255" width="110"/>
```

d) **Save and check in the file.**

18. Add new access rights to the *accessrights.xml* file for the DRM commands:

a) **In the */system/conf* folder, right-click the *accessrights.xml* file and select Check Out.**

b) **Open the file.**

c) **Locate the following line:**

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

d) **Just before this line, add the following `<method>` code:**

Note: For ease of use, the following method code is also available in a text file, in the following location:

%UpgradePackageDir%/drm-upgrade/accessrights_newDRMmethods.xml

For example:

```
C:\temp\Upgrade Packages\DITA_CMS_4.3_upgrade_package\
drm-upgrade\accessrights_newDRMmethods.xml
```

```
  <!-- Possible type front-end / api -->
  <method name="AddLibrary" type="front-end" multiselect-disabled="false"
hide-when-disabled="true">
  <notify enabled="false"/>
  <conditionset operator="any">
   <condition>
    <current>
     <type name="version">
      <statuses>
       <status>Authoring:development</status>
      </statuses>
     </type>
    </current>
    <users>
     <roles>
      <role name="Information Architect"/>
     </roles>
     <groups>
      <group name="System Administrators"/>
     </groups>
    </users>
   </condition>
  </conditionset>
 </method>
 <!-- Possible type front-end / api -->
 <method name="ChangePrimary" type="front-end" multiselect-disabled="true"
hide-when-disabled="true">
  <notify enabled="false"/>
  <conditionset operator="any">
   <condition>
    <current>
     <type name="topic">
      <statuses>
       <status>Authoring:*</status>
      </statuses>
     </type>
     <type name="image">
      <statuses>
       <status>Authoring:*</status>
      </statuses>
     </type>
     <type name="resource">
      <statuses>
       <status>Authoring:*</status>
      </statuses>
     </type>
    </current>
    <users>
     <roles>
      <role name="Information Architect"/>
     </roles>
     <groups>
      <group name="System Administrators"/>
     </groups>
    </users>
   </condition>
  </conditionset>
 </method>
 <!-- Possible type front-end / api -->
 <method name="CreateTag" type="front-end" multiselect-disabled="false"
```

```
hide-when-disabled="true">
  <notify enabled="false"/>
  <conditionset operator="any">
   <condition>
    <current>
     <type name="version">
      <statuses>
       <status>Authoring:*</status>
      </statuses>
     </type>
    </current>
    <users>
     <roles>
      <role name="Information Architect"/>
     </roles>
     <groups>
      <group name="System Administrators"/>
     </groups>
    </users>
   </condition>
  </conditionset>
 </method>
 <!-- Possible type front-end / api -->
 <method name="DeleteDRMObject" type="front-end" multiselect-disabled="false"
hide-when-disabled="true">
  <notify enabled="false"/>
  <conditionset operator="any">
   <condition>
    <current>
     <type name="product">
      <statuses>
       <status>Authoring:open</status>
      </statuses>
     </type>
     <type name="release">
      <statuses>
       <status>Authoring:open</status>
      </statuses>
     </type>
     <type name="version">
      <statuses>
       <status>Authoring:open</status>
      </statuses>
     </type>
    </current>
    <users>
     <roles>
      <role name="Information Architect"/>
     </roles>
     <groups>
      <group name="System Administrators"/>
     </groups>
    </users>
   </condition>
  </conditionset>
 </method>
 <!-- Possible type front-end / api -->
 <method name="Refactor" type="front-end" multiselect-disabled="true"
hide-when-disabled="true">
  <notify enabled="false"/>
  <conditionset operator="any">
   <condition>
    <current>
     <type name="map">
      <statuses>
       <status>Authoring:*</status>
```

```
      </statuses>
     </type>
     <type name="topic">
      <statuses>
       <status>Authoring:*</status>
      </statuses>
     </type>
     <type name="image">
      <statuses>
       <status>Authoring:*</status>
      </statuses>
     </type>
     <type name="resource">
      <statuses>
       <status>Authoring:*</status>
      </statuses>
     </type>
    </current>
    <users>
     <roles>
      <role name="Writer"/>
      <role name="Information Architect"/>
     </roles>
     <groups>
      <group name="System Administrators"/>
     </groups>
    </users>
   </condition>
  </conditionset>
</method>
<!-- Possible type front-end / api -->
<method name="RemoveLibrary" type="front-end">
 <!-- 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="version">
     <statuses>
      <status>Authoring:development</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
   <users>
    <roles>
     <role name="Information Architect"/>
    </roles>
    <groups>
     <group name="System Administrators"/>
    </groups>
   </users>
  </condition>
 </conditionset>
</method>
<!-- Possible type front-end / api -->
<method name="RenameVersion" type="front-end">
 <!-- 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="version">
     <statuses>
      <status>Authoring:development</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
   <users>
    <roles>
     <role name="Information Architect"/>
    </roles>
    <groups>
     <group name="System Administrators"/>
    </groups>
   </users>
  </condition>
 </conditionset>
</method>
<!-- Possible type front-end / api -->
<method name="UpdateLibrary" type="front-end">
 <!-- 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="version">
     <statuses>
      <status>Authoring:development</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
   <users>
    <roles>
     <role name="Information Architect"/>
    </roles>
    <groups>
     <group name="System Administrators"/>
    </groups>
   </users>
  </condition>
 </conditionset>
</method>
<method name="ChangeLayer" type="front-end">
 <!-- 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="version">
     <statuses>
      <status>Authoring:development</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
   <users>
    <roles>
```

```
      <role name="Information Architect"/>
     </roles>
     <groups>
      <group name="System Administrators"/>
     </groups>
    </users>
   </condition>
  </conditionset>
 </method>
```

e) **In the `<availablemethods>` element, add the following actions:**

```
<action name="DeleteDitaval" />
<action name="ManageDitaval" />
<action name="AddLibrary" />
<action name="ChangePrimary" />
<action name="CreateTag" />
<action name="DeleteDRMObject" />
<action name="Refactor" />
<action name="RemoveLibrary" />
<action name="RenameVersion" />
<action name="UpdateLibrary" />
<action name="ChangeLayer" />
```

f) **Look for the following method name:**

```
CloneDRMVersion
```

g) **Change it to the following:**

```
CloneVersion
```

There should be two occurrences of this name.

h) **Look for the following method name:**

```
SynchronizeDRMOVersion
```

i) **Change it to the following:**

```
SynchronizeVersion
```

There should be two occurrences of this name.

j) **Save and check in the file.**

19. **Add a new search index to the** *advancedSearchIndexes* **file to search with the filename in the Search view:**

a) **In the** */system/conf* **folder, right-click the** *advancedSearchIndexes.xml* **file and select Check Out.**

b) **Open the file.**

c) **Add the following index to the file:**

```
<index display="Filename" listable="false" name="Name"
type="property" views="ALL"/>
```

d) **Save, close, and check in the file.**

20. **Add the Assignments Comments column to the *display.xml* file:**
    a) **In the */system/conf* folder, right-click the *display.xml* file and select Check Out.**
    b) **Open the file.**
    c) **Add the following `<key>` element to the file, in a `<columns>` element:**

       Note: Add the key in alphabetical order of label.

```
<key halign="LEFT" label="Assignment Comments" name="assignment_comment" sortOrder="ASC"
 sortType="ALPHA" type="Index" visibility="255" width="30"/>
```

    d) **Save and check in the file.**

21. **Edit the *resources.xml* file to add a type for non-DITA files and remove an unknown type, if present:**
    a) **Right-click the */system/conf/resources.xml* file and click Check Out.**
    b) **Double-click the file to open it in the XML editor area.**
    c) **Add the following lines to the file:**

```
<!-- Default type used by the import process -->
<type description="">unknown</type>
```

       For example, the file should look as follows:

```
<resource>
 <resourcetypes>
  <type description="">pdf-graphics</type>
  <type description="">pdf-legal</type>
  <type description="">css</type>
  <type description="">html</type>
  <!-- Default type used by the import process -->
  <type description="">unknown</type>
 </resourcetypes>
</resource>
```

    d) **If the `<type description="">ditaval-file</type>` line is present, remove it from the file.**
    e) **Save and check in the file.**

22. **Edit the *users.dtd* file to prevent text from being added in the `<role>` and `<group>` elements.**
    a) **Browse to */system/conf/* to locate the *users.dtd* file.**
    b) **Right-click the file and click Check Out.**
    c) **Double-click the file to open it in the XML editor area.**

d) **Locate the element `<!ELEMENT role (#PCDATA)` and replace `(#PCDATA)` with `EMPTY`.**

For example: `<!ELEMENT role EMPTY`

e) **Locate the element `<!ELEMENT group (#PCDATA)` and replace `(#PCDATA)` with `EMPTY`.**

For example: `<!ELEMENT group EMPTY`

f) **Save, close, and check in the file.**

23. **If you had customized your Preview view, use your backup copy to re-implement your changes to the Preview view.**

Update the Index Definition document

This procedure describes how to modify the Index Definition document to upgrade your Content Store to DITA CMS 4.3.

These changes are required to implement new features.

> **Note:** IXIASOFT recommends that you change the Index Definition document during off hours only. This operation will require a reindexing of the modified indexes, which will reduce the performance of the TEXTML Server.

When adding new indexes to the index definition, IXIASOFT recommends that you add them in alphabetical order.

To update the Index Definition document:

1. **Open a TEXTML Server Console Java.**
2. **Connect to the server and Content Store to upgrade.**
3. **Expand the Content Store node to display the Index Definition.**
4. **Right-click Index Definition and select Lock.**
5. **Open the Index Definition document in an XML editor.**
6. **Under `<indexes>` add the following `<index>` element:**

```
<index CUSTOMPROPERTY="True" NAME="assignment_comment" SYNC="False">
 <!-- System index required by the DITA CMS -->
 <stringindex KEEPEXTRACTEDVALUES="True">
  <elements>
   <element DEPTH="INFINITE" XPATH="string-join(//assignment/comment/text(),', ')"/>
  </elements>
 </stringindex>
</index>
```

7. **Under `<indexes>` add the following `<index>` element:**

```
<index CUSTOMPROPERTY="True" NAME="fragmentreferences" SYNC="True">
 <!-- System index required by the DITA CMS -->
 <stringindex KEEPEXTRACTEDVALUES="True">
  <elements>
   <element DEPTH="0"
XPATH="//reference[@contentDependency='true'][not(@keyref)][not(@conkeyref)]/concat('srcRef=',
 @srcRef, ', text=', text())"/>
  </elements>
 </stringindex>
</index>
```

8. **Under `<indexes>` add the following `<index>` element:**

```
<index NAME="layer" SYNC="True">
   <stringindex KEEPEXTRACTEDVALUES="True">
      <elements>
        <element DEPTH="INFINITE" XPATH="//*[contains(@class, ' product/layer
')]/@content"/>
      </elements>
   </stringindex>
</index>
```

9. **Look for the following index:**

```
<index CUSTOMPROPERTY="True" NAME="branchTag" SYNC="True">
```

10. **Set the `SYNC` attribute to `False`.**

    For example:

```
<index CUSTOMPROPERTY="True" NAME="branchTag" SYNC="False">
```

11. **Look for the following index:**

```
<index CUSTOMPROPERTY="True" NAME="branchedFrom" SYNC="True">
```

12. **Set the `SYNC` attribute to `False`.**

    For example:

```
<index CUSTOMPROPERTY="True" NAME="branchedFrom" SYNC="False">
```

13. **Look for the following index:**

```
<index CUSTOMPROPERTY="True" NAME="branchedTo" SYNC="True">
```

14. **Set the `SYNC` attribute to `False`.**

    For example:

```
<index CUSTOMPROPERTY="True" NAME="branchedTo" SYNC="False">
```

15. Under <summaries>\<summary NAME="fullsummary">, add the following <field> elements:

Note: IXIASOFT recommends that you add them in alphabetical order.

```
<field NAME="CreationDate" TYPE="Property" VALUE="All"/>
<field NAME="CreationTime" TYPE="Property" VALUE="All"/>
 <field NAME="Name" TYPE="Property" VALUE="All"/>
```

16. Look for the following index:

```
<index CUSTOMPROPERTY="True" NAME="keyref-allreferences" SYNC="True">
   ...
</index>
```

17. Change it as follows:

```
<index CUSTOMPROPERTY="True" NAME="keyref-allreferences" SYNC="True">
 <!-- System index required by the DITA CMS -->
 <stringindex KEEPEXTRACTEDVALUES="True">
  <elements>
   <element XPATH="/customproperties[lastmodtime]/reference[@keyref or
@conkeyref][@scope='local' or not(@scope)]/@srcRef" DEPTH="INFINITE"/>
   <!-- Patched for 4.0 and 4.1 legacy content -->
   <element XPATH="/customproperties[not(lastmodtime)]/reference[@keyref or
@conkeyref][not(contains(@srcRef, '/'))][@scope='local' or not(@scope)]/@srcRef"
DEPTH="INFINITE"/>
   <!-- Patch for 4.0 and 4.1 legacy content -->
   <element XPATH="for $e in /customproperties[not(lastmodtime)]/reference[@keyref or
@conkeyref][contains(@srcRef, '/')][@scope='local' or not(@scope)] return
substring-before($e/@srcRef, '/')" DEPTH="INFINITE"/>
  </elements>
 </stringindex>
</index>
```

18. Look for the following index:

```
<index CUSTOMPROPERTY="True" NAME="keyref-contentDependencies" SYNC="True">
   ...
</index>
```

19. Change it as follows:

```
<index CUSTOMPROPERTY="True" NAME="keyref-contentDependencies" SYNC="True">
 <!-- System index required by the DITA CMS -->
 <stringindex KEEPEXTRACTEDVALUES="True">
  <elements>
   <element DEPTH="0"
XPATH="/customproperties[lastmodtime]/reference[@contentDependency='true'][@keyref or
@conkeyref][@scope='local' or not(@scope)]/@srcRef"/>
   <!-- Patch for 4.0 and 4.1 legacy content -->
   <element DEPTH="0"
XPATH="/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][contains(@class,
 ' map/topicref ')][@keyref][@scope='local' or not(@scope)]/@srcRef"/>
   <element DEPTH="0"
XPATH="/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][contains(@class,
 ' topic/image ')][@keyref][@scope='local' or not(@scope)]/@srcRef"/>
   <element DEPTH="0" XPATH="for $e in
/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][@conkeyref][@scope='local'
```

```
 or not(@scope)] return substring-before($e/@srcRef, '/')"/>
  </elements>
 </stringindex>
</index>
```

20. Look for the following index:

```
<index CUSTOMPROPERTY="True" NAME="keyref-conrefDependencies" SYNC="True">
   ...
</index>
```

21. Change it as follows:

```
<index CUSTOMPROPERTY="True" NAME="keyref-conrefDependencies" SYNC="True">
   <!-- System index required by the DITA CMS -->
   <stringindex KEEPEXTRACTEDVALUES="True">
      <elements>
         <element DEPTH="0"
XPATH="/customproperties[lastmodtime]/reference[@conkeyref]/@srcRef"/>
         <!-- Patched for 4.1 and 4.0 legacy content -->
         <element DEPTH="0"
XPATH="/customproperties[not(lastmodtime)]/reference[@contentDependency='true'][@conkeyref]/@srcRef"/>

         <!-- Patch for 4.1 and 4.0 legacy content -->
         <element DEPTH="0" XPATH="for $e in
/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][@conkeyref]
return substring-before($e/@srcRef, '/')"/>
      </elements>
   </stringindex>
</index>
```

22. Look for the following index:

```
<index CUSTOMPROPERTY="True" NAME="keyref-references" SYNC="True">
   ...
</index>
```

23. Change it as follows:

```
<index CUSTOMPROPERTY="True" NAME="keyref-references" SYNC="True">
   <!-- System index required by the DITA CMS -->
   <stringindex KEEPEXTRACTEDVALUES="True">
      <elements>
         <element
XPATH="/customproperties[lastmodtime]/reference[@contentDependency='false'][@keyref or
@conkeyref]/@srcRef" DEPTH="INFINITE"/>
         <!-- Patched for 4.1 and 4.0 legacy content -->
         <element
XPATH="/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][not(contains(@class,
 ' map/topicref '))][not(contains(@class, ' topic/image '))][@keyref][not(contains(@srcRef,
 '/'))]/@srcRef" DEPTH="INFINITE"/>
         <!-- Patch for 4.1 and 4.0 legacy content -->
         <element XPATH="for $e in
/customproperties[not(lastmodtime)]/reference[@contentDependency='false'][not(contains(@class,
 ' map/topicref '))][not(contains(@class, ' topic/image '))][@keyref][contains(@srcRef,
'/')] return substring-before($e/@srcRef, '/')" DEPTH="INFINITE"/>
      </elements>
   </stringindex>
</index>
```

24. Locate the following <index> element:

```
<index NAME="type" SYNC="True">
<!-- Index referenced by the Advanced Search on attributes -->
 <stringindex KEEPEXTRACTEDVALUES="False">
  <elements>
   <element DEPTH="INFINITE" XPATH="//@type"/>
  </elements>
 </stringindex>
</index>
```

25. In the index, delete `SYNC="True"`.

Example:

```
<index NAME="type">
<!-- Index referenced by the Advanced Search on attributes -->
 <stringindex KEEPEXTRACTEDVALUES="False">
  <elements>
   <element DEPTH="INFINITE" XPATH="//@type"/>
  </elements>
 </stringindex>
</index>
```

26. Locate the following <index> element:

```
<index NAME="wordCount" SYNC="True">
 <numericindex KEEPEXTRACTEDVALUES="True">
  <integerindexproperties/>
  <elements>
   <element DEPTH="0" XPATH="string-length(normalize-space(string(/))) -
string-length(replace(normalize-space(string(/)), ' ', ''))" />
  </elements>
 </numericindex>
</index>
```

27. In the index, delete `SYNC="True"`.

```
<index NAME="wordCount">
 <numericindex KEEPEXTRACTEDVALUES="True">
  <integerindexproperties/>
  <elements>
   <element DEPTH="0" XPATH="string-length(normalize-space(string(/))) -
string-length(replace(normalize-space(string(/)), ' ', ''))" />
  </elements>
 </numericindex>
</index>
```

28. Right-click the Index Definition document and select Check In.

Update the **accessrights.xml** file

This section provides configuration updates that are required in the the **accessrights.xml** file.

In the access rights configuration, the methods of type `outputtype` (for example, Dita2Pdf, Dita2htmlhelp, Dita2EclipseHelp, Dita2xhtml, etc.) should be updated to replace the `<type`

`name="*">` element with the specific object types to which they can be applied. This ensures that Build Manifests can be generated using the BuildManifest output type only (and not the Dita2Pdf output type, for example).

To fix this issue:

1. **Expand the Content Store's `Repository` node and browse to *system/conf* folder.**
2. **Right-click the accessrights.xml file and select Check Out.**
3. **Open the accessrights.xml file in an XML editor.**
4. **Look for methods with the following type: `type="outputtype".`**

   For example:

```
<method name="Dita2Pdf" type="outputtype">
 <notify enabled="false"/>
 <conditionset operator="any">
  <condition>
   <current>
    <type name="*">
     <statuses>
      <status>Authoring:*</status>
      <status>Localization:*</status>
      <status>Published:*</status>
     </statuses>
    </type>
    ...
```

5. **Replace the `<type name="*">` … `</type>` element with the following:**

```
<type name="snapshot">
 <statuses>
  <status>Authoring:*</status>
  <status>Localization:*</status>
 </statuses>
</type>
<type name="map">
 <statuses>
  <status>Authoring:*</status>
  <status>Localization:*</status>
  <status>Published:*</status>
 </statuses>
</type>
<type name="topic">
 <statuses>
  <status>Authoring:*</status>
  <status>Localization:*</status>
  <status>Published:*</status>
 </statuses>
</type>
<type name="image">
 <statuses>
  <status>Authoring:*</status>
  <status>Localization:*</status>
  <status>Published:*</status>
 </statuses>
</type>
```

   This ensures that the output type is applied to snapshots, maps, topics, and images only.

6. **Repeat for all methods with the `type="outputtype"` type.**
7. **Save and check in the *accessrights.xml* file.**
8. **Inform users of the changes.**

   The changes will be applied automatically once users close and then reopen their DITA CMS. Users can also apply the changes without restarting their DITA CMS by clicking ***DITA CMS*** > ***Synchronize Configuration***.

Update the equivalence.xml file and delete the bookmap-stub-generate.ditamap template

This section provides configuration updates that are recommended but not required.

Delete the bmanifest type from the equivalence.xml file

The *equivalence.xml* file contains two types for build manifests:

- `build-manifest`
- `bmanifest`

This is shown below:

```
<equivalence type="build-manifest" standardSearch="true" groupName="Others">
 <object type="build-manifest" icon="/system/conf/icons/manifest.gif"/>
 <object type="bmanifest" icon="/system/conf/icons/manifest.gif"/>
</equivalence>
```

The bmanifest type is not used and can be removed:

1. Browse to the */system/conf* folder.
2. Right-click *equivalence.xml* and select **Check out**.
3. Delete the line for the `bmanifest` object type.

   Note: This type may not be there. In this case, ignore this procedure.
4. Save and check in the file.

Delete the bookmap-stub-generate.ditamap template

The *bookmap-stub-generate.ditamap* template was created to test the DITA CMS 4.1 feature for generating stubs from a map template. You can safely remove this template:

1. Browse to the */system/templates/maps* folder.
2. Right-click *bookmap-stub-generate.ditamap* and select **Delete**.
3. Click **Yes** to confirm.

Upgrade Core 4.3.61 to latest 4.4 release

This section contains the instructions to apply all the configuration changes required to upgrade your Core 4.3.61 Content Store to the latest build in the 4.4 release.

Download the DITA CMS Core 4.4 upgrade package

The DITA CMS core 4.4 upgrade package contains files required to upgrade your Content Store to the latest 4.4 version.

To download the DITA CMS upgrade package:

1. **Go to the following URL:**

   http://cms.ixiasoft.com/downloads/4.3_to_4.4_upgrade/

2. **Copy the** *DITA_CMS_4.4_upgrade_package-<date>.zip* **file to your computer.**
3. **Unzip the upgrade package to a working directory.**

   For example:

   ```
   C:\temp\
   ```

   Note: Throughout this document, the `%UpgradePackageDir%` expression is used to represent the upgrade package directory.

The upgrade package contains the following directories:

- `base-upgrade`: Contains the files required to upgrade to DITA CMS
- `drm-upgrade`: Contains the files required to upgrade the Dynamic Release Management module to the latest release

Import the Core 4.4 configuration files

This procedure describes how to import the upgrade's configuration files for DITA CMS Core 4.4.

To import the new files into your Content Store:

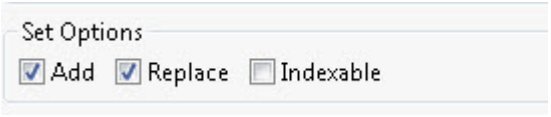
1. **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**
   a) **Select *Window* > *Open Perspective* > *Other***
   b) **Click TEXTML Administration.**
   c) **Click OK.**

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 the** *Repository* **node and select Insert Documents.**

   The **Insert Documents Dialog** window appears.
6. **Click Add Folder and browse to the** *base-upgrade\system* **folder in the upgrade package directory (for example,** *C:\temp\DITA_CMS_[version]_upgrade_package\base-upgrade\system***).**
7. **Select the `system` folder and click OK.**

   The system folder is listed in the **Insert Documents Dialog** window.
8. **Select the Add and Replace options in the Set Options panel and leave Indexable unchecked.**

   For example:

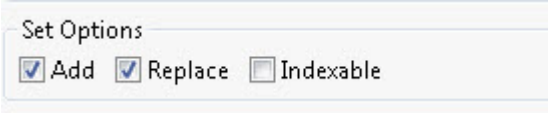
   

9. **Click OK to import the system files.**
10. **Right-click the** *Repository* **node and select Insert Documents.**

    The **Insert Documents Dialog** window appears.
11. **Click Add Folder and browse to the** *drm-upgrade\system* **folder in the upgrade package directory (for example,** *C:\temp\DITA_CMS_[version]_upgrade_package\drm-upgrade\system***).**
12. **Select the `system` folder and click OK.**

    The system folder is listed in the **Insert Documents Dialog** window.
13. **Select the Add and Replace options in the Set Options panel and leave Indexable unchecked.**

    For example:

14. **Click OK to import the system files.**

Update the Index Definition document

This procedure describes the necessary changes that must be made in the Index Definition document.

These changes are required to implement new features.

> **Note:** IXIASOFT recommends that you change the Index Definition document during off hours only. This operation will require a reindexing of the modified indexes, which will reduce the performance of the TEXTML Server.

When adding new indexes to the index definition, IXIASOFT recommends that you add them in alphabetical order.

To update the Index Definition document:

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 node to display the Index Definition.**
6. **Right-click Index Definition and select Lock.**
7. **Open the Index Definition document in an XML editor.**
8. **Under `<indexes>` add the following `<index>` element:**

```
<index CUSTOMPROPERTY="True" NAME="active_collaborativereviewer" SYNC="True">
 <!-- System index required by the Web Platform -->
 <stringindex KEEPEXTRACTEDVALUES="True">
  <elements>
   <!-- TODO: This index needs to be set with the proper role and statuses -->
   <element DEPTH="INFINITE"
XPATH="//assignment[@role='Reviewer'][//status='Authoring:open']/assignedTo"/>
  </elements>
 </stringindex>
</index>
```

9. **Under `<indexes>` add the following `<index>` element:**

```
<index CUSTOMPROPERTY="True" NAME="completedReviewAssignmentCount" SYNC="False">
 <!-- System index required by the Web Platform -->
```

```
 <numericindex KEEPEXTRACTEDVALUES="True">
  <integerindexproperties/>
  <elements>
   <!-- TODO: This index needs to be set with the proper role and statuses -->
   <element DEPTH="INFINITE" XPATH="count(//assignment[@role='Reviewer' or
@role='Translation Reviewer']/assignedTo[@state='completed'])"/>
  </elements>
 </numericindex>
</index>
```

10. Under <indexes> add the following <index> element:

```
<index CUSTOMPROPERTY="True" NAME="totalReviewAssignmentCount" SYNC="False">
 <!-- System index required by the Web Platform -->
 <numericindex KEEPEXTRACTEDVALUES="True">
  <integerindexproperties/>
  <elements>
   <!-- TODO: This index needs to be set with the proper role and statuses -->
   <element DEPTH="INFINITE" XPATH="count(//assignment[@role='Reviewer' or
@role='Translation Reviewer']/assignedTo)"/>
  </elements>
 </numericindex>
</index>
```

11. Look for the following index:

```
<index NAME="title" SYNC="True">
 <!-- System index required by the DITA CMS -->
 <stringindex KEEPEXTRACTEDVALUES="True">
  <elements>
   <element DEPTH="INFINITE" XPATH="/*/@title"/>
   <element DEPTH="0" XPATH="normalize-space(string-join(for $e in (/*/*[contains(@class,'
 topic/title ')]//text()) return string(normalize-space($e)), ' '))"/>
   <!-- The following XPath resolves @conkeyref attributes into the target element id, pre
 and postfixed with "%" -->
   <!--<element DEPTH="0" XPATH="normalize-space(string-join(for $e in
(/*/*[contains(@class,' topic/title ')]//node()) return (if ($e/@conkeyref) then concat('%',
 substring-after($e/@conkeyref,'/'), '%') else if ($e/self::text()) then
string(normalize-space($e)) else ''), ' '))"/>-->
  </elements>
 </stringindex>
</index>
```

12. Under the <elements>, add the following lines:

```
<!-- BEGIN DEV-2960 20170718 Support DITAVAL REF (1.3) -->
<element DEPTH="0"
XPATH="string(/processing-instruction()[local-name()='ixiasoft.ditaval.title'])"/>
<!-- END DEV-2960 20170718 -->
```

The result should look like this example:

```
<index NAME="title" SYNC="True">
 <!-- System index required by the DITA CMS -->
 <stringindex KEEPEXTRACTEDVALUES="True">
  <elements>
   <!-- BEGIN DEV-2960 20170718 Support DITAVAL REF (1.3) -->
   <element DEPTH="0"
XPATH="string(/processing-instruction()[local-name()='ixiasoft.ditaval.title'])"/>
   <!-- END DEV-2960 20170718 -->
```

```
   <element DEPTH="INFINITE" XPATH="/*/@title"/>
   <element DEPTH="0" XPATH="normalize-space(string-join(for $e in (/*/*[contains(@class,'
 topic/title ')]//text()) return string(normalize-space($e)), ' '))"/>
   <!-- The following XPath resolves @conkeyref attributes into the target element id, pre
 and postfixed with "%" -->
   <!--<element DEPTH="0" XPATH="normalize-space(string-join(for $e in
(/*/*[contains(@class,' topic/title ')]//node()) return (if ($e/@conkeyref) then concat('%',
 substring-after($e/@conkeyref,'/'), '%') else if ($e/self::text()) then
string(normalize-space($e)) else ''), ' '))"/>-->
  </elements>
 </stringindex>
</index>
```

13. **Look for the element `<summaries>`. Under `<summary NAME="fullsummary">`, add the following `<field>` elements to the `<fieldlist>`:**

    Note: IXIASOFT recommends that you add them in alphabetical order.

```
<field NAME="completedReviewAssignmentCount" TYPE="Index" VALUE="All"/>
<field NAME="totalReviewAssignmentCount" TYPE="Index" VALUE="All"/>
```

14. **Right-click the Index Definition document and select Check In.**

Update the system configuration

Each of the topics in this section describe manual edits that must be made to specific configuration files. All of the following changes to the configuration files are mandatory.

Update the accessrights.xml configuration file

Several mandatory changes are required in the *accessrights.xml* file to support new features.

To update the file:

1. **Expand the Content Store's Repository node and browse to** */system/conf/* **to locate the** *accessrights.xml* **file.**
2. **Right-click the file and click Check Out.**
3. **Double-click the file to open it in the XML editor area.**
4. **Locate the `Assign to` method:**

```
<method name="Assign to" type="front-end" hide-when-disabled="true">
```

5. **In the `Assign to` method, locate the condition which contains the `map` type.**

   For example:

```
<condition>
 <!-- Object on which action is taken -->
 <current>
  <type name="map">
   <statuses>
```

```
    <status>Authoring:*</status>
   </statuses>
  </type>
 </current>
 <!-- Action user must be in this list -->
 <users>
  <roles>
   <role name="Writer"/>
   <role name="Information Architect"/>
   <role name="Project Coordinator"/>
  </roles>
  <groups>
   <group name="System Administrators"/>
  </groups>
 </users>
</condition>
```

6. **Add the following lines to include the new `collaborative-review` type to the condition.**

```
<type name="collaborative-review">
 <statuses>
  <status>Authoring:*</status>
 </statuses>
</type>
```

The result should look like the following example:

```
<condition>
 <!-- Object on which action is taken -->
 <current>
  <type name="map">
   <statuses>
    <status>Authoring:*</status>
   </statuses>
  </type>
  <type name="collaborative-review">
   <statuses>
    <status>Authoring:*</status>
   </statuses>
  </type>
 </current>
 <!-- Action user must be in this list -->
 <users>
  <roles>
   <role name="Writer"/>
   <role name="Information Architect"/>
   <role name="Project Coordinator"/>
  </roles>
  <groups>
   <group name="System Administrators"/>
  </groups>
 </users>
</condition>
```

7. **Locate the `Show Preview` method:**

```
<method name="Show Preview" type="front-end" hide-when-disabled="true">
```

8. **In the condition, locate the `snapshot` type and delete the following line:**

```
<status>Published:*</status>
```

The result should look like the following example:

```
<type name="snapshot">
 <statuses>
  <status>Authoring:*</status>
  <status>Localization:*</status>
 </statuses>
</type>
```

9. **In the condition, add the following lines to include the new `collaborative-review` type.**

```
<type name="collaborative-review">
 <statuses>
  <status>Authoring:*</status>
 </statuses>
</type>
```

The result should look like the following example:

```
<method name="Show Preview" type="front-end" hide-when-disabled="true">
 <notify enabled="false"/>
 <conditionset operator="any">
  <condition>
   <current>
    <type name="image">
     <statuses>
      <status>Authoring:*</status>
      <status>Localization:*</status>
      <status>Published:*</status>
     </statuses>
    </type>
    <type name="topic">
     <statuses>
      <status>Authoring:*</status>
      <status>Localization:*</status>
      <status>Published:*</status>
     </statuses>
    </type>
    <type name="map">
     <statuses>
      <status>Authoring:*</status>
      <status>Localization:*</status>
      <status>Published:*</status>
     </statuses>
    </type>
    <type name="snapshot">
     <statuses>
      <status>Authoring:*</status>
      <status>Localization:*</status>
     </statuses>
    </type>
    <type name="collaborative-review">
     <statuses>
      <status>Authoring:*</status>
     </statuses>
    </type>
```

```
   </current>
   <users>
    <roles/>
    <groups>
     <group name="System Administrators"/>
     <group name="Everyone"/>
    </groups>
   </users>
  </condition>
 </conditionset>
</method>
```

10. Locate the `Change Status` method.

```
<method name="Change Status" type="front-end" hide-when-disabled="false">
```

11. Add the following line to include the new `collaborative-review` type to the condition.

```
<type name="collaborative-review">
 <statuses>
  <status>Authoring:*</status>
 </statuses>
</type>
```

The result should look like the following example:

```
<method name="Change Status" type="front-end" hide-when-disabled="false">
 <!-- 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>
     <status>Localization:*</status>
    </statuses>
   </type>
   <type name="topic">
    <statuses>
     <status>Authoring:*</status>
     <status>Localization:*</status>
    </statuses>
   </type>
   <type name="image">
    <statuses>
     <status>Authoring:*</status>
     <status>Localization:*</status>
    </statuses>
   </type>
   <type name="resource">
    <statuses>
     <status>Authoring:*</status>
     <status>Localization:*</status>
    </statuses>
   </type>
   <type name="snapshot">
    <statuses>
     <status>Authoring:*</status>
    </statuses>
```

```
   </type>
   <type name="build-manifest">
    <statuses>
     <status>Authoring:*</status>
    </statuses>
   </type>
   <type name="collaborative-review">
    <statuses>
     <status>Authoring:*</status>
    </statuses>
   </type>
  </current>
  <!-- Action user must be in this list -->
  <users>
   <roles/>
   <groups>
    <group name="System Administrators"/>
    <group name="Everyone"/>
   </groups>
  </users>
 </condition>
```

12. Locate the `Delete` method:

```
<method name="Delete" type="front-end"  hide-when-disabled="true">
```

13. Locate the condition that contains the `map` type for the Authoring status:

For example:

```
<type name="map">
 <statuses>
  <status>Authoring:*</status>
 </statuses>
</type>
```

14. Below the `map` type, add the following lines to include the new `collaborative-review` type to the condition:

```
<type name="collaborative-review">
 <statuses>
  <status>Authoring:*</status>
 </statuses>
</type>
```

The result should look like the following example:

```
<method name="Delete" 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>
   <!-- Object on which action is taken -->
   <current>
    <type name="topic">
     <statuses>
      <status>Authoring:*</status>
```

```
    </statuses>
   </type>
  </current>
  <!-- Action user must be in this list -->
  <users>
   <roles>
    <role name="Information Architect"/>
    <role name="Writer"/>
   </roles>
   <groups>
    <group name="System Administrators"/>
   </groups>
  </users>
 </condition>
 <condition>
  <!-- Object on which action is taken -->
  <current>
   <type name="map">
    <statuses>
     <status>Authoring:*</status>
    </statuses>
   </type>
   <type name="collaborative-review">
    <statuses>
     <status>Authoring:*</status>
    </statuses>
   </type>
  </current>
  <!-- Action user must be in this list -->
  <users>
   <roles>
    <role name="Writer"/>
    <role name="Information Architect"/>
   </roles>
   <groups>
    <group name="System Administrators"/>
   </groups>
  </users>
 </condition>
```

15. Locate the following line:

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

16. Just before this line, add the following new methods:

```
<method name="CreateReview" type="front-end" multiselect-disabled="true"
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>
   <!-- Object on which action is taken -->
   <current>
    <type name="map">
     <statuses>
      <status>Authoring:*</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
```

```
   <users>
    <roles>
     <role name="Writer"/>
    </roles>
    <groups>
     <group name="System Administrators"/>
    </groups>
   </users>
  </condition>
 </conditionset>
</method>
<method name="CloseReview" type="front-end" multiselect-disabled="true"
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>
   <!-- Object on which action is taken -->
   <current>
    <type name="map">
     <statuses>
      <status>Authoring:*</status>
     </statuses>
    </type>
    <type name="collaborative-review">
     <statuses>
      <status>Authoring:open</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
   <users>
    <roles>
     <role name="Writer"/>
    </roles>
    <groups>
     <group name="System Administrators"/>
    </groups>
   </users>
  </condition>
 </conditionset>
</method>
<method name="SetReviewAsDone" type="front-end" multiselect-disabled="true"
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>
   <!-- Object on which action is taken -->
   <current>
    <type name="map">
     <statuses>
      <status>Authoring:*</status>
     </statuses>
    </type>
    <type name="collaborative-review">
     <statuses>
      <status>Authoring:closed</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
```

```
   <users>
    <roles>
     <role name="Writer"/>
    </roles>
    <groups>
     <group name="System Administrators"/>
    </groups>
   </users>
  </condition>
 </conditionset>
</method>
<method name="ViewAnnotations" type="front-end" multiselect-disabled="true"
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>
   <!-- Object on which action is taken -->
   <current>
    <type name="map">
     <statuses>
      <status>Authoring:*</status>
     </statuses>
    </type>
    <type name="collaborative-review">
     <statuses>
      <status>Authoring:*</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
   <users>
    <roles>
     <role name="Writer"/>
    </roles>
    <groups>
     <group name="System Administrators"/>
    </groups>
   </users>
  </condition>
 </conditionset>
</method>
```

17. Locate the following line:

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

18. In the `<availablemethods>` element, add the following actions:

Note: IXIASOFT recommends that you add them in alphabetical order.

```
<action name="CloseReview" />
<action name="CreateReview" />
<action name="SetReviewAsDone" />
<action name="ViewAnnotations" />
```

19. Save, close, and check in the file.

Update the display.xml configuration file

A few mandatory changes are required in the display.xml file to support new features.

To update the file:

1. **Expand the Content Store's Repository node and browse to** */system/conf/* **to locate the** *display.xml* **file.**
2. **Right-click the file and click Check Out.**
3. **Double-click the file to open it in the XML editor area.**
4. **In the `columns` element, add the following `key` elements:**

   Note: IXIASOFT recommends that you add them in alphabetical order.

```
<key halign="LEFT" label="Marked as Finished"
name="completedReviewAssignmentCount" sortOrder="ASC" sortType="NUMERIC"
type="Index" visibility="255" width="110"/>
```

```
<key halign="LEFT" label="Total Reviewers" name="totalReviewAssignmentCount"
sortOrder="ASC" sortType="NUMERIC" type="Index" visibility="255" width="110"/>
```

5. **Save, close, and check in the file.**

Update the equivalence.xml configuration file

A few mandatory changes are required in the equivalence.xml file to support new features.

To update the file:

1. **Expand the Content Store's Repository node and browse to** */system/conf/* **to locate the** *equivalence.xml* **file.**
2. **Right-click the file and click Check Out.**
3. **Double-click the file to open it in the XML editor area.**
4. **In the `equivalences` element, add the following lines to include the new types for the Collaborative Review:**

```
<equivalence type="collaborative-review" standardSearch="true" groupName="Others">
 <object type="collaborative-review" icon="/system/conf/icons/manifest.gif"/>
</equivalence>
```

```
<equivalence type="annotations" standardSearch="false" groupName="Others">
 <object type="annotations"/>
</equivalence>
```

5. **Save, close, and check in the file.**

Update the roles.xml configuration file

A few mandatory changes are required in the roles.xml file to support new features.

To update the file:

1. **Expand the Content Store's Repository node and browse to** */system/conf/* **to locate the** *roles.xml* **file.**
2. **Right-click the file and click Check Out.**
3. **Double-click the file to open it in the XML editor area.**
4. **Locate the Writer role or the equivalent in your deployment:**

   For example:

```
<role maxassignee="1" name="Writer" topworkload="10">
```

5. **Add the following lines to include the new `Collaborative-Review` doctype to the role:**

```
<doctype name="Collaborative-Review">
 <timeline value="Active">
  <status value="Authoring:open"/>
  <status value="Authoring:closed"/>
 </timeline>
</doctype>
```

```
<role maxassignee="1" name="Writer" topworkload="10">
 <doctype name="Map">
  <timeline value="Active">
   <status value="Authoring:work"/>
   <status value="Published:done"/>
  </timeline>
  <timeline value="Retained">
   <status value="Authoring:approval"/>
  </timeline>
 </doctype>
 <doctype name="Topic">
  <timeline value="Incoming">
   <status value="Authoring:contribute"/>
   <status value="Authoring:review"/>
  </timeline>
  <timeline value="Active">
   <status value="Authoring:work"/>
  </timeline>
 </doctype>
 <doctype name="Image">
  <timeline value="Active">
   <status value="Authoring:work"/>
  </timeline>
 </doctype>
 <doctype name="Resource">
  <timeline value="Active">
   <status value="Authoring:work"/>
  </timeline>
 </doctype>
 <doctype name="Collaborative-Review">
```

```
  <timeline value="Active">
   <status value="Authoring:open"/>
   <status value="Authoring:closed"/>
  </timeline>
 </doctype>
</role>
```

6. **Locate the Reviewer role or the equivalent in your deployment:**

   For example:

```
<role defaultvotingsystem="Unanimous" name="Reviewer" topworkload="25">
```

7. **Add the following lines to include the new `Collaborative-Review` doctype to the role:**

```
<doctype name="Collaborative-Review">
 <timeline value="Active">
  <status value="Authoring:open"/>
 </timeline>
</doctype>
```

```
<role defaultvotingsystem="Unanimous" name="Reviewer" topworkload="25">
 <doctype name="Map">
  <timeline value="Active">
   <status value="Authoring:review"/>
   <status value="Authoring:approval"/>
  </timeline>
 </doctype>
 <doctype name="Topic">
  <timeline value="Active">
   <status value="Authoring:review"/>
  </timeline>
 </doctype>
 <doctype name="Collaborative-Review">
  <timeline value="Active">
   <status value="Authoring:open"/>
  </timeline>
 </doctype>
</role>
```

8. **Save, close, and check in the file.**

Update the triggers.xml configuration file

A mandatory change is required in the *triggers.xml* file to support a new feature.

The new triggers are designed to support email notification to reviewers when a Collaborative Review assigned to them is open in DITA CMS Web. The triggers are disabled by default, but are ready to be enabled when you deploy Collaborative Review.

To update the file:

1. **Expand the Content Store's Repository node and browse to** */system/conf/* **to locate the** *triggers.xml* **file.**

2. **Right-click the file and click Check Out.**
3. **Double-click the file to open it in the XML editor area.**
4. **At the end of the file after the last trigger, add the following:**

```
<!-- trigger that creates the tickets for email notifications when Collaborative Review
is assigned -->
<!-- <trigger apply-to="changeStatus" class="com.ixiasoft.cms.triggers.CreateTicketTrigger"

  name="mailTicket" objtype="collaborative-review" schedule="after">
        <parameters>
            <param name="initial-status" value="Authoring:closed"/>
            <param name="end-status" value="Authoring:open"/>
        </parameters>
    </trigger>

  <trigger apply-to="assignTo" class="com.ixiasoft.cms.triggers.assignments.CreateTicket"

  name="mailTicket" objtype="collaborative-review" schedule="after">
        <parameters>
            <param name="valid-status" value="Authoring:open"/>
        </parameters>
    </trigger>-->
```

5. **Save, close, and check in the file.**

(Optional) Import custom dictionary files for the DITA CMS standard dictionaries

The Content Store is missing the initial empty custom dictionary files required for users to add custom words to the Dictionary Manager.

> **Important:** If the */system/dicts/customs* folder already exists in your Content Store and it already contains the necessary custom dictionary files, do not import the empty custom dictionary files or all your existing custom words will be lost.

When a dictionary is added to DITA CMS, it requires both the language's standard dictionary files as well as its matching custom dictionary files. For more information, see the instructions for dictionaries in the *DITA CMS Administrator's Guide*.

By default, the Content Store contains the following standard dictionaries:

- English (United States): en-us.aff and en-us.dic
- French (France): fr-fr.aff and fr-fr.dic

For the Dictionary Manager to work, the custom dictionary files for the English (United States) and French (France) standard dictionaries must be imported into the Content Store.

To import the files:

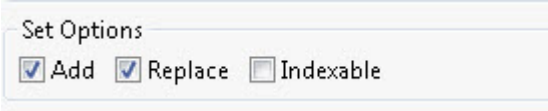
1. **Right-click the *Repository* node and select Insert Documents.**

   The **Insert Documents Dialog** window appears.

2. **Click Add Folder and browse to the** *optional_customdictionaries* **folder in the upgrade package directory (for example,** *C:\temp\DITA_CMS_4.4_upgrade_package\optional_customdictionaries***).**
3. **Select the `system` folder and click OK.**

   The system folder is listed in the **Insert Documents Dialog** window.
4. **Select the Add and Replace options in the Set Options panel and leave Indexable unchecked.**

   For example:

5. **Click OK to import the system files.**

2 Upgrade your Content Store to current release

Topics:

To upgrade your Content Store to the latest DITA CMS version, you must import the required files and perform all the configuration changes listed in this section.

Download the DITA CMS Core 4.5 upgrade package

The DITA CMS Core 4.5 upgrade package contains files required to upgrade your Content Store to the latest 4.5 version.

To download the DITA CMS upgrade package:

1. **Go to the following URL:**

   http://cms.ixiasoft.com/downloads/4.4_to_4.5_upgrade/

2. **Copy the upgrade package appropriate for your Content Store:**

   - If your company's DTD plugin uses DITA 1.2, download *DITA_CMS_4.5_DITA1.2_upgrade_package-[date].zip* file to your computer.
   - If your company's DTD plugin uses DITA 1.3, download *DITA_CMS_4.5_DITA1.3_upgrade_package-[date].zip* file to your computer.

3. **Unzip the upgrade package to a working directory.**

   For example:

   ```
   C:\temp\
   ```

   Note: Throughout this document, the `%UpgradePackageDir%` expression is used to represent the upgrade package directory.

The upgrade package contains the following directories:

- `base-upgrade`: Contains the files required to upgrade to DITA CMS
- `drm-upgrade`: Contains the files required to upgrade the Dynamic Release Management module to the latest release, if any are required

Back up your Content Store

If you haven't already done so, make a backup copy of your Content Store.

Perform an on-demand backup of the docbase

If you need to manually trigger a backup of the docbase, you can launch the backup from the TEXTML Server interface.

Note: Content Store data is stored in a TEXTML Server docbase. To back up a Content Store, you back up the TEXTML Server docbase that stores the Content Store data.

To trigger a backup:

1. **On the computer where the TEXTML Server is installed, create a directory for the backups.**
2. **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.**
3. **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.**
4. **When the Connect as dialog opens, type your username and password and click OK.**
5. **Double-click the name of your docbase to open a connection to the Content Store.**
6. **Right-click the docbase and select Backup.**

   The **Start backup** window opens:

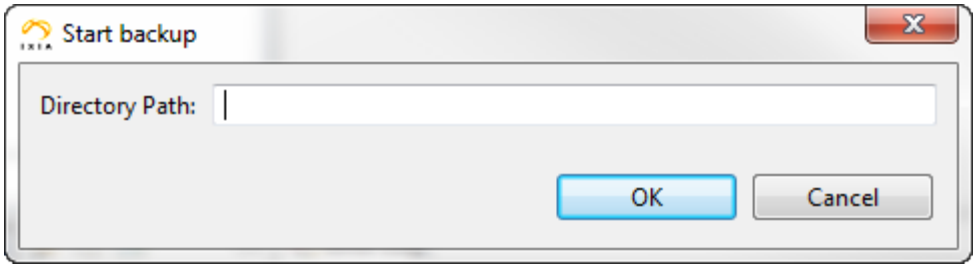


Figure 3: Backing up a docbase

> **Important note:**
>
> BEWARE: If the destination folder is not empty, TEXTML Server deletes the contents of the folder before starting the backup.

7. **In the Directory Path box, type the full path to the folder you created for the docbase backups.**

   For example (replace `[version]` by the release version) :

```
C:\docbase_backups\[version]\docbaseA
```

```
/docbase_backups/[version]/docbaseA
```

> **Note:** The TEXTML Server service or daemon ***must*** have write privileges to the top-level folder storing the backups; for example, the *docbase_backups* directory.

8. **Click OK.**

   The backup is initiated. To verify, look in the **Properties** view under ***General*** > ***Status***. When the backup is initiated, the value for **Status** changes from **Ready** to **Ready Backing-up**.

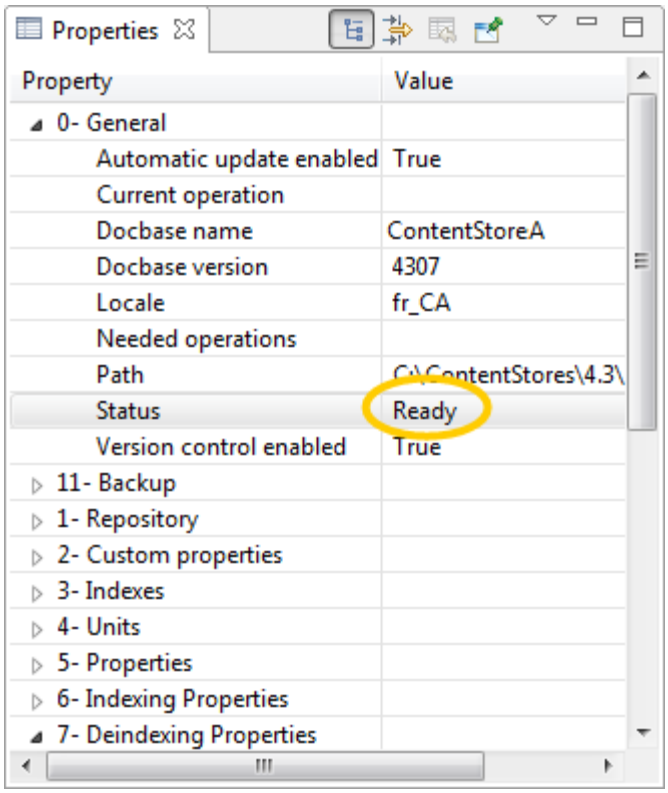


Figure 4: Example of Properties view:

9. **Wait while the backup occurs. Depending on the size of the docbase and your server activity, this could take considerable time.**

   Note: While the backup is occurring, a temporary file is created (*backuperror.tmp*) under the backup folder. If no errors occur during backup, this file is deleted and a new file called *backup.info* is created when the backup is complete.

10. **To confirm that the backup was successful, look in your target directory:**

    - If the *backup.info* file is present, then the backup was successful.
    - It the backup is completed but the *backuperror.tmp* is present, then an error occurred and the backup is corrupted.

    If an error occurs, or to get more information about the backup operation, look at the logs. They are available in the log directory.

On Windows:

```
%program_data%\IxiaSoft\TextmlServer[TEXTML Server version]\Log
```

For example:

```
C:\ProgramData\IxiaSoft\TextmlServer[TEXTML Server version]\Log
```

On Linux:

```
<textml_install_dir>/<instance_name>/log/
```

For example:

```
/opt/ixiasoft/textmlserver/ts01/log/
```

Import the Core configuration files

This procedure describes how to import the upgrade's configuration files for DITA CMS Core.

> **Important:** If you have incorporated your specializations or constraints directly in the *IxiaDitabase.dtd* file in the *com.ixiasoft.dita.dtd* plugin instead of creating your own company's DTD plugin (for example,*com.company.dita.dtd*), do NOT import the upgrade configuration files and contact IXIASOFT Services team.

To verify your plugin:

1. Expand the Content Store's Repository node and browse to *system**conf* to locate the *systemid.xml* file.
2. Verify the value set in the `mapdefault` attribute:
   - If the value is: `<id mapdefault="-//COMPANY//DTD IXIA DITA Map//EN"` where COMPANY is the name of company, then you have created your own company's plugin and you can proceed with the upgrade.
   - If the value is: `<id mapdefault="-//IXIA//DTD IXIA DITA Map//EN"`, then you cannot proceed and must call IXIASOFT Services team.

To import the new files into your Content Store:

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.**

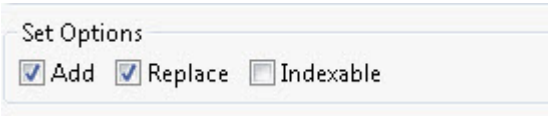
c) **Click OK.**

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 the** *Repository* **node and select Insert Documents.**

   The **Insert Documents Dialog** window appears.
6. **Click Add Folder and browse to the** *base-upgrade\system* **folder in the upgrade package directory (for example,** *C:\temp\DITA_CMS_[version]_[DITA version]_upgrade_package\base-upgrade\system***).**
7. **Select the `system` folder and click OK.**

   The system folder is listed in the **Insert Documents Dialog** window.
8. **Select the Add and Replace options in the Set Options panel and leave Indexable unchecked.**

   For example:



9. **Click OK to import the system files.**
10. **FOR DRM mode: No additional configuration files need to be imported for this release.**

Update the system configuration

Each of the topics in this section describe manual edits that must be made to specific configuration files. All of the following changes to the configuration files are mandatory.

Update your company's DTD plugins

Some mandatory changes are required in your company's DTD plugins for the IXIASOFT MathML integration whether you use MathML or not.

Making these changes now even if you are not using MathML ensures that you will have the correct information in the files in the future.

To update the files: for example: *com.company.dita.dtd*

1. **Expand the Content Store's Repository node and browse to** *\system\plugins* **to locate your company's DTD plugins folder; for example,** *com.[company].dita.dtd* **where [company] is the name of your company.**
2. **Open your company's DTD plugins folder (***com.company.dita.dtd***) and open the** *dtd* **folder. For each file in the** *dtd* **folder, replace the following lines as described:**
   a) **Right-click a file and click Check Out.**
   b) **Double-click the file to open it in the XML editor area.**
   c) **Replace each of the following lines (even if it appears only in a commented out section):**
      - Find `-//OASIS//ENTITIES DITA 1.3 MathML Domain//EN` and replace with `-//IXIA//ENTITIES DITA 1.3 MathML Domain//EN`
      - Find `-//OASIS//ELEMENTS DITA 1.3 MathML Domain//EN` and replace with `-//IXIA//ELEMENTS DITA 1.3 MathML Domain//EN`
      - Find `-//OASIS//ENTITIES DITA 1.3 Equation Domain//EN` and replace with `-//IXIA//ENTITIES DITA 1.3 Equation Domain//EN`
      - Find `-//OASIS//ELEMENTS DITA 1.3 Equation Domain//EN` and replace with `-//IXIA//ELEMENTS DITA 1.3 Equation Domain//EN`
   d) **Save, close, and check in the file.**
   e) **Repeat with each file.**

Repair the daily_reminder.xsl template

An error in the version 4.4.42 daily_reminder.xsl template must be corrected.

If you imported the DITA CMS Core version 4.4.42 daily_reminder.xsl template into the Content Store, then the following change is required to repair the parser error.

1. **Expand the Content Store's Repository node and browse to** *\system\scheduler\templates* **to locate the** *daily_reminder.xsl* **file.**
2. **Right-click the file and click Check Out.**
3. **Double-click the file to open it in the XML editor area.**
4. **Locate the following `div` class:**

```
<div class="intro">
 <xsl:text>These items have been assigned to you and are due in </xsl:text>
 <xsl:value-of select="$days"/>
 <xsl:text> days or less. You can view your assignments on the
  <xsl:element name="a">
   <xsl:attribute name="href"><xsl:value-of select="$WCR"/></xsl:attribute>DITA CMS Web
```

```
  </xsl:element>
 .</xsl:text>
</div>
```

5. **Replace the entire `div` class with the following and reapply your customizations, if any, as required:**

```
<div class="intro">
 <xsl:text>These items have been assigned to you and are due in </xsl:text>
 <xsl:value-of select="$days"/>
 <xsl:text> days or less. You can view your assignments on the </xsl:text>
 <xsl:element name="a">
  <xsl:attribute name="href"><xsl:value-of select="$WCR"/></xsl:attribute>DITA CMS Web
 </xsl:element>
 <xsl:text>.</xsl:text>
</div>
```

Tip: The corrections to the `div` class are highlighted in bold.

6. **Save, close, and check in the file.**

(DRM only) Update the accessrights.xml configuration file

A mandatory change is required to correct an error in the *accessrights.xml* file for deployments using the Dynamic Release Management.

To update the file:

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** *accessrights.xml* **file.**
6. **Right-click the file and click Check Out.**
7. **Double-click the file to open it in the XML editor area.**
8. **Locate the `ChangeLayer` method:**

```
<method name="ChangeLayer" type="front-end">
```

9. **In the `ChangeLayer` method, locate the condition which contains the `version` type.**

   For example:

```
<method name="ChangeLayer" type="front-end">
 <!-- 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="version">
     <statuses>
      <status>Authoring:development</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
   <users>
    <roles>
     <role name="Information Architect"/>
    </roles>
    <groups>
     <group name="System Administrators"/>
    </groups>
   </users>
  </condition>
 </conditionset>
</method>
```

10. **Replace the `version` type with the following lines:**

```
<type name="product">
 <statuses>
  <status>Authoring:open</status>
 </statuses>
</type>
```

The result should look like the following example:

```
<method name="ChangeLayer" type="front-end">
 <!-- 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="product">
     <statuses>
      <status>Authoring:open</status>
     </statuses>
    </type>
   </current>
   <!-- Action user must be in this list -->
   <users>
    <roles>
     <role name="Information Architect"/>
    </roles>
    <groups>
     <group name="System Administrators"/>
    </groups>
```

```
   </users>
  </condition>
 </conditionset>
</method>
```

11. Save, close, and check in the file.

Update the DITA CMS version in the configuration

After you have updated the Content Store to a new build, you should change the build number in the textmlservercfg.properties file to reflect the current version number. This will also force users to update their DITA CMS Eclipse Client to the installed build version.

To edit the textmlservercfg.properties file:

1. **Expand the Content Store's Repository node and browse to** */system/conf/* **to locate the** *textmlservercfg.properties* **file.**
2. **Right-click** *textmlservercfg.properties* **and click Check Out.**
3. **Double-click the file to open it in the XML editor area.**
4. **Set the value of `ixia.cms.version` to the build of the DITA CMS that you are installing.**

   For example:

   ```
   ixia.cms.version=4.5.35
   ```

5. **Save, close, and check in the file.**
6. **Inform users of the changes and request that they close and reopen their DITA CMS to apply the changes.**

3 Upgrade the DITA CMS Eclipse Client

To upgrade the DITA CMS Eclipse Client, you upgrade the plugins to the latest version of the DITA CMS.

See the *DITA CMS Eclipse Client Installation Guide* for the upgrade procedure.

4 Install a new Output Generator and upgrade your configuration

Topics:

This section describes how to upgrade an existing installation of the Output Generator.

Install a new Output Generator

Install a new Output Generator as described in the *Output Generator Installation Guide* and test your installation.

Migrate your Output Generator configuration to the new version when you are using DITA-OT 1.8.5 or 2.3.1

After you have installed and tested the latest version of the Output Generator, you need to migrate your configuration to the new version.

> **Remember:** If you have more than one DITA-OT in your Output Generator, remember to perform the migration for each of them.

To migrate the configuration:

1. **Copy the contents of the** *%OutputGenDir%/conf/client/* **folder from your previous version to the folder in the new version.**

   For example, copy the contents of the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[previous version]\conf\client

   to the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[new version]\conf\client

2. **Copy the contents of the** *%OutputGenDir%/data/resources/client/* **folder from your previous version to the folder in the new version.**

   For example, copy the contents of the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[previous version]\data\resources\client

   to the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[new version]\data\resources\client

3. **Copy the contents of the** *%OutputGenDir%/libs/client/* **folder from your previous version to the folder in the new version.**

   For example, copy the contents of the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[previous version]\libs\client

   to the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[new version]\libs\client

4. **Copy all your custom transformation scenarios (e.g.,** *conductor-acme.xml***) from the** *%OutputGenDir%/data/* **folder to the folder in the new version.**
5. **Copy all the** **-client.xml* **files from the** *%OutputGenDir%/data/* **folder to the folder in the new version.**

   For example, copy the following files:

   - *outgen-init-client.xml*
   - *commontargets-client.xml*
   - *bmanifest-extension-client.xml*
   - *bmanifest-init-client.xml*

6. **Copy the** *outgen-init.properties* **file from the** *%OutputGenDir%/data/* **folder in the previous version and paste it in the same folder in the new version.**
7. **Copy the** *outgen-init-cr.properties* **file from the** *%OutputGenDir%/data/* **folder in the previous version and paste it in the same folder in the new version.**
8. **Export a copy of your company's DTD plugin from the Content Store:**
   a) **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**

      1. 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.
      2. When the **Connect as** dialog opens, type your username and password and click **OK**.
      3. Double-click the name of your docbase to open a connection to the Content Store.

   b) **Expand the Content Store's Repository node and browse to** */system/plugins* **to locate your company's DTD plugin; for example:** *com.company.dita.dtd*
   c) **Right-click the plugin and click Export.**

   The plugin is saved as a folder locally as *Content\system\plugins\com.company.dita.dtd* where company is the name of your company.

   > **Important:** DO NOT export the *com.ixiasoft.dita.dtd* plugin or *com.ixiasoft.dita13.dtd* and use it in the Output Generator.

9. **Copy your** *com.company.dita.dtd* **plugin from the location where you exported it and paste it to the new version of the Output Generator as follows:**

   - If you are using DITA-OT 1.8.5: Paste it in the *%OutputGenDir%\data\DITA-OT1.8.5\plugins* folder.
   - If you are using DITA-OT 2.3.1: Paste it in the *%OutputGenDir%\data\dita-ot-2.3.1\plugins* folder.

10. **Transfer your custom transformation plugins from your previous version of the Output Generator to the new version as follows:**

Option	Description
If you are using DITA-OT 1.8.5	1. Copy your custom plugins from the *%OutputGenDir%\data\DITA-OT1.8.5\plugins* folder in your previous version. 2. Paste them into the *plugins* folder in the new version.
If you are using DITA-OT 2.3.1	1. Copy your custom plugins from the *%OutputGenDir%\data\dita-ot-2.3.1\plugins* folder in your previous version. 2. Paste them into the *plugins* folder in the new version.

Migrate your Output Generator configuration to the new version when you are using a different DITA-OT version

After you have installed and tested the latest version of the Output Generator, you need to migrate your configuration to the new version.

To migrate the configuration:

1. **Copy the contents of the** *%OutputGenDir%/conf/client/* **folder from your previous version to the folder in the new version.**

   For example, copy the contents of the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[previous version]\conf\client

   to the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[new version]\conf\client

2. **Copy the contents of the** *%OutputGenDir%/data/resources/client/* **folder from your previous version to the folder in the new version.**

   For example, copy the contents of the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[previous version]\data\resources\client

   to the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[new version]\data\resources\client

3. **Copy the contents of the** *%OutputGenDir%/libs/client/* **folder from your previous version to the folder in the new version.**

   For example, copy the contents of the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[previous version]\libs\client

   to the following folder:

   C:\ixiasoft\OutputGenerators\Prod_[new version]\libs\client

4. **Copy all your custom transformation scenarios (e.g.,** *conductor-acme.xml***) from the** *%OutputGenDir%/data/* **folder to the folder in the new version.**
5. **Copy all the** **-client.xml* **files from the** *%OutputGenDir%/data/* **folder to the folder in the new version.**

   For example, copy the following files:

   - *outgen-init-client.xml*
   - *commontargets-client.xml*
   - *bmanifest-extension-client.xml*
   - *bmanifest-init-client.xml*

6. **Copy the** *outgen-init.properties* **file from the** *%OutputGenDir%/data/* **folder in the previous version and paste it in the same folder in the new version.**
7. **Copy the** *outgen-init-cr.properties* **file from the** *%OutputGenDir%/data/* **folder in the previous version and paste it in the same folder in the new version.**
8. **Copy the entire DITA-OT folder from your previous version of the Output Generator and paste it in the same location in the new version of the Output Generator. For example, the** *%OutputGenDir%/data/DITA-OTx.x.x* **folder.**
9. **Export a copy of your DTD from the Content Store:**

   a) **Open the TEXTML Administration perspective by clicking the TEXTML Administration shortcut on the tool bar. If the shortcut is not displayed, follow these steps:**

      1. 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.
      2. When the **Connect as** dialog opens, type your username and password and click **OK**.
      3. Double-click the name of your docbase to open a connection to the Content Store.

   b) **Expand the Content Store's Repository node and browse to** */system/plugins* **to locate your company's DTD plugin; for example:** *com.company.dita.dtd*

   c) **Right-click the plugin and click Export.**

The plugin is saved as a folder locally as *Content/system/plugins/com.company.dita.dtd* where company is the name of your company.

Important: DO NOT export the *com.ixiasoft.dita.dtd* plugin or *com.ixiasoft.dita13.dtd* and use it in the Output Generator.

10. **Copy** *com.company.dita.dtd* **plugin from the location where you exported it and paste it into your** *DITA-OT/plugins* **folder in the new version of the Output Generator (for example, the** *%OutputGenDir%/data/DITA-OTx.x.x/plugins* **folder).**
11. **In your** *DITA-OT/plugins* **folder in the new version of the Output Generator (for example, the** *%OutputGenDir%/data/DITA-OTx.x.x/plugins* **folder), add or replace the plugins with the following plugins:**

Option	Description
If your DITA-OT version is lower than 1.6.2	Please contact IXIASOFT customer support.
If your DITA-OT is in the DITA-OT 1.x family and 1.6.2 or higher	1. Browse to the location where you extracted the installation files for the new version of the Output Generator. 2. Copy the following plugins from the DITA-OT 1.8.5 folder (for example: *%OutputGenDir%/data/DITA-OT1.8.5/plugins*): • *com.ixiasoft.dita.dtd* • *com.ixiasoft.images* • *com.ixiasoft.pdf.review* • *org.oasis-open.dita.mathml.doctypes* • *org.w3c.mathml3* • *org.w3c.svg1.0* • *org.w3c.svg1.1* 3. Paste them into your *DITA-OT/plugins* folder in the new version of the Output Generator (for example, the *%OutputGenDir%/data/DITA-OTx.x.x/plugins* folder).
If your DITA-OT is in the DITA-OT 2.x family	1. Browse to the location where you extracted the installation files for the new version of the Output Generator. 2. Copy the following plugins from the DITA-OT 2.3.1 folder (for example: *%OutputGenDir%/data/dita-ot-2.3.1/plugins* folder): • *com.ixiasoft.dita13.dtd*

Option	Description
	• *com.ixiasoft.images* • *com.ixiasoft.pdf.review20* • *com.ixiasoft.xhtml.cr* • *org.w3c.svg1.0* • *org.w3c.svg1.1*

3. Paste them into your *DITA-OT/plugins* folder in the new version of the Output Generator (for example, the *%OutputGenDir%/data/DITA-OTx.x.x/plugins* folder).
4. If you were using the Ixiasoft DITA 1.2 MathML3 integration (before it was available in DITA 1.3) in your previous version, copy the following plugin from the DITA-OT 1.8.5 folder (for example: *%OutputGenDir%/data/DITA-OT1.8.5/plugins*) and paste it in your *DITA-OT/plugins* folder in the new version of the Output Generator (for example, the *%OutputGenDir%/data/DITA-OTx.x.x/plugins* folder):
   - *org.oasis-open.dita.mathml.doctypes*

Configure the location of the DITA-OT catalog

When you install the Output Generator, you need to configure the location of the catalog for the DITA-OT version you are using.

The Output Generator needs the DITA-OT catalog to parse the content sent by the DITA CMS.

> **Note:** You can support multiple versions of the DITA-OT, as long as your DTDs are the same in all the versions. If you are using multiple versions of the DITA-OT, enter only one of the versions in the file below. You can enter any version you are using.

To configure the DITA-OT catalog to use in your deployment:

1. **Open the** *%OutputGenDir%/data/catalogs/catalog-dita-outgen.xml* **file.**

   > **Note:** If this is the first time that you are installing the Output Generator, open the *catalog-dita-outgen.xml.empty* file and save it as *catalog-dita-outgen.xml* (remove the *empty* suffix).

2. **Set the `<nextCatalog>` value to the location of the DITA-OT** *catalog-dita.xml* **file.**

By default, the Output Generator uses the DITA-OT 2.3.1 catalog. If you are not using DITA-OT 2.3.1 in your deployment, you need to specify the location of the catalog for the DITA-OT version you are using. For example:

```
<?xml version="1.0" ?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog" prefer="public">

<!-- Make sure only one of the following catalog is uncommented -->
<nextCatalog catalog="../DITA-OT1.8.5/catalog-dita.xml"/>
<!--<nextCatalog catalog="../dita-ot-2.3.1/catalog-dita.xml"/>-->

</catalog>
```

Note: The directory nomenclature used by the DITA-OT changed between versions 1.8 and 2.x, so make sure to use the actual directory name on the Output Generator machine when setting up this field.

3. **Save and close** *catalog-dita-outgen.xml*.

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-2.3.1-integrator.bat*: Runs the integrator on the DITA-OT 2.3.1

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:	• On Windows: In the *%OutputGenDir%/data/* directory, double-click the *_DITA-OT-1.8.5-integrator.bat* file. • On Linux: 1. In the *%OutputGenDir%/data/%OT_Dir%/* directory, run `startcmd.sh`.

	2. At the command prompt, enter: `ant -f integrator.xml` The integration build runs and you should soon see BUILD SUCCESSFUL.
DITA-OT 2.x family:	• On Windows: In the *%OutputGenDir%/data/* directory, double-click the *_DITA-OT-2.3.1-integrator.bat* file. • On Linux: In the *%OutputGenDir%/data/%OT_Dir%/bin* directory, enter the following command: `./ant -f ../integrator.xml (Linux)` The integration build runs and you should soon see BUILD SUCCESSFUL.

Restart the Output Generator and test your scenarios.

After you have run the integrator on all the DITA-OTs that you are using, the final step is to restart the Output Generator and test your transformation scenarios.

5 Upgrade the Scheduler

Topics:

This chapter describes how to upgrade an existing installation of the Scheduler.

Uninstall the Scheduler service

To uninstall the Scheduler service:

1. **Stop the Scheduler service.**
2. **Uninstall the Scheduler service:**
   - Windows: Run the *UninstallScheduler-NT.bat* in the `%SchedulerDir%\bin` directory.
   - Linux: Run the following command:

```
/sbin/chkconfig --del scheduler
```

Download and extract the Scheduler server files

This procedure describes how to extract the Scheduler files to a Linux or Windows server.

You will need one instance of Scheduler per DITA CMS Content Store.

To install Scheduler on a server:

1. **On the server, create a directory where the Scheduler will be installed; for example:**

```
Windows: C:\Ixiasoft\Scheduler\
Linux: /opt/ixiasoft/Scheduler/
```

   You must have the following permissions on the directory:

   - Windows: Read and write permissions
   - Linux: Read, write, and execute permissions (750)

   Note: The `%SchedulerDir%` expression will be used to represent the Scheduler directory.

2. **Go to the IXIASOFT Scheduler download page:**
   http://cms.ixiasoft.com/downloads/scheduler/
3. **Click the link to the latest version of the Scheduler and continue clicking the links until you reach the following files:**
   - *scheduler-<version>_bin.zip*
   - *scheduler-<version>_conf.zip*

   Where `<version>` is the version of the Scheduler.
4. **Download these files to the `%SchedulerDir%` directory.**

5. **Extract the files to the `%SchedulerDir%` directory.**
6. **When the files are extracted, delete the *.zip* files from the directory.**

   The following Scheduler directories are installed on your server:

   - **bin**: Contains the Scheduler bat files
   - **conf**: Contains the XML configuration files and templates
   - **libs**: Contains the Scheduler program libraries
   - **logs**: Initially empty; it will contain the Scheduler logs
   - **temp**: Initially empty; it will be used during processing

Install and start the Scheduler service (Windows)

This procedure describes how to install and start the Scheduler service on a Windows server.

Important notes:

- Because the Scheduler connects to the Content Store on the TEXTML Server, the TEXTML Server must be running when you start the Scheduler. Otherwise, the Scheduler service will not start.
- If you restart the TEXTML Server, you must also restart the Scheduler.

To install and start the Scheduler service on a Windows server:

1. **In Windows, click Start, right-click Command Prompt, and select Run as administrator.**
2. **Click Yes in the confirmation window.**

   A command-line interface opens.

3. **Navigate to the Scheduler bin file:**

   For example:

   ```
   cd C:\ixiasoft\Scheduler\bin
   ```

4. **Enter the following command:**

   ```
   InstallScheduler-NT.bat
   ```

   The Scheduler is installed.

5. **In the `\bin` folder, start the Scheduler service. Type:**

   ```
   net start Scheduler
   ```

The server confirms that the Scheduler has been started:

```
The Ixiasoft Scheduler service is starting....
The Ixiasoft Scheduler service was started successfully.
```

Install and start the Scheduler service (Linux)

This procedure describes how to install and start the Scheduler service on Linux.

To install and start the Scheduler service on a Linux server:

1. **Open a command-line interface to the Linux server.**
2. **As user `root`, create a symlink that points to scheduler.sh.**

```
ln -sf /opt/ixiasoft/scheduler/bin/scheduler.sh /etc/init.d/scheduler
```

3. **Install Scheduler as a service:**

```
/sbin/chkconfig --add scheduler
```

4. **Change the run level:**

```
/sbin/chkconfig --level 24 scheduler off
```

5. **Start the Scheduler service:**

```
/sbin/service scheduler start
```

6. **Check for successful startup:**

```
[root@writix bin]# cd -
/etc/init.d
[root@writix init.d]# ./scheduler console
Running Ixiasoft Scheduler...
```