

# DITACMS

## Upgrading the DITA CMS from 4.2 to 4.3



# Table of contents

<b>Chapter 1: Overview</b>	
Copyright notice	6
Upgrade steps	6
New in this documentation release	8
<b>Chapter 2: Upgrade your Content Store</b>	
Download the DITA CMS 4.3 upgrade package	12
Back up your Content Store	12
Perform an on-demand backup of the docbase	12
Import and configure system files	15
Update the Index Definition document	27
Configure your Content Store to use the DITA-OT 2.3.1 with DITA 1.2 29	
Recommended updates	31
Additional recommended updates	33
<b>Chapter 3: Upgrade the DITA CMS Eclipse Client</b>	
<b>Chapter 4: Install a new Output Generator and upgrade your configuration</b>	
What's new in the Output Generator 4.3	38
Install a new Output Generator	38
Migrate your Output Generator 4.2 configuration	38

<b>Configure the Output Generator to use the DITA-OT 2.3.1 with DITA 1.2</b>	<b>44</b>
<b>Configure the location of the DITA-OT catalog</b>	<b>47</b>
<b>Run the integrator on the DITA-OT</b>	<b>47</b>
<b>Restart the Output Generator and test your scenarios.</b>	<b>48</b>
<b>Chapter 5: Upgrade the Scheduler</b>	
<b>Uninstall the Scheduler 4.2 service</b>	<b>50</b>
<b>Download and extract the Scheduler server files</b>	<b>50</b>
<b>Install and start the Scheduler service (Windows)</b>	<b>51</b>
<b>Install and start the Scheduler service (Linux)</b>	<b>52</b>
<b>Replace the daily_reminder.xsl template in Scheduler</b>	<b>52</b>
<b>Chapter 6: Upgrade the WCR application</b>	
<b>Chapter 7: Upgrade the Web Author</b>	
<b>Download and extract the Web Author package</b>	<b>58</b>
<b>Upgrade the CMS Application Server</b>	<b>58</b>
<b>Upgrade the Web Author Application</b>	<b>60</b>
<b>Upgrade the online help</b>	<b>61</b>
<b>Test your DITA CMS Web installation</b>	<b>61</b>

# 1

## Overview

### Topics:

- [Copyright notice](#)
- [Upgrade steps](#)
- [New in this documentation release](#)

This document describes how to upgrade your DITA CMS 4.2 deployment to the latest DITA CMS 4.3 build.

It assumes that you are running the latest build of the DITA CMS 4.2 and that you have applied all the 4.2 configuration changes.

If that is not the case, please go to the following URL and apply the configuration changes described in the *Updating an Existing DITA CMS v4.2 Deployment (Update Guide)*:

<http://www.ixiasoft.com/en/products/dita-cms/documentation/4-2/dita-cms-42-release-notes/>

**Note:** If you are upgrading a DRM deployment, see the [DITA CMS 4.3 Upgrade Guide for DRM](#).

### Upgrading to DITA 1.3

If you want to upgrade to DITA 1.3, please see [Upgrading a Deployment to DITA 1.3](#)

## Copyright notice

---

©2017 IXIASOFT Technologies Inc. All rights reserved. Except as otherwise expressly permitted by IXIASOFT Technologies Inc., this publication, or parts thereof, may not be reproduced or distributed in any form, by any method, for any purpose.

This publication and the information contained herein are made available by IXIASOFT Technologies Inc. "as is." IXIASOFT Technologies Inc. disclaims all warranties, either express or implied, including but not limited to any implied warranties of merchantability or fitness for a particular purpose regarding these materials.

## Upgrade steps

---

To upgrade and use the DITA CMS 4.3, you need to upgrade the following components:

**Table 1: DITA CMS Core components**

Step	Component	Procedure
1.	Content Store	<b>Upgrade your Content Store</b> on page 11
2.	DITA CMS Eclipse Client	<b>Upgrade the DITA CMS Eclipse Client</b> on page 35
3.	Output Generator	<b>Install a new Output Generator and upgrade your configuration</b> on page 37
4.	Scheduler	<b>Upgrade the Scheduler</b> on page 49
5.	WCR	See <b>Table 2: Web components</b> on page 6
6.	Web Author	See <b>Table 2: Web components</b> on page 6

### Web components (DITA CMS Web, Web Author, WCR)

DITA CMS now provides DITA CMS Web, which replaces Web Author. The following table describes the procedures for installing and upgrading existing web components:

**Table 2: Web components**

4.2 deployment	4.3 deployment	Procedures
• no web component	• DITA CMS Web	• <b>Install DITA CMS Web</b>

4.2 deployment	4.3 deployment	Procedures
<ul style="list-style-type: none"> <li>WCR only</li> </ul>	<ul style="list-style-type: none"> <li>WCR only</li> </ul> <hr/> <ul style="list-style-type: none"> <li>DITA CMS Web</li> <li>WCR</li> </ul>	<ul style="list-style-type: none"> <li><b>Upgrade the WCR application</b> on page 55</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Uninstall Tomcat 6 and Glassfish 3</li> <li><b>Install the DITA CMS web</b></li> <li>Install the WCR as described in the <b>WCR Installation Guide</b> in the Tomcat 7 installation of DITA CMS Web</li> </ul> <p>Contact IXIASOFT Services for more information.</p>
<ul style="list-style-type: none"> <li>Web Author only</li> </ul>	<ul style="list-style-type: none"> <li>Web Author only</li> </ul> <hr/> <ul style="list-style-type: none"> <li>DITA CMS Web only</li> </ul> <hr/> <ul style="list-style-type: none"> <li>DITA CMS Web</li> <li>Web Author</li> </ul> <p><b>(This is not recommended)</b></p>	<ul style="list-style-type: none"> <li><b>Upgrade the Web Author</b> on page 57</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Uninstall Tomcat 6 and Glassfish 3</li> <li><b>Install the DITA CMS web</b></li> </ul> <hr/> <p>These two components cannot be installed on the same machine. To use both:</p> <ul style="list-style-type: none"> <li>Upgrade your existing Web Author installation on your current machine as described in <b>Upgrade the Web Author</b> on page 57</li> <li><b>Install DITA CMS Web</b> on another machine</li> </ul> <p>Contact IXIASOFT Services for more information.</p>
<ul style="list-style-type: none"> <li>Web Author</li> <li>WCR</li> </ul>	<ul style="list-style-type: none"> <li>Web Author</li> <li>WCR</li> </ul> <hr/> <ul style="list-style-type: none"> <li>WCR</li> <li>DITA CMS Web</li> </ul>	<ul style="list-style-type: none"> <li><b>Upgrade the Web Author</b> on page 57</li> <li><b>Upgrade the WCR application</b> on page 55</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Uninstall Tomcat 6 and Glassfish 3</li> <li><b>Install the DITA CMS web</b></li> <li>Install the WCR as described in the <b>WCR Installation Guide</b> in the Tomcat 7 installation of DITA CMS Web.</li> </ul>

4.2 deployment	4.3 deployment	Procedures
		Contact IXIASOFT Services for more information.
	<ul style="list-style-type: none"> <li>• Web Author</li> <li>• WCR</li> <li>• DITA CMS Web</li> </ul> <p><b>(This is not recommended)</b></p>	<p>Web Author and DITA CMS Web cannot be installed on the same machine. To use both and the WCR:</p> <ul style="list-style-type: none"> <li>• Upgrade your existing Web Author installation on your current machine as described in <b>Upgrade the Web Author</b> on page 57</li> <li>• <b>Install DITA CMS Web</b> on another machine</li> <li>• Install the WCR as described in the <b>WCR Installation Guide</b> in the Tomcat 7 installation of DITA CMS Web</li> </ul> <p>Contact IXIASOFT Services for more information.</p>

## New in this documentation release

### End of Support

The IXIASOFT Web Author and Web Collaborative Review modules are being merged into a unique DITA CMS Web component to take your content creation and reviewing processes to the next level. The DITA CMS Web component will include the authoring and reviewing features available today as well as new features, such as a map view, topic and map creation, as well as map editing.

As such, all support for Web Author and Web Collaborative Reviewer will stop by December 31st, 2017.

### August 2017

- Updated the following procedure to include configuration changes introduced in 4.3.33, 4.3.47, 4.3.57, and 4.3.61:
  - **Import and configure system files** on page 15

- **Update the Index Definition document** on page 27
- **Recommended updates** on page 31
- Added the upgrade path for the new DITA CMS web in **Upgrade steps** on page 6
- Added the following procedures to allow DITA 1.2 environments to use the new DITA-OT 2.3.1:
  - **Configure your Content Store to use the DITA-OT 2.3.1 with DITA 1.2** on page 29
  - **Configure the Output Generator to use the DITA-OT 2.3.1 with DITA 1.2** on page 44

## October 2016

- First release of this document.



# 2

## Upgrade your Content Store

### Topics:

- [Download the DITA CMS 4.3 upgrade package](#)
- [Back up your Content Store](#)
- [Import and configure system files](#)
- [Update the Index Definition document](#)
- [Configure your Content Store to use the DITA-OT 2.3.1 with DITA 1.2](#)
- [Recommended updates](#)

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 4.3 upgrade package

---

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

To download the DITA CMS 4.3 upgrade package:

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

[http://cms.ixiasoft.com/downloads/4.2\\_to\\_4.3\\_upgrade](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:\ixiasoft\
```

**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 (applies only if you are using the DRM module)
- `output-gen-upgrade`: Contains the files required to upgrade the Output Generator

## 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.

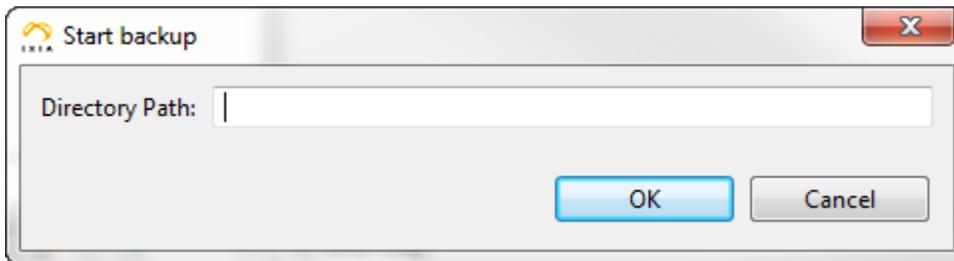
**Note:** Content Store data is stored in a TEXTML Server docbase. To back up a Content Store, you back up the TEXTML Server docbase that stores the Content Store data.

To trigger a backup:

- 1. On the computer where the TEXTML Server is installed, create 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**.

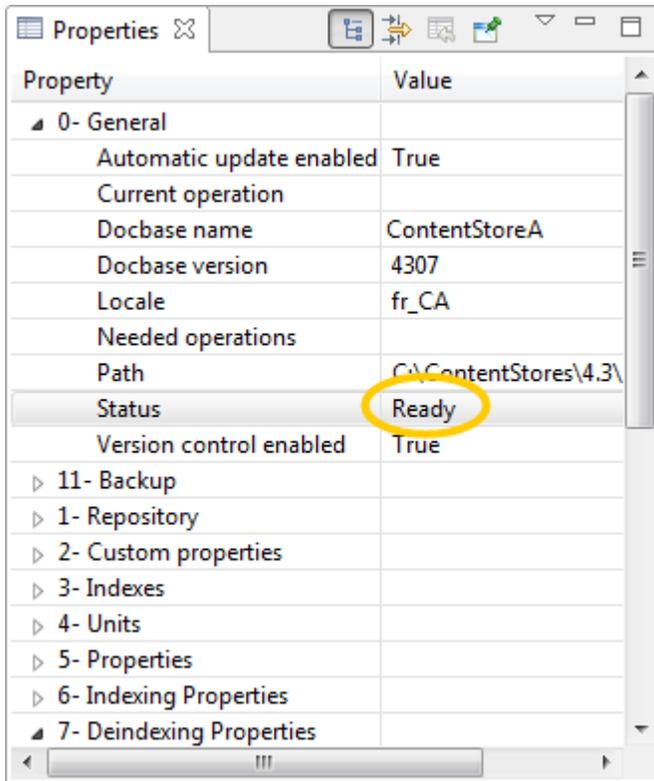


Figure 2: Example of Properties view:

**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.
- If 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 and configure system files

---

This procedure describes the system files to import and configure for DITA CMS.

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

For more information about these new features, see the *What's New in DITA CMS 4.4* document.

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.

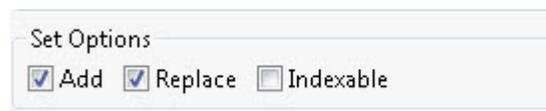
- b) Click **Add Folder** and browse to the *base-upgrade\system* folder in the upgrade package directory (for example, *C:\vixiasoft\DITA\_CMS\_4.3\_upgrade\_package\base-upgrade\system*).

- 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.

## 6. 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.**

## 7. 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.**

## 8. 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.**

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

**Note:** Perform this step even if you are not using DRM.

- 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.**

**10. 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.

**11. 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.**

## 12. 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.**

## 13. 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="*">

```

- 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">

```

```

<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.

#### 14. 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.**

## 15. 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.**

## 16. Add the Assignments Comments column to the `display.xml` file:

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

## 17. Edit the `resources.xml` file to add a type for non-DITA files and remove an unknown type, if present:

- Right-click the `/system/conf/resources.xml` file and click Check Out.**
- Double-click the file to open it in the XML editor area.**
- 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.

**18. 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.

**19. Update the DITA CMS version in the configuration:**

- a) Right-click the `/system/conf/textmlservercfg.properties` file and select Check out.
- b) Open the file.
- c) Set the value of `ixia.cms.version` to the build of the DITA CMS that you are installing.

For example:

```
ixia.cms.version=1.2.3
```

- d) Save and check in `textmlservercfg.properties`.

**20. 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 <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"/>
```

## 9. 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>
```

## 10. 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>
```

## 11. 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>
```

## 12. 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>
```

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

## Configure your Content Store to use the DITA-OT 2.3.1 with DITA 1.2

In DITA CMS 4.3, the Output Generator supports version 2.3.1 of the DITA-OT, which includes the DITA 1.3 DTDs. To use DITA 1.2 with the DITA-OT 2.3.1, you must update your DTDs.

**Note:** If you are still using *ixiaditabase.dtd* and not your own shell plugin to generate the output, IXIASOFT does not recommend using the DITA-OT 2.3.1.

These changes are required because when IXIASOFT integrated MathML into previous versions of the DITA CMS, the integration was based on a non-final version of the DITA 1.3 specification. The final integration, available in the DITA-OT 2.3.1, is very slightly different from the original IXIASOFT integration and may not be compatible with your MathML content.

**Note:** To use DITA-OT 2.3.1 with DITA 1.2, you will also need to upgrade the Output Generator, as described in [Configure the Output Generator to use the DITA-OT 2.3.1 with DITA 1.2](#) on page 44.

To use DITA-OT 2.3.1 with DITA 1.2:

### 1. Change the public IDs for the MathML integration to point to the IXIASOFT integration of the MathML DTDs in the *IxiaDitabase.dtd* file:

- a) In the `/system/plugins/com.ixiasoft.dita.dtd/dtd` folder, right-click the *IxiaDitabase.dtd* file and select **Check Out**.
- b) **Open the file.**
- c) **Look for the first MathML Integration section.**

For example:

```
<!-- ===== MathML Integration ===== -->
<!--
    Un-comment the following block to enable
    MathML elements directly inside topics    -->

<!ENTITY % mathml-d-dec
    PUBLIC "-//OASIS//ENTITIES DITA 1.3 MathML Domain//EN"
    "../../../org.oasis-open.dita.mathml.doctypes/dtd/mathmlDomain.ent"
>%mathml-d-dec;

<!ENTITY % equation-d-dec
    PUBLIC "-//OASIS//ENTITIES DITA 1.3 Equation Domain//EN"
    "../../../org.oasis-open.dita.mathml.doctypes/dtd/equationDomain.ent"
>%equation-d-dec;
```

**Note:** This section may be commented out if you are not using MathML in your deployment.

- d) **Change OASIS to IXIA in the public ID and remove the 1.3.**

For example:

```
<!-- ===== MathML Integration ===== -->
<!--
    Un-comment the following block to enable
    MathML elements directly inside topics    -->

<!ENTITY % mathml-d-dec
  PUBLIC "-//IXIA//ENTITIES DITA MathML Domain//EN"
    "../..org.oasis-open.dita.mathml.doctypes/dtd/mathmlDomain.ent"
>%mathml-d-dec;

<!ENTITY % equation-d-dec
  PUBLIC "-//IXIA//ENTITIES DITA Equation Domain//EN"
    "../..org.oasis-open.dita.mathml.doctypes/dtd/equationDomain.ent"
>%equation-d-dec;
```

e) **Look for the second MathML Integration section.**

For example:

```
<!-- ===== MathML Integration ===== -->
<!--
    Un-comment the following block to enable
    MathML elements directly inside topics    -->

<!ENTITY % mathml-d-def
  PUBLIC "-//OASIS//ELEMENTS DITA 1.3 MathML Domain//EN"
    "../..org.oasis-open.dita.mathml.doctypes/dtd/mathmlDomain.mod"
>%mathml-d-def;

<!ENTITY % equation-d-def
  PUBLIC "-//OASIS//ELEMENTS DITA 1.3 Equation Domain//EN"
    "../..org.oasis-open.dita.mathml.doctypes/dtd/equationDomain.mod"
>%equation-d-def;
```

**Note:** This section may be commented out if you are not using MathML in your deployment.

f) **Change OASIS to IXIA in the public ID and remove the 1.3.**

For example:

```
<!-- ===== MathML Integration ===== -->
<!--
    Un-comment the following block to enable
    MathML elements directly inside topics    -->

<!--
<!ENTITY % mathml-d-def
  PUBLIC "-//IXIA//ELEMENTS DITA MathML Domain//EN"
    "../..org.oasis-open.dita.mathml.doctypes/dtd/mathmlDomain.mod"
>%mathml-d-def;

<!ENTITY % equation-d-def
  PUBLIC "-//IXIA//ELEMENTS DITA Equation Domain//EN"
    "../..org.oasis-open.dita.mathml.doctypes/dtd/equationDomain.mod"
>%equation-d-def;
```

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

**2. Change the catalog for the MathML plugins to reflect the new public IDs:**

- a) In the `/system/plugins/org.oasis-open.dita.mathml.doctypes` folder, right-click the `catalog.xml` file and select **Check Out**.
- b) **Open the file.**
- c) **In the public IDs, change OASIS for IXIA and remove the 1.3.**

The resulting file should be as follows:

```
<?xml version="1.0" ?>
<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog" prefer="public">
  <group>
    <public publicId="-//IXIA//ENTITIES DITA MathML Domain//EN"
uri="dtd/mathmlDomain.ent"/>
    <public publicId="-//IXIA//ELEMENTS DITA MathML Domain//EN"
uri="dtd/mathmlDomain.mod"/>
    <public publicId="-//IXIA//ENTITIES DITA Equation Domain//EN"
uri="dtd/equationDomain.ent"/>
    <public publicId="-//IXIA//ELEMENTS DITA Equation Domain//EN"
uri="dtd/equationDomain.mod"/>
  </group>
</catalog>
```

3. **Perform the previous steps in your own shell DTDs (for example, by modifying `CompanyDitabase.dtd` in your `com.company.dita.dtd` plugin).**

## 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**.

## Additional recommended updates

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.



# 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:

- **What's new in the Output Generator 4.3**
- **Install a new Output Generator**
- **Migrate your Output Generator 4.2 configuration**
- **Configure the Output Generator to use the DITA-OT 2.3.1 with DITA 1.2**
- **Configure the location of the DITA-OT catalog**
- **Run the integrator on the DITA-OT**
- **Restart the Output Generator and test your scenarios.**

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

**Note:** The Output Generator 4.3 was revamped to integrate DITA-OT 2.3.1 and DITA 1.3 and to ease the integration of another DITA-OT version. You cannot simply upgrade your existing 4.2 Output Generator. You must install a new one and import your transformation scenarios.

## What's new in the Output Generator 4.3

---

The following features are new to version 4.3 of the Output Generator:

- The Output Generator now supports version 2.3.1 of the DITA-OT, which includes the DITA 1.3 DTDs.
- The Output Generator now packages two versions of the OT:
  - 1.8.5 ("full easy install" version)
  - 2.3.1 ("zip package" version)
- By default, DITA-OT 2.3.1 is used to generate the output. You can overwrite this behavior by setting the `outgen.ot.version` property in two places:
  1. In the new `outgen-init.properties` file. This defines the default DITA-OT used to generate output for all your transformation scenarios.
  2. In the conductor file, at the top of the file. This defines the DITA-OT used for all the targets in the conductor file and overwrites any other value defined for the `outgen.ot.version` property.
- Documentation about the conductors and properties was moved to directory `%OutputGenDir%\data\conductor-doc`.
- It is no longer necessary to specify the `outgen.ot.release.family` property. This property will be deprecated.

## 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 4.2 configuration

---

After you have installed and tested version 4.3 of the Output Generator, you need to migrate your 4.2 configuration.

1. **Copy the contents of your 4.2 `%OutputGenDir%/conf/client/` directory to the 4.3 directory.**

For example, copy the contents of the following directory:

```
C:\xiasoft\OutputGenerators\Prod_4.2\conf\client\
```

to the following directory:

`C:\xilinx\OutputGenerators\Prod_4.3\conf\client\`

**2. Copy the contents of your 4.2 `%OutputGenDir%/data/resources/client/` directory to the 4.3 directory.**

For example, copy the contents of the following directory:

`C:\xilinx\OutputGenerators\Prod_4.2\data\resources\client\`

to the following directory:

`C:\xilinx\OutputGenerators\Prod_4.3\data\resources\client\`

**3. Copy the contents of your 4.2 `%OutputGenDir%/libs/client/` directory to the 4.3 directory.**

For example, copy the contents of the following directory:

`C:\xilinx\OutputGenerators\Prod_4.2\libs\client\`

to the following directory:

`C:\xilinx\OutputGenerators\Prod_4.3\libs\client\`

**4. Copy all your custom transformation scenarios (e.g., `conductor-acme.xml`) from the 4.2 `%OutputGenDir%/data/` directory to the 4.3 directory.**

**5. Copy all the `*-client.xml` files from the 4.2 `%OutputGenDir%/data/` directory to the 4.3 directory.**

For example, copy the following files:

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

**6. Copy all the DITA-OTs that you are using from the 4.2 `%OutputGenDir%/data/` directory to the 4.3 directory.**

For example, if you are using the 1.8.5 and 2.1.1 OTs, copy the following directories:

- `C:\xilinx\OutputGenerators\Prod_4.2\data\DITA-OT1.8.5`
- `C:\xilinx\OutputGenerators\Prod_4.2\data\dita-ot-2.1.1`

To the following directory:

- `C:\xilinx\OutputGenerators\Prod_4.3\data\`

**7. When requested if you want to replace existing files, say Yes.**

**8. Integrate the Subject Scheme DTD into the *IxiaMap.dtd* for all your DITA-OTs:**

a) **Open the `%OutputGenDir%/data/%DITA-OT%/plugins/com.ixiasoft.dita.dtd/dtd/IxiaMap.dtd` file with a text editor.**

b) **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;
```

c) **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;
```

d) **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;
   ..."
```

e) **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;
   ..."
```

f) **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;
```

g) **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;
```

h) **Save and close the file.**

i) **Repeat this step for all the DITA-OTs that you are using (e.g., DITA-OT 1.8.5, DITA-OT 2.1.1, etc.).**

9. **Integrate the Subject Scheme DTD into the *IxiaMap-releasemanagement.dtd* for all your DITA-OTs:**

**Note:** Perform this step even if you are not using DRM.

a) **Open the `%OutputGenDir%/data/%DITA-OT%/plugins/com.ixiasoft.dita.dtd/dtd/IxiaMap-releasemanagement.dtd` file with a text editor.**

b) **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;
```

c) **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;
```

d) **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;
           ...
```

e) **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;
           ...
```

f) **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;
```

g) **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;
```

h) **Save and close the file.**

- i) **Repeat this step for all the DITA-OTs that you are using (e.g., DITA-OT 1.8.5, DITA-OT 2.1.1, etc.).**

## 10. 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.

## 11. For all the DITA-OTs, add required DTD files:

- a) **Go to the**  
`%OutputGenDir%/data/%DITA-OT%/plugins/com.ixiasoft.dita.dtd/dtd/releasemanagement`  
**directory.**
- b) **Copy the files from the**  
`%UpgradePackageDir%/output-gen-upgrade/releasemanagement` **directory**  
**to the** *releasemanagement* **directory.**

This step will copy the following files:

- *container.mod*
- *product.mod*
- *version-tag.dtd*
- *version-tag.mod*

**Note:** When requested if you want to replace existing files, say **Yes**.

- c) **Repeat this step for all the DITA-OTs that you are using (e.g., DITA-OT 1.8.5, DITA-OT 2.1.1, etc.).**

## 12. For all DITA-OTs, replace incorrect catalog files for the SVG DTDs:

- a) **Copy the** *catalog.xml* **file from the**  
`%UpgradePackageDir%/output-gen-upgrade/org.w3c.svg1.0` **directory to the**  
`%OutputGenDir%/data/%DITA-OT%/plugins/org.w3c.svg1.0` **directory.**
- b) **Copy the** *catalog.xml* **file from the**  
`%UpgradePackageDir%/output-gen-upgrade/org.w3c.svg1.1` **directory to the**  
`%OutputGenDir%/data/%DITA-OT%/plugins/org.w3c.svg1.1` **directory.**
- c) **Repeat this step for all the DITA-OTs that you are using (e.g., DITA-OT 1.8.5, DITA-OT 2.1.1, etc.).**

## 13. By default, DITA-OT 2.3.1 is used to generate the output. To use a different version by default:

- a) **Open the `%OutputGenDir%/data/outgen-init.properties.orig` and save it as `outgen-init.properties`.**
- b) **Set the `outgen.ot.version` property to the version of the DITA-OT that you want to use.**

For example:

```
Property file for the ANT scripting
# Any property defined in this file will have priority over any other
# property defined in any ANT script file, except for the one defined in the
# wrapper.conf in the section "Java Additional Parameters".

outgen.ot.version=1.8.5

#The outgen.ot.release.family property is now optional
#outgen.ot.release.family=2.x
```

- c) **Save and close the file.**

## Configure the Output Generator to use the DITA-OT 2.3.1 with DITA 1.2

---

In DITA CMS 4.3, the Output Generator supports version 2.3.1 of the DITA-OT, which includes the DITA 1.3 DTDs. To use DITA 1.2 with the DITA-OT 2.3.1, you must update your DTDs in the Output Generator.

**Note:** If you are still using `ixiaditabase.dtd` and not your own shell plugin to generate the output, IXIASOFT does not recommend using the DITA-OT 2.3.1.

### 1. Copy your plugin to the DITA-OT 2.3.1 directory.

For example, assuming that your plugin is called `com.company.dita.dtd` and you are currently using DITA-OT 1.8.5, move the following directory:

```
C:\ixiasoft\OutputGenerators\Prod_4.3\data\DITA-OT1.8.5\plugins\com.company.dita.dtd
```

To the following location:

```
C:\ixiasoft\OutputGenerators\Prod_4.3\data\dita-ot-2.3.1\plugins\com.company.dita.dtd
```

### 2. Change the public IDs for the MathML integration to point to the IXIASOFT integration of the MathML DTDs in your `CompanyDitabase.dtd` file:

- a) **Open the `%OutputGenDir%/data/dita-ot-2.3.1/plugins/com.company.dita.dtd/dtd/CompanyDitabase.dtd` file with a text editor.**
- b) **Look for the first `MathML Integration` section.**

For example:

```
<!-- ===== MathML Integration ===== -->
<!--
    Un-comment the following block to enable
    MathML elements directly inside topics    -->

<!ENTITY % mathml-d-dec
    PUBLIC "-//OASIS//ENTITIES DITA 1.3 MathML Domain//EN"
        "../org.oasis-open.dita.mathml.doctypes/dtd/mathmlDomain.ent"
>%mathml-d-dec;

<!ENTITY % equation-d-dec
    PUBLIC "-//OASIS//ENTITIES DITA 1.3 Equation Domain//EN"
        "../org.oasis-open.dita.mathml.doctypes/dtd/equationDomain.ent"
>%equation-d-dec;
```

**Note:** This section may be commented out if you are not using MathML in your deployment.

c) **Change OASIS to IXIA in the public ID and remove the 1.3.**

For example:

```
<!-- ===== MathML Integration ===== -->
<!--
    Un-comment the following block to enable
    MathML elements directly inside topics    -->

<!ENTITY % mathml-d-dec
    PUBLIC "-//IXIA//ENTITIES DITA MathML Domain//EN"
        "../org.oasis-open.dita.mathml.doctypes/dtd/mathmlDomain.ent"
>%mathml-d-dec;

<!ENTITY % equation-d-dec
    PUBLIC "-//IXIA//ENTITIES DITA Equation Domain//EN"
        "../org.oasis-open.dita.mathml.doctypes/dtd/equationDomain.ent"
>%equation-d-dec;
```

d) **Look for the second MathML Integration section.**

For example:

```
<!-- ===== MathML Integration ===== -->
<!--
    Un-comment the following block to enable
    MathML elements directly inside topics    -->

<!ENTITY % mathml-d-def
    PUBLIC "-//OASIS//ELEMENTS DITA 1.3 MathML Domain//EN"
        "../org.oasis-open.dita.mathml.doctypes/dtd/mathmlDomain.mod"
>%mathml-d-def;

<!ENTITY % equation-d-def
    PUBLIC "-//OASIS//ELEMENTS DITA 1.3 Equation Domain//EN"
        "../org.oasis-open.dita.mathml.doctypes/dtd/equationDomain.mod"
>%equation-d-def;
```

**Note:** This section may be commented out if you are not using MathML in your deployment.

e) **Change OASIS to IXIA in the public ID and remove the 1.3.**

For example:

```
<!-- ===== MathML Integration ===== -->
<!--
                    Un-comment the following block to enable
                    MathML elements directly inside topics      -->

<!--
<!ENTITY % mathml-d-def
    PUBLIC "-//IXIA//ELEMENTS DITA MathML Domain//EN"
        "../org.oasis-open.dita.mathml.doctypes/dtd/mathmlDomain.mod"
>%mathml-d-def;

<!ENTITY % equation-d-def
    PUBLIC "-//IXIA//ELEMENTS DITA Equation Domain//EN"
        "../org.oasis-open.dita.mathml.doctypes/dtd/equationDomain.mod"
>%equation-d-def;
```

f) **Save and close the file.**

### 3. Copy the IXIASOFT integration of the MathML plugin to the DITA-OT 2.3.1 directory.

For example, if you are using DIT-OT 1.8.5, copy the following directory:

*C:\xiasoft\OutputGenerators\Prod\_4.3\data\DITA-OT1.8.5\plugins\org.oasis-open.dita.mathml.doctypes*

To the following location:

*C:\xiasoft\OutputGenerators\Prod\_4.3\data\dita-ot-2.3.1\plugins\*

### 4. Change the catalog for the MathML plugins to reflect the new public IDs:

a) **Open the**

*%OutputGenDir%/data/dita-ot-2.3.1/plugins/org.oasis-open.dita.mathml.doctypes/catalog.xml*  
**file with a text editor.**

b) **In the public IDs, change OASIS for IXIA and remove the 1.3.**

The resulting file should be as follows:

```
<?xml version="1.0" ?>

<catalog xmlns="urn:oasis:names:tc:entity:xmlns:xml:catalog" prefer="public">

  <group>
    <public publicId="-//IXIA//ENTITIES DITA MathML Domain//EN"
      uri="dtd/mathmlDomain.ent"/>
    <public publicId="-//IXIA//ELEMENTS DITA MathML Domain//EN"
      uri="dtd/mathmlDomain.mod"/>
    <public publicId="-//IXIA//ENTITIES DITA Equation Domain//EN"
      uri="dtd/equationDomain.ent"/>
    <public publicId="-//IXIA//ELEMENTS DITA Equation Domain//EN"
      uri="dtd/equationDomain.mod"/>
  </group>

</catalog>
```

## 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 including 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:	<ul style="list-style-type: none"> <li>• On Windows: In the <i>%OutputGenDir%/data/</i> directory, double-click the <i>_DITA-OT-1.8.5-integrator.bat</i> file.</li> <li>• On Linux:             <ol style="list-style-type: none"> <li>1. In the <i>%OutputGenDir%/data/%OT_Dir%/</i> directory, run <code>startcmd.sh</code>.</li> <li>2. At the command prompt, enter:                 <pre>ant -f integrator.xml</pre> </li> </ol> </li> </ul> <p>The integration build runs and you should soon see BUILD SUCCESSFUL.</p>
DITA-OT 2.x family:	<ul style="list-style-type: none"> <li>• On Windows: In the <i>%OutputGenDir%/data/</i> directory, double-click the <i>_DITA-OT-2.3.1-integrator.bat</i> file.</li> <li>• On Linux: In the <i>%OutputGenDir%/data/%OT_Dir%/bin</i> directory, enter the following command:             <pre>./ant -f ../integrator.xml (Linux)</pre> </li> </ul> <p>The integration build runs and you should soon see BUILD SUCCESSFUL.</p>

## 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:

- **Uninstall the Scheduler 4.2 service**
- **Download and extract the Scheduler server files**
- **Install and start the Scheduler service (Windows)**
- **Install and start the Scheduler service (Linux)**
- **Replace the daily\_reminder.xsl template in Scheduler**

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

## Uninstall the Scheduler 4.2 service

---

To uninstall the Scheduler service:

**1. Stop the Scheduler 4.2 service.**

**2. Uninstall the Scheduler 4.2 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...
```

## Replace the `daily_reminder.xsl` template in Scheduler

---

The `daily_reminder.xsl` template was updated to include the Comments provided when a document was assigned to a user.

**Note:** This procedure assumes that you already imported the Scheduler templates. If that is not the case, see the *Scheduler Installation Guide*.

To replace the `daily_reminder.xsl` template:

1. **Open a TEXTML Server Console Java.**

**2. Connect to the server and Content Store to upgrade.****3. Make a backup of the *daily\_reminder.xsl* file:**

This ensures that you keep any customization you made to this template.

- a) **Expand the Content Store's Repository node and browse to the */system/scheduler/templates* folder.**
- b) **Right-click the *daily\_reminder.xsl* file and select Rename.**
- c) **In the New Document Name box, enter *daily\_reminder.cms42.xsl*.**
- d) **Click OK.**

**4. Navigate to the *Repository\system\scheduler\templates* directory.****5. Right-click the *templates* directory and select Insert Documents...**

The **Insert Documents Dialog** window appears.

**6. Click Add File and browse to the *%SchedulerDir%\conf\templates* directory.**

For example, *C:\ixiasoft\Scheduler\conf\templates*.

**7. Select the *daily\_reminder.xsl* file and click OK.****8. In the Set Options:**

- Select **Add**.
- Select **Replace**.
- Clear **Indexable**.

**9. Click OK.**

The *daily\_reminder.xsl* file is imported.

**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**.



# 6

## Upgrade the WCR application

This procedure describes how to upgrade the Web Collaborative Reviewer (WCR) application.

**Note:** Carry out this procedure at a time of low network activity, since the update will cause users to lose their connection to the WCR application.

**To update the WCR application:**

- 1. Open a connection to the Tomcat server where the existing WCR application is installed.**
- 2. Delete the old wcr folder from `%TomcatDir%/webapps/`.**
- 3. Copy the wcr.war file into the `%TomcatDir%/webapps/` directory.**

Tomcat automatically expands the WAR file and installs the application in the `wcr` directory.

- 4. Wait a few seconds while the WCR is installed.**

**Note:** You do not need to restart WCR or Tomcat after an update, because the existing WCR configuration file in `%TomcatDir%/conf` is not overwritten during the update.



# 7

## Upgrade the Web Author

### Topics:

- **Download and extract the Web Author package**
- **Upgrade the CMS Application Server**
- **Upgrade the Web Author Application**
- **Upgrade the online help**
- **Test your DITA CMS Web installation**

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

It includes the following steps:

1. Download the Web Author package
2. Upgrade the CMS Application Server
3. Upgrade the Web Author Application
4. Upgrade the online help

**Make sure that both the CMS Application Server and the Web Author Application components provided by IXIASOFT are upgraded to the same software version.**

## Download and extract the Web Author package

---

The files required to install the Web Author are packaged in a zip file provided by IXIASOFT.

To download the Web Author package:

**1. Go to the IXIASOFT Web Author download page:**

<http://cms.ixiasoft.com/downloads/webauthor/>

**2. Click the DITA CMS version; for example, 4 . 3.**

**3. Click the latest build available for the DITA CMS; for example, 4 . 4 . x.**

**4. Select *WebAuthor-<version>.zip*, where *<version>* is the latest version of the Web Author (for example, *WebAuthor-4.4.X.zip*).**

**5. Extract the zip file.**

This package contains three directories, as follows:

- `contentstore`: Contains configuration that must be imported in the DITA CMS Content Store.
- `glassfish`: Contains files used during the installation of the GlassFish Server and the CMS Application Server.
- `webauthor`: Contains files used during the installation of the Tomcat Server and the Web Author Application.

You will need these files when installing and configuring the Web Author.

## Upgrade the CMS Application Server

---

To update the CMS Application Server, you upload the *glassfish/cmsappserver.ear* file and deploy it again.

**1. Confirm that the CMS Application Server is connecting to the correct Content Store:**

- a) Open the *%GlassFishDir%/domains/cmsappserver/config/CMSAppServer.config* file with a text editor.**
- b) Make sure that the `<textml-connection>` element points to the correct Content Store. If that is not the case, update the attributes and save the file.**

**2. In a browser window, enter the following URL:**

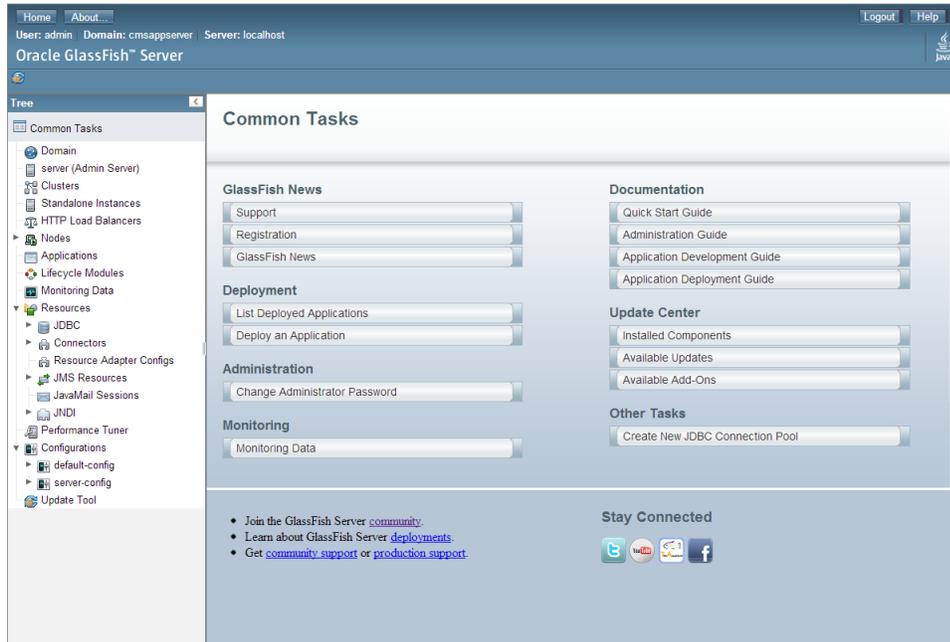
```
http://<glassfishserver>:4848
```

Where *<glassfishserver>* is the address of the machine where the GlassFish Server is installed.

The **Oracle GlassFish Server Administration Console** login page is displayed.

3. Enter the User Name and Password specified when you installed GlassFish (for example, **admin** and **admin**).
4. Click **Login**.

The **GlassFish Server Administration Console** is displayed:



5. In the **Common Tasks** panel, select **Applications**.

The **Applications** window is displayed.

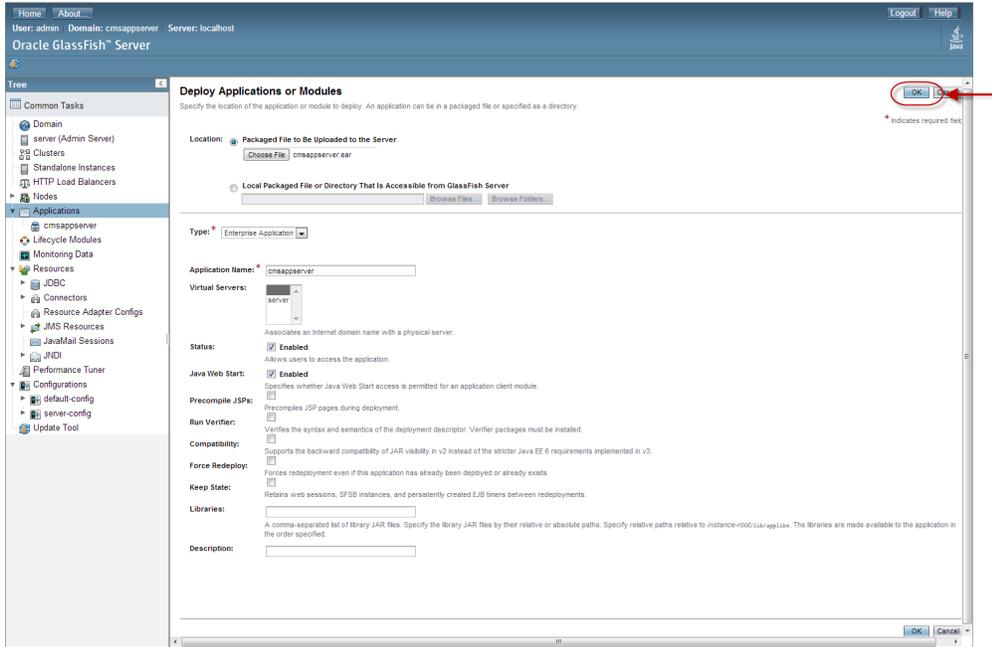
6. Click the **Redeploy** action for the **cmsappserver** application.

The **Redeploy Applications or Modules** window is displayed.

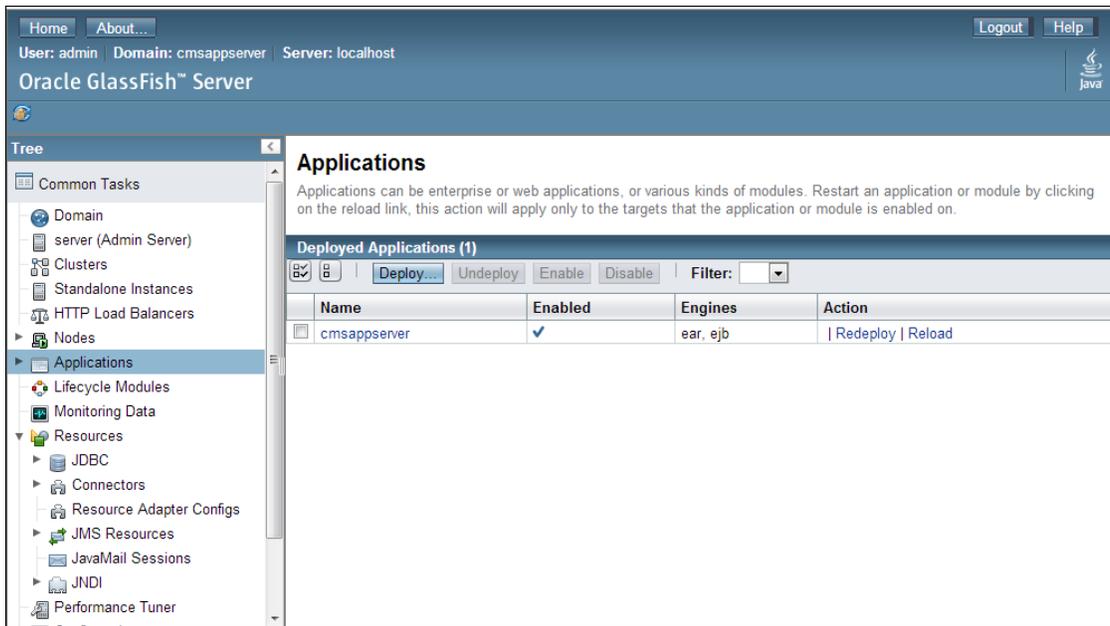
7. Click **Choose File**, browse to the directory where you saved the *glassfish/cmsappserver.ear* file provided in the **Web Author** package, and click **Open**.

The **Deploy Applications or Modules** window is displayed again.

8. Keep all options as is, as shown below, and click **OK**.



The CMS Application Server is upgraded and enabled, as shown in the following diagram:



## Upgrade the Web Author Application

To update the Web Author Application, you copy the *webauthor/webauthor.war* file and deploy it again.

### 1. Stop the Tomcat service.

2. Delete the `%TomcatDir%/webapps/webauthor` directory.
3. Delete the `%TomcatDir%/webapps/webauthor.war` file.
4. Copy the `webauthor/webauthor.war` file provided in the IXIASOFT Web Author package to the following directory:

```
%TomcatDir%/webapps
```

Where `%TomcatDir%` is the location of the Tomcat installation. For example:

```
C:\Program Files\Apache Software Foundation\Tomcat 6.0\webapps (Windows)
/opt/tomcat6/webapps (Linux)
```

5. Restart the Tomcat service.

## Upgrade the online help

---

To get the latest documentation, you need to upgrade the online help.

1. Start the DITA CMS and open the TEXTML Administration perspective.
2. Connect to your server and doctype.
3. Expand the doctype's repository node and navigate to the `system/cms.webauthor/help` folder.
4. Right-click the `help` folder and select **Insert Documents**.  
The **Insert Documents Dialog** window appears.
5. Click **Add File** and browse to the `contentstore/cms.webauthor/help` directory of the extracted Web Author package provided by IXIASOFT.
6. Select the `cms.webauthor.hlp` file and click **Open**.
7. In the **Set Options** area, select **Add and Replace**.
8. Click **OK**.

The file is added to the DITA CMS configuration.

## Test your DITA CMS Web installation

---

The final step is to test your installation.

To test your installation:

1. Enter the following URL in a web browser:

```
http://<tomcatserver:port>/<DITA_CMS_Web_Application>/
```

Where:

- `<tomcatserver:port>` is the address of the Tomcat server
- `<DITA_CMS_Web_Application>` is the name of the DITA CMS Web Application `.war` file (webauthorx by default)

For example:

```
http://acmetomcatserver:8090/webauthorx
```

The **DITA CMS Web** login page is displayed, indicating that the application was installed correctly.

For example:

