

DITACMS

DITA CMS Eclipse Client Installation Guide

Table of contents

Copyright notice

New in this documentation release

Packaging and deploying the DITA CMS Eclipse Client

Client installation overview	9
Understanding the plug-in deployment options	10
XML Editor requirements	13
Create an Eclipse update site	14
Create a bookmarks file	16
Plug-in dependencies	19
Prepare the initial Eclipse SDK package	20
Download the Eclipse SDK	22
Package the Java Runtime Environment (JRE) version 1.7	23
Edit the eclipse.ini file	25
(Optional) Enable Kerberos security	27
(Optional) Include the DITA CMS plug-ins and Oxygen XML Editor/Author as dropins	29
(Optional) Deploy the plug-ins using an update site	31
(Optional) Configure the initial working environment for DITA CMS Eclipse clients	32

(Optional) Configure the DITA CMS client in a remote solution	34
Finalize the package	36
Deploy the package	36
Troubleshooting	37
Updating the DITA CMS Eclipse Client	39
Updating the DITA CMS plug-ins when using dropins	39
Updating the DITA CMS plug-ins using an update site	40
Appendix A: List of DITA CMS plug-ins	
Appendix B: DITA CMS Eclipse Client software requirements	
Appendix C: XMetaL XMAX requirements	

Copyright notice

©2011-2016 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.

New in this documentation release

DITA CMS 4.2.33 Updates

- A configuration file is now available to preconfigure the DITA CMS Eclipse clients. This will allow users launch their DITA CMS Eclipse clients configured with the correct connection information and DITA perspective. See **(Optional) Configure the initial working environment for DITA CMS Eclipse clients** on page 32.

DITA CMS 4.2 Updates

- A new parameter (`-Ddita.cms.mode=%mode%`) is now required in the *eclipse.ini* file. See **Edit the eclipse.ini file** on page 25.
- The DITA CMS 4.2 requires version 17.0 and up of the oXygen XML Editor.

Packaging and deploying the DITA CMS Eclipse Client

This document describes how to install the DITA CMS Eclipse Client.

The DITA CMS Eclipse Client is a set of plug-ins that run on top of the Eclipse SDK framework. This document provides the procedure for creating the initial Eclipse SDK package, which is customized for the DITA CMS. It also describes the different options available for deploying the DITA CMS plug-ins and provides the procedures for each option.

Audience

This section is intended for system administrators who package and deploy the DITA CMS Eclipse Client on user workstations and in remote access solutions.

Client installation overview

To install the DITA CMS Eclipse Client, you prepare the initial Eclipse SDK package and then install the plug-ins.

- **Prepare the initial Eclipse SDK package.** This is done by downloading the standard Eclipse SDK and customizing it for DITA CMS.
- **Install the DITA CMS plug-ins.** There are two options for installing the DITA CMS Eclipse plug-ins:
 - Through **dropins**: When Eclipse starts, it checks the contents of its *dropins* directory and loads the newest plug-ins available. To install a new plug-in or a new version of an existing plug-in, you can simply copy it to the *dropins* directory and it will be loaded at startup.
 - Through an **update site**: An update site is a URI (web site, FTP site, file server, etc.) that contains the plug-ins available for installation. Users configure their Eclipse to connect to this URI and check if a new plug-in or a new version of an existing plug-in is available. If it is, then users can install it.

You can install the DITA CMS Eclipse Client directly on user workstations or in a remote access solution (for example, a Citrix or Remote Application server or virtual machines). If you install the client on workstations or virtual machines, you can use either the dropins or update site method. If you install the client on a server farm (Citrix or Remote Application server), you must use dropins.

You can configure the DITA CMS to authenticate users. The DITA CMS supports:

- **Kerberos authentication**—Kerberos is an authentication protocol, based on the concept of "tickets", that allows nodes to prove their identity to one another in a secure manner. TEXTML Server can use the Kerberos protocol to authenticate users and provide a secure connection between itself and the DITA CMS Eclipse Client. The DITA CMS supports Kerberos authentication on Windows only (for both server and client, using Active Directory).
- **Local authentication**—In this scheme, TEXTML Server uses the local operating system (OS) to authenticate users. The local OS validates the username and password supplied by the user against the authentication method configured on the local OS. If a centralized user database is used (Active Directory, NIS, etc.), the local OS configuration is used to reach the centralized database.

The authentication scheme that you select will impact the initial Eclipse SDK configuration.

Understanding the plug-in deployment options

The DITA CMS Eclipse Client is packaged as plug-ins that provide a set of features that can be easily installed. There are two options for deploying the DITA CMS Eclipse plug-ins: dropins and update sites.

Using dropins

Eclipse includes a directory called *dropins*. When Eclipse starts, it checks the contents of its *dropins* directory and loads the newest plug-ins available.

To install a new plug-in or a new version of an existing plug-in, you simply copy it to the *dropins* directory and it will be loaded at startup. When you create the initial Eclipse package, you can also include the DITA CMS plug-in as dropins to simplify the initial installation.

The DITA CMS plug-ins are available as dropins from the IXIASOFT download site. For dropins, there is one plug-in per zip file, so each DITA CMS plug-in is packaged in its own zip file, as follows:

```
com.ixiasoft.<plug-in-name>-<plug-in-version>.zip
```

For example:

```
com.ixiasoft.dita-cms-admin-4.2.22.zip
```

To install these plug-ins using the dropins option, you extract the zip file to the Eclipse *dropins* directory. When extracted, each plug-in contains two directories:

- *features*

- `\plugins`

For example:

```
\eclipse\dropins\com.ixiasoft.dita-cms-admin-4.2.22\features\  
\eclipse\dropins\com.ixiasoft.dita-cms-admin-4.2.22\plugins\  

```

You repeat this step for every plug-in to install. This procedure is described in detail later on.

Using an update site

An update site is a URI (web site, FTP site, file server, etc.) that contains the latest DITA CMS plug-ins. Users configure their Eclipse to connect to this URI and check if a new plug-in or a new version of an existing plug-in is available. If it is, then users can install it.

Note: While all URI types are supported, IXIASOFT recommends using a website or file server. This document describes how to set up an update site using a web server. For more information concerning another method, please contact IXIASOFT Support.

To install the plug-ins using an update site, you:

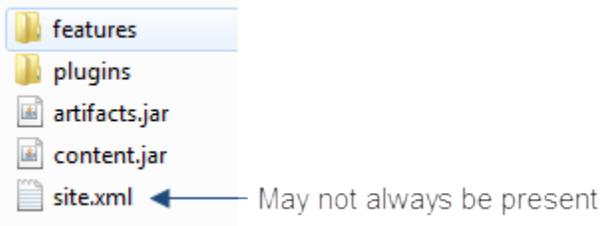
- Create your update site
- Make the plug-ins available from the update site

This procedure is described in detail later on. The DITA CMS plug-ins are available from the IXIASOFT download site. For update sites, there is one or more plug-ins per zip file, so plug-ins are packaged in an *update site zip file*. The following table describes how the IXIASOFT plug-ins are set up on the download site:

Update zip file	Plug-ins
<i>dita-cms-<version>.zip</i>	<ul style="list-style-type: none"> • DITA Content Management System • Collaborative Reviewer • oXygen XML Extension (Editor) • oXygen XML Extension (Author) • XMetaL XMAX Extension • HyperSTE • Dynamic Release Management (DRM) module <p>Note: The DRM is an optional module that requires complex configuration updates. IXIASOFT Services must have already converted your system configuration and</p>

Update zip file	Plug-ins
	data before you can install this plug-in. Do not install this module without IXIASOFT Services help, otherwise the DITA CMS installation may corrupt your data.
<i>dita-doc-<version>.zip</i>	DITA 1.2 Specification
<i>dita-cms-admin-<version>.zip</i>	DITA CMS Administration
<i>textml-admin-<version>.zip</i>	TEXTML Server Administration

Each zip file has the following structure:



The *site.xml* file of each zip file contains the URLs of the plug-ins available on the site. This file may not always be present.

Note: The DITA CMS Eclipse Client cannot be installed on server farms using update sites. Dropins must be used for server farms.

Using a hybrid solution

For workstations and virtual machines, you can deploy the DITA CMS Eclipse Client plug-ins using a *hybrid solution*, which is a combination of dropins and update site.

It is up to you to determine the best solution for your deployment. If your writing team is small, using dropins only might be the simplest solution. For larger deployments, you might want to use update sites only or a hybrid solution.

You might, for example, deploy the plug-ins as follows:

- Scenario 1:
 - Update sites: DITA CMS client and DITA Specification plug-ins are delivered using an update site
 - Dropins: DITA CMS and TEXTML Server Administration plug-ins are delivered using dropins

- Scenario 2:
 - Update sites: The DITA CMS plug-ins are provided using an update site
 - Dropins: The Oxygen XML Editor/Author software is included in the initial installation package, since the files are very large
- Scenario 3:
 - Update sites: DITA CMS client and DITA Specification plug-ins are delivered using an update site
 - Dropins: The Oxygen XML Editor/Author software and the DITA CMS and TEXTML Server Administration plug-ins are included in the initial installation package

You need to determine the best deployment method based on your requirements.

Note: If you install a plug-in using the dropins method, you cannot use an update site to update it later on. A plug-in must be installed and updated using the same method.

XML Editor requirements

To edit DITA files with the DITA CMS, you need an XML editor. You can use the default XML editor provided with Eclipse, but for an improved user experience, DITA CMS provides extensions for two popular XML editors, oXygen XML and XMetaL XMAX.

These extensions add the DITA CMS contextual menus to the XML editors. The DITA CMS installation does not include oXygen XML and XMetaL XMAX editors. You need to buy one of these editors and install it on the user workstations. Note that:

- If you are using oXygen XML, you can install it as a plug-in with the other DITA CMS plug-ins. This is described later on.

Note: Version 17.0 of oXygen XML is required for DITA CMS 4.2.

- If you are using XMetaL XMAX Editor, see <http://xmetal.com/content-xmetal-xmax/> for the installation instructions. You **must** install XMetaL XMAX 8.0 before you install the DITA CMS extension. For more information about the installation, see **XMetaL XMAX requirements** on page 47.

Note: Version 8.0 of XMetaL XMAX is required for DITA CMS 4.2.

Create an Eclipse update site

If you plan to deploy plug-ins using an update site, you need to determine which update site to use.

IXIASOFT provides update sites that you can use to deploy the DITA CMS plug-ins to your users. However, you will not be able to control the plug-ins that will be pushed to your users. As soon as IXIASOFT releases an update to a plug-in, it will be available for your users. Once an update has been installed on a machine, it is not possible to go back to a previous version.

If you want to control the updates that are delivered to your users, you should create your own update sites. This ensures that you can select the DITA CMS plug-ins to distribute and make available to your users. It is also good practice to create two sets of update sites:

- Test update sites: You use these update sites to download and test the updated plug-ins.
- Production update sites: They contain the plug-ins that you have tested and approved for deployment.

To create an update site for the DITA CMS plug-ins:

Note: This procedure assumes that you know how to create a web server. Contact your system administrator for more information if that is not the case.

1. Create a web server.

You can use any web server, such as Apache, IIS, Tomcat, etc. The web server only needs to serve static files (no scripting is required). The update site can even be created with the Tomcat server that is used for the Web Collaborative Reviewer.

The examples used in this procedure assume that the web server created has the following properties:

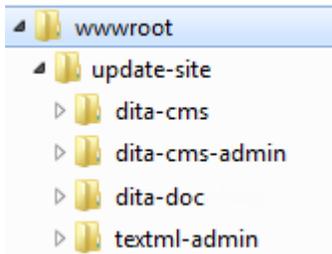
- The root of the web site is *C:\wwwroot*
- The name of the server is *update.acme.com*

2. In the web site root directory, create two directories to store your test and production update sites; for example, *test-update-site* and *update-site*.

3. In each update site directory, create a directory for each of the following plug-ins:

- *dita-cms*
- *dita-cms-admin*
- *dita-doc*
- *textml-admin*

For example:



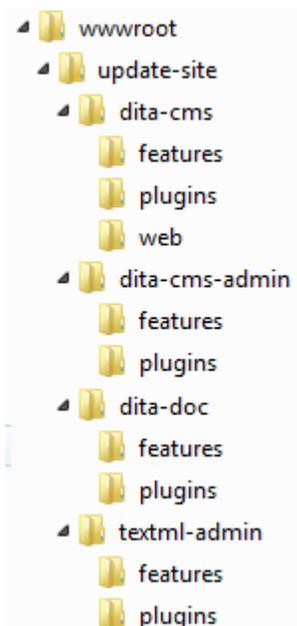
4. Download the plug-ins from the IXIASOFT download site at the following URL:

<http://cms.ixiasoft.com/downloads/plugins/4.2/>

5. Unzip the content of each update site zip file to the appropriate folder; for example:

Update site zip file	Directory
<i>dita-cms-<version>.zip</i>	<i>C:\wwwroot\update-site\dita-cms</i>
<i>dita-doc-<version>.zip</i>	<i>C:\wwwroot\update-site\dita-doc</i>
<i>dita-cms-admin-<version>.zip</i>	<i>C:\wwwroot\update-site\dita-cms-admin</i>
<i>textml-admin-<version>.zip</i>	<i>C:\wwwroot\update-site\textml-admin</i>

For example:



6. Test the plug-ins. When ready, repeat the procedure to install the plug-ins on the production update site.

Create a bookmarks file

Once you have set up your update sites, you must provide your users with the list of URIs that they need to connect to.

Instead of asking your users to enter many URIs manually, you can create bookmarks files. A *bookmarks file* is an XML file that contains a list of update sites; users simply import this list into Eclipse.

The bookmarks file can contain the URIs of the DITA CMS plug-ins as well as public URIs (for example, for oXygen XML Editor and Author plug-in updates). IXIASOFT recommends that you create two bookmarks files: one for CMS administrators and one for regular DITA CMS users. This approach ensures that the administrator plug-ins are distributed to CMS administrators only and are not installed by regular users.

The following table lists the URIs that you should include in each bookmarks file:

Bookmarks file	Update site name	Used by
<i>bookmarks.xml</i>	<ul style="list-style-type: none"> <i>dita-cms</i> <i>dita-doc</i> <p>This file may also contain the following public URIs:</p> <ul style="list-style-type: none"> Eclipse BIRT plug-in: Required if you want to add business intelligence and reporting tools (BIRT) to your Eclipse environment oXygen XML Editor/Author update plug-ins 	<ul style="list-style-type: none"> Regular users CMS administrators
<i>bookmarks_admin.xml</i>	<ul style="list-style-type: none"> <i>dita-cms-admin</i> <i>textml-admin</i> 	<ul style="list-style-type: none"> CMS administrators only

Note: You could also create a bookmarks file for the test update sites, if appropriate.

Bookmarks file structure

The bookmarks file is an XML file with the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookmarks>
  <site url="site_url" selected="true" name="site_name"/>
</bookmarks>
```

Each URI is stored in a `<site>` element, where:

- *site_url* is the URI of the update site; for example:
http://update.acme.com/dita-cms-admin/site.xml
- *site_name* is the name of the update site as it will appear in Eclipse; for example: **DITA CMS Administration**

To create the bookmarks files:

1. Create a bookmarks file to configure the update sites for a regular DITA CMS user.

Specify the URIs for your update site; for example, for the *update.acme.com* update site, the *bookmarks.xml* file would look as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookmarks>
  <site
    url="http://update.acme.com/dita-doc/site.xml"
    selected="true"
    name="DITA 1.2 Specification"/>
  <site
    url="http://update.acme.com/dita-cms/site.xml"
    selected="true"
    name="IXIASOFT DITA-CMS 4.2"/>
</bookmarks>
```

2. Create a bookmarks file to configure the update sites that are specific to the DITA CMS administrator.

Specify the URIs for your update site; for example, for the *update.acme.com* update site, the *bookmarks_admin.xml* file would look as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookmarks>
  <site
    url="http://update.acme.com/dita-cms-admin/site.xml"
    selected="true"
    name="DITA CMS Administration"/>
  <site
    url="http://update.acme.com/textml-admin"
    selected="true"
    name="TEXTML Server Administration"/>
</bookmarks>
```

3. Add the URLs for the BIRT and oXygen XML update sites to the *bookmarks.xml* file, as follows:

Note: Only add the URLs that are necessary for your deployment.

url	name
http://download.eclipse.org/birt/update-site/2.5	Eclipse BIRT
http://oxygenxml.com/InstData/Editor/Eclipse/site.xml	oXygen XML Editor

url	name
http://oxygenxml.com/InstData/Author/Eclipse/site.xml	oXygen XML Author

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<bookmarks>
  ...
  <site
    url="http://oxygenxml.com/InstData/Editor/Eclipse/site.xml"
    selected="true"
    name="oXygen XML Editor"/>
  <site
    url="http://download.eclipse.org/birt/update-site/2.5"
    selected="true"
    name="Eclipse BIRT"/>
  ...
</bookmarks>
```

Using the IXIASOFT update site

You can also use the IXIASOFT update site directly to deploy plug-ins to your users. In this case, you simply provide these URLs to your users so that they can update their DITA CMS themselves.

The following table lists the URLs for the IXIASOFT update sites:

Update site	URL
DITA CMS Administration	http://cms.ixiasoft.com/update-site-4.2/dita-cms-admin/site.xml
TEXTML Administration	http://cms.ixiasoft.com/update-site-4.2/textml-admin
DITA CMS (includes the following plug-ins):	http://cms.ixiasoft.com/update-site-4.2/dita-cms/site.xml
<ul style="list-style-type: none"> • DITA Content Management System • Collaborative Reviewer • oXygen XML Extension (Editor) • oXygen XML Extension (Author) • XMetaL XMAX Extension • HyperSTE 	

Update site	URL
<ul style="list-style-type: none"> Dynamic Release Management 	
DITA 1.2 Specification	http://cms.ixiasoft.com/update-site-4.2/dita-doc/site.xml

Plug-in dependencies

Some plug-ins have *dependencies* on other plug-ins; if these required plug-ins are not present, installation will fail.

The following table lists the plug-in dependencies. Make sure that all required plug-ins are already installed or are provided in your bookmarks files.

Plug-in	Requirements
DITA CMS Administration	<ul style="list-style-type: none"> TEXTML Server Administration <p>You can install the TEXTML Server Administration plug-in at the same time as you install the DITA CMS Administration plug-in</p>
oXygen XML Extension (Editor)	<ul style="list-style-type: none"> oXygen XML Editor (third-party plug-in) <p>You can install the oXygen XML Editor plug-in at the same time as you install the oXygen XML Extension (Editor) plug-in. Your users will also need a valid Editor license.</p>
oXygen XML Extension (Author)	<ul style="list-style-type: none"> oXygen XML Author (third-party plug-in) <p>You can install the oXygen XML Author plug-in at the same time as you install the oXygen XML Extension (Author) plug-in. Your users will also need a valid Author license.</p>
XMetaL XMAX Extension	<ul style="list-style-type: none"> XMetaL XMAX (third-party Windows application) <p>XMetaL XMAX is not installed as a plug-in; it's installed as a Windows application. You don't have to install it before you install the XMetaL XMAX Extension, but your users will not be able to edit content. Your users will also need a valid XMetaL XMAX license.</p>

Plug-in	Requirements
HyperSTE for XMetaL XMAX	<ul style="list-style-type: none"> XMetaL XMAX Extension <p>You can install the XMetaL XMAX Extension plug-in at the same time as you install the HyperSTE for XMetaL XMAX plug-in.</p>

Prepare the initial Eclipse SDK package

The DITA CMS plug-ins are installed on a version of the Eclipse SDK framework that is customized for DITA CMS.

To determine what goes into your initial package, you need to answer the following questions:

- Will Kerberos authentication be used on your Windows deployments?
- For workstations and virtual machines, what deployment option will you use for the plug-ins:
 - Dropins only, and include them in your initial package?
 - An update site only?
 - A hybrid option, where the initial package contains dropins but other plug-ins are provided with an update site?

The following table describes what you need to include into your initial package, depending on your answers:

Configuration step	Workstation/Virtual machine			Server farm
	Dropins only	Update site	Hybrid	Dropins
Download the Eclipse SDK on page 22	X	X	X	X
Package the Java Runtime Environment (JRE) version 1.7 on page 23	X	X	X	X
Edit the eclipse.ini file on page 25	X	X	X	X
(Optional) Configure the initial working environment for DITA CMS Eclipse clients on page 32	X	X	X	X

Configuration step	Workstation/Virtual machine			Server farm
	Dropins only	Update site	Hybrid	Dropins
(Optional) Include the DITA CMS plug-ins and Oxygen XML Editor/Author as dropins on page 29	X		X	X
(Optional) Deploy the plug-ins using an update site on page 31		X	X	
(Optional) Enable Kerberos security on page 27	(X)	(X)	(X)	(X)
(Optional) Configure the DITA CMS client in a remote solution on page 34				X
Finalize the package on page 36	X	X	X	X

You need to create a package for every platform supported, for example:

- Windows 32-bit
- Windows 64-bit
- Linux 32-bit
- Linux 64-bit

Also, if you plan to deploy your plug-ins using dropins, you need to create two packages per platform supported, as follows:

- A regular user package
- A CMS administrator package

This is required since the administrator plug-ins should not be available to regular users.

For example, if you plan to install the DITA CMS using dropins and your deployment supports Linux 64-bit and Windows 64-bit, you would need to create the following four packages:

- Linux 64-bit for regular users
- Linux 64-bit for CMS administrators
- Windows 64-bit for regular users
- Windows 64-bit for CMS administrators

Download the Eclipse SDK

The first step is to download version 4.3 of the Eclipse SDK.

To prepare the initial Eclipse SDK package:

1. Go to the following URL:

<http://archive.eclipse.org/eclipse/downloads/>

2. Select the "4.3.2" build name.

3. Download the Eclipse SDK for your platform.

Important: If you are using XMetaL XMAX, you must download the 32-bit version of the Eclipse SDK.

4. Create a packaging directory for each package required.

To uniquely identify the different packages that you create, IXIASOFT recommends that you use the following naming convention:

```
cms<version>_[admin_]32|x64
```

Where:

- `<version>`: is the version of the DITA CMS, for example, 42 for 4.2
- `[admin]`: indicates that this package is specific to CMS administrators. This is only required for deployments that provide plug-ins using dropins.
- `32|x64`: indicates whether this package is for the 32-bit or 64-bit version of the platform

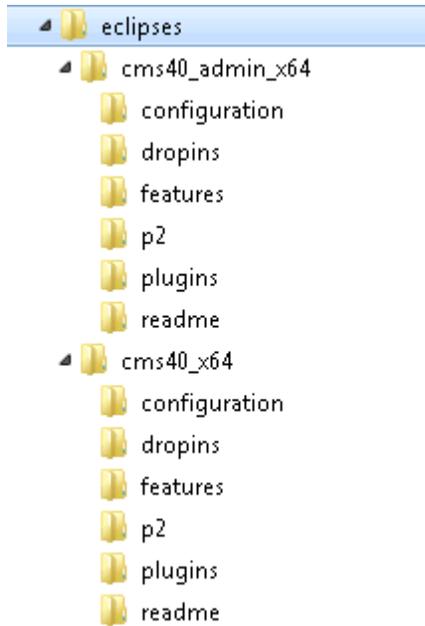
For example, for a deployment that uses dropins and supports Linux 64-bit and Windows 64-bit, you would need to create the following directories:

```
C:\eclipses\cms42_x64           <-- For regular users
C:\eclipses\cms42_admin_x64   <-- For CMS administrators
%installdir%\eclipses/cms42_x64 <-- For regular users
%installdir%\eclipses/cms42_admin_x64 <-- For CMS administrators
```

5. Extract the Eclipse SDK to a temporary directory (for example, `C:\temp_sdk`).

6. Copy the contents of the extracted *eclipse* directory (for example, `C:\temp_sdk\eclipse`) to every packaging directory.

The structure should be as follows:



Important: Do NOT start Eclipse from this directory. When you start Eclipse, it creates installation files that are specific to the computer on which it is running. If you start Eclipse from your packaging directory, these files will be included when you deploy the DITA CMS.

Package the Java Runtime Environment (JRE) version 1.7

Eclipse is a Java-based program. By default, it uses the version of the Java Runtime Environment (JRE) that is installed on the computer. Since DITA CMS requires version 1.7.0_51 and up of the JRE, the Eclipse SDK must be modified to include Java 1.7.

IXIASOFT recommends that you package the JRE version 1.7 in each packaging directory to ensure that the DITA CMS Eclipse Client uses the correct one.

For each packaging directory:

1. **If JRE 1.7 is not installed on your machine, download and install it from the following URL:**

<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>

Note: By default, the JRE is installed in the following directories:

Operating system	JRE	Default directory
Windows 32 bit	JRE 32 bit	<i>C:\Program Files\Java\jre7\</i>
Windows 64 bit	JRE 32 bit	<i>C:\Program Files (x86)\Java\jre7\</i>

Operating system	JRE	Default directory
Windows 64 bit	JRE 64 bit	<i>C:\Program Files\Java\jre7\</i>
Linux	N/A	There are no default in Linux, so take note of the directory where you install the JRE.

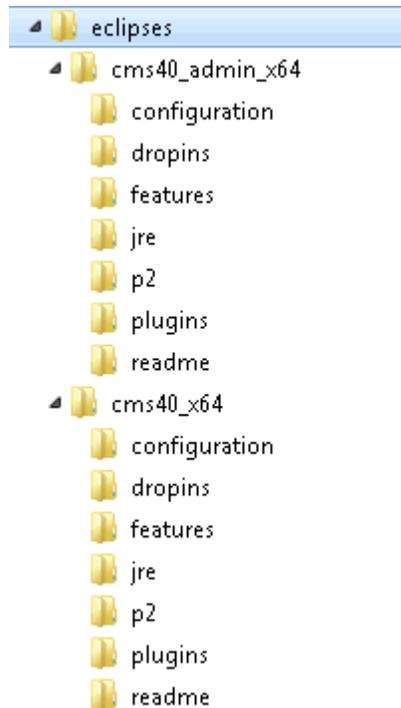
2. In each packaging directory, create a sub-directory called *jre*; for example:

```
C:\eclipses\CMS42_x64\jre
C:\eclipses\CMS42_admin_x64\jre
```

When Eclipse starts, it checks if the *jre* directory is included in its installation and, if so, it uses the Java runtime environment from this directory.

3. Copy the content of the *jre7* directory (for example, *C:\Program Files\Java\jre7*) to the *jre* directories.

The structure should now look as follows:



4. Repeat this procedure for every packaging directory.

Edit the eclipse.ini file

The default *eclipse.ini* file packaged with the Eclipse SDK must be configured for DITA CMS.

By default, the *eclipse.ini* file packaged with the Eclipse SDK looks as follows (the example below applies to Windows 64 bit):

```
eclipse.ini
-startup
plugins/org.eclipse.equinox.launcher_1.3.0.v20130327-1440.jar
--launcher.library
plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.200.v20140116-2212
-product
org.eclipse.epp.package.standard.product
--launcher.defaultAction
openFile
--launcher.XXMaxPermSize
256M
-showsplash
org.eclipse.platform
--launcher.XXMaxPermSize
256m
--launcher.defaultAction
openFile
--launcher.appendVmargs

-vmargs
-Dosgi.requiredJavaVersion=1.6
-Xms40m
-Xmx512m
```

Section 1: Platform-specific, do not modify

Section 2: Update as described below

Update Section 2 to include configuration required by the DITA CMS.

For each packaging directory:

1. Open the *eclipse.ini* file located in your packaging directory (for example, *C:\eclipses\cms42_x64\eclipse.ini*).
2. Replace the lines after the `-vmargs` line with the following:

```
-vmargs
-Xms256m
-Xmx1024m
-Dxia.level=DEBUG
-Dsun.java2d.d3d=false
-Dosgi.requiredJavaVersion=1.6
-Dosgi.bundlefile.limit=1000
-Ddita.cms.mode=%mode%
```

Where:

Field	Description
<code>-Xms256m</code>	Minimum amount of memory used by Eclipse at startup. You can modify this parameter as appropriate for your deployment.
<code>-Xmx1024m</code>	Maximum amount of memory used by Eclipse. You can modify this parameter as appropriate for your deployment.
<code>-Dixia.level=DEBUG -Dsun.java2d.d3d=false -Dosgi.requiredJavaVersion=1.6 -Dosgi.bundlefile.limit=1000</code>	Configuration parameters required by the DITA CMS. Do not modify unless specifically requested by IXIASOFT support.
<code>-Ddita.cms.mode=%mode%</code>	<p>DITA CMS work mode. Valid values are:</p> <ul style="list-style-type: none"> • STD: Standard mode (i.e., without the DRM module) • DRM: Dynamic Release Management Mode <p>For example:</p> <pre>-Ddita.cms.mode=STD</pre> <p>Note: This value must match the value of the <code>dita.cms.mode</code> parameter specified in the <code>textmlservercfg.properties</code> file for the Content Store.</p>
<code>-Duitracefilename=c:\logs\UITrace.log</code>	(Optional) Enables a trace log that lists the IDs for the different interface items in a perspective. For more information, see "Enable the trace log" in the <i>DITA CMS Administrator's Guide</i> .
<code>-Dixia.initial.preferences=preferences.properties</code>	(Optional) Defines the configuration file to set up the initial working environment for DITA CMS Eclipse client.

3. Repeat this procedure for every packaging directory.

(Optional) Enable Kerberos security

If your deployment uses Kerberos authentication, you must configure the DITA CMS Eclipse client and include Kerberos security files.

Note: The procedure below describes how to create two Kerberos security files: *krb5.ini* and *login.conf*. The latest versions of these files are also available from the IXIASOFT download site at the following URL:

<http://cms.ixiasoft.com/downloads/eclipse/4.2>

You can use these files as templates and configure them as described below.

Pre-requisites

- To use Kerberos authentication, TEXTML Server must be running on Windows. The client application can be running on Windows or Linux.

For every Windows packaging directory that must support Kerberos:

1. Create the *krb5.ini* Kerberos configuration file.

This file specifies the Kerberos configuration used for authentication. Configure it as follows:

- `realm`: Kerberos realm name. This is the Fully Qualified Domain Name (FQDN) of your Windows domain.
- `kdc`: Kerberos Key Distribution Center (KDC) host name and port (Windows domain controller).

For example:

```
[domain_realm]
.acme.local = ACME.LOCAL
acme.local = ACME.LOCAL
acme = ACME.LOCAL
[libdefaults]
  dns_lookup_kdc = true
  dns_lookup_realm = true
[logging]
[realms]
ACME.LOCAL = {
  kdc = dc1.acme.local
  kdc = dc2.acme.local
  admin_server = dc1.acme.local
}
```

2. Save this file in the packaging directory; for example:

```
C:\eclipses\cms42_x64\krb5.ini
```

3. Create the *login.conf* login configuration file.

This file specifies the Java login configuration for Active Directory. It must include the following code:

```
TextmlClientLogin
{
    com.sun.security.auth.module.Krb5LoginModule required useTicketCache=false;
};
```

4. Save this file in the packaging directory; for example:

```
C:\eclipses\CMS42_x64\login.conf
```

5. Set the `java.security.krb5.conf` variable to the location of the `krb5.ini` file.

This variable is configured in the `eclipse.ini` file and must appear after the `-vmargs` line. For example, in the `C:\eclipses\CMS42_x64\eclipse.ini` file, set it as follows:

```
...
-vmargs
-Xms256m
-Xmx1024m
-Dxia.level=DEBUG
-Dsun.java2d.d3d=false
-Dosgi.requiredJavaVersion=1.6
-Dosgi.bundlefile.limit=1000
-Ddita.cms.mode=STD
-Djava.security.krb5.conf=krb5.ini
-Djava.security.auth.login.config=login.conf
```

6. Set the `java.security.auth.login.config` variable to the location of the `login.conf` file.

For example:

```
...
-vmargs
-Xms256m
-Xmx1024m
-Dxia.level=DEBUG
-Dsun.java2d.d3d=false
-Dosgi.requiredJavaVersion=1.6
-Dosgi.bundlefile.limit=1000
-Ddita.cms.mode=STD
-Djava.security.krb5.conf=krb5.ini
-Djava.security.auth.login.config=login.conf
```

7. Repeat this procedure for every Windows packaging directory that must support Kerberos.

Note: You also need to configure the TEXTML Server to enable Kerberos authentication. See the *TEXTML Server Installation Guide* for more information.

(Optional) Include the DITA CMS plug-ins and Oxygen XML Editor/Author as dropins

You can include the DITA CMS plug-ins as well as the Oxygen XML Editor/Author in your initial package using dropins.

This ensures that users will not all have to download large files from the web at the same time.

To include these plug-ins, you download them from the IXIASOFT and Eclipse download sites and extract them to the *dropins* folder of your packaging directories, as follows:

1. Download the DITA CMS plug-ins to install from the DITA CMS 4.2 download site, located at the following URL:

<http://cms.ixiasoft.com/downloads/dropins/4.2/>

The plug-ins to download depend on whether you are preparing a package for a regular user or CMS administrator. Use the table below for the list of plug-ins to install:

Package type	Plug-ins to install
CMS administrator; for example: <code>C:\eclipses\cms42_admin_x64\dropins\</code>	<p>Mandatory:</p> <ul style="list-style-type: none"> <code>com.ixiasoft.dita-cms-admin-<version>.zip</code> <code>com.ixiasoft.textml.textmladmin-<version>.zip</code> <code>com.ixiasoft.dita.help.doc.dita_<version>.zip</code> <code>com.ixiasoft.eclipse.collaborative.viewer-<version>.zip</code> <code>com.ixiasoft.eclipse.dita-cms-<version>.zip</code> <p>One of the following, according to your deployment:</p> <ul style="list-style-type: none"> <code>com.ixiasoft.eclipse.oxygen.author-<version>.zip</code> <code>com.ixiasoft.eclipse.oxygen.editor-<version>.zip</code> <code>com.ixiasoft.eclipse.xmetal.editor-<version>.zip</code> <p>Optional:</p> <ul style="list-style-type: none"> <code>com.ixiasoft.eclipse.hyperSTE-<version>.zip</code> <code>com.ixiasoft.eclipse.ReleaseManagement-<version>.zip</code>
Regular user; for example: <code>C:\eclipses\cms42_x64\dropins</code>	<p>Mandatory:</p> <ul style="list-style-type: none"> <code>com.ixiasoft.dita.help.doc.dita_<version>.zip</code> <code>com.ixiasoft.eclipse.collaborative.viewer-<version>.zip</code>

Package type**Plug-ins to install**

- `com.ixiasoft.eclipse.dita-cms-<version>.zip`

One of the following, according to your deployment:

- `com.ixiasoft.eclipse.oxygen.author-<version>.zip`
- `com.ixiasoft.eclipse.oxygen.editor-<version>.zip`
- `com.ixiasoft.eclipse.xmetal.editor-<version>.zip`

Optional:

- `com.ixiasoft.eclipse.hyperSTE-<version>.zip`
- `com.ixiasoft.eclipse.ReleaseManagement-<version>.zip`

2. For each DITA CMS plug-in to install, in the *dropins* folder of your Eclipse packaging directory, create a subfolder for the plug-in you are installing; use the plug-in name as the folder name.

For example:

```
C:\eclipses\cms42_x64\dropins\com.ixiasoft.dita-cms-4.2.5\  
C:\eclipses\cms42_admin_x64\dropins\com.ixiasoft.dita-cms-4.2.5\  
C:\eclipses\cms42_admin_x64\dropins\com.ixiasoft.dita-cms-4.2.5\
```

3. Extract the content of the downloaded plug-in to this directory.
4. Repeat from step 2 for every plug-in to install.
5. To include oXygen XML, go to the following URLs and click the download button for the Plugin zip distribution:

- For oXygen XML Editor:
http://www.oxygenxml.com/download_oxygenxml_editor.html?os=Eclipse
- For oXygen XML Author:
http://www.oxygenxml.com/download_oxygenxml_author.html?os=Eclipse

6. Extract the plug-in directly to the *dropins* directory.

Note: Because the oXygen XML zip file already contains a subdirectory, you do not need to create a subfolder, as you did for the DITA CMS plug-ins. Otherwise, you will get an error saying that the path is too long.

Once completed, the structure should be similar to the following (the example below shows a CMS administrator package):



7. Repeat this procedure for every packaging directory that uses dropins.

(Optional) Deploy the plug-ins using an update site

If you plan to deploy the plug-ins using an update site, no additional configuration is required in the initial package. However, at runtime the user must import the update site bookmarks into Eclipse. This is described later on.

(Optional) Configure the initial working environment for DITA CMS Eclipse clients

You can include a configuration file in your deployment package to launch the DITA CMS Eclipse clients with preconfigured settings.

When the users launch their DITA CMS Eclipse clients, DITA CMS uses the configuration file to automatically open the defined perspective and configure the connection information. This allows the users to simply enter their passwords and begin working in the correct environment without spending time closing unneeded perspectives and following connection instructions.

The configuration file, called *preferences.properties*, includes settings to define:

- The initial perspective, which is usually the DITA perspective
- The domain and user name
- The TEXTML Server connection details
- The Output Generator connection details

To set up the configuration file:

1. **Download the sample *preferences.properties* file from the following URL:**
<http://cms.ixiasoft.com/downloads/eclipse/4.2/>.
2. **Edit the file to provide the required information:**

Property	Value
<code>dita.cms.initial.perspective</code>	Type the name of the perspective you want opened by default upon starting the DITA CMS Eclipse client, which will typically be the DITA perspective. Recommended value: <code>com.ixiasoft.dita.eclipse.gui.perspective.DitaPerspective</code>
<code>com.ixiasoft.eclipse.dita/textml.hostname</code>	Type the name of the machine hosting the TEXTML Server.
<code>com.ixiasoft.eclipse.dita/textml.username</code>	Type the login name of the user
<code>com.ixiasoft.eclipse.dita/textml.docbase</code>	Type the name of the Content Store.
<code>com.ixiasoft.eclipse.dita/textml.domain</code>	Type the domain name.

Property	Value
<code>com.ixiasoft.eclipse.dita/texttml.port</code>	Type the TEXTML Server port number. If your deployment is using SSL, enter the secure port. Default value: 2500
<code>com.ixiasoft.eclipse.dita/texttml.secure</code>	Type one of the following: <ul style="list-style-type: none"> <code>true</code>: when the Secure Sockets Layer (SSL) protocol is enabled <code>false</code>: when SSL is not enabled
<code>com.ixiasoft.eclipse.dita/outputGenerator.hostname</code>	Type the name of the machine hosting the Output Generator.
<code>com.ixiasoft.eclipse.dita/outputGenerator.port</code>	Type the Output Generator port number. Default value: 1500
<code>com.ixiasoft.eclipse.dita/outputGenerator.monitor.port</code>	Type the local port where output generation progress is sent. Default value: 0

3. Save the file as *preferences.properties* in the packaging directory at the same level as the *eclipse.ini* file; for example:

```
C:\eclipses\CMS42_x64\preferences.properties
```

4. Set the `ixia.initial.preferences` variable to the location of the *preferences.properties* file. This variable is configured in the *eclipse.ini* file and must appear after the `-vmargs` line. For example:

```
-vmargs
-Xms256m
-Xmx1024m
-Dixia.level=DEBUG
-Dsun.java2d.d3d=false
-Dosgi.requiredJavaVersion=1.6
-Dosgi.bundlefile.limit=1000
-Djava.security.krb5.conf=krb5.ini
```

```
-Djava.security.auth.login.config=login.conf  
-Ddita.cms.mode=STD  
-Dixia.initial.preferences=preferences.properties
```

5. Repeat this procedure for every packaging directory.

(Optional) Configure the DITA CMS client in a remote solution

If your deployment includes a remote access solution, then you need to install the DITA CMS Eclipse Client in a Citrix or Remote Desktop farm or on virtual machines.

Virtual machines installations are performed as standard user workstation installations.

For Citrix or Remote Desktop servers, two options are possible depending on the number of users that you must support:

- Install the DITA CMS on a single server that will support all users
- Install the DITA CMS on multiple servers that are load balanced (i.e., in a server farm)

Single server requirements

When installing the DITA CMS Eclipse Client on a single remote server:

- Install the plug-ins using the dropins method only. Update sites are not supported in a remote solution.
- Users don't have write access to the Eclipse directory, so the system administrator must install the updates.

Server farm requirements

When installing the DITA CMS in a server farm:

- Install the plug-ins using the dropins method only. Update sites are not supported in a remote solution.
- Users don't have write access to the Eclipse directory, so the system administrator must install the updates.
- The users' workspaces must be installed on a shared folder so that users can always access their workspaces, no matter which server they are accessing. They must also be able to access this workspace from their workstation.
- Add the following line to the *eclipse.ini* file:

```
-Dcom.ixiasoft.cms.override.hostname=<logical_citrix_server_name>
```

where `<logical_citrix_server_name>` is a name such as `Citrix-Farm-Prod`. This value must be the same for each `eclipse.ini` file on all the servers in your farm.

This ensures that files are locked properly by your users. The DITA CMS Eclipse Client uses the hostname and the username to create a unique lock information. If there are multiple servers in a farm, the hostname used for the lock information is the local server name. With a load balancer, once a session is running, the user will connect back to the same session on the same server. But if the session is closed (the user logs off or closes the application), there is no guarantee that the user will log back in to the same server. Setting the parameter above ensures that files are always locked correctly.

- Create a batch file to start Eclipse, as shown below. Update the values in bold as appropriate for your deployment:

Note: The source of this file is available on the IXIASOFT download site at the following URL:

http://cms.ixiasoft.com/downloads/templates/4.2/start_cms42.bat.txt

Remove the `.txt` extension to make the file executable.

```
@echo off

REM Spaces are not supported in the batch file

REM The examples in this file assume the following directory structure, paths, and filenames:
REM c:\eclipses
REM c:\eclipses\start-cms42.bat
REM c:\eclipses\start-cms42-admin.bat
REM c:\eclipses\cms42_x64\
REM c:\eclipses\cms42_x64\eclipse.exe
REM c:\eclipses\cms42_admin_x64\
REM c:\eclipses\cms42_admin_x64\eclipse.exe

REM Path to directory where all instances of eclipses are installed
set eclipsefolder=C:\eclipses

REM Define the name of this eclipse instance.
REM Update as appropriate for your deployment. This should be the name of the directory in
  the eclipse folder.
set instance=cms42_x64

REM Path to directory where all the user data will be stored. It can also be %home% or
%userprofile%
REM Update as appropriate for your deployment
set datadrive=d:\DitaCMSUsersData

REM The following values define where the workspace and configuration data will be saved
set data=%datadrive%\%username%\workspaces\%instance%
set config=%datadrive%\%username%\configs\%instance%\config

REM If the user creates an empty txt file with the following name in his/her workspace,
then the DITA CMS will start in debug mode
set debug=%data%\debug.txt

if exist %debug% set activedebug=-debug
```

```
echo Workspace folder is %data%
echo Config folder is %config%
if not "%activatedebbug%"==" " echo Debug mode is on
if not "%activatedebbug%"==" " echo The start command is
if not "%activatedebbug%"==" " echo start %eclipsefolder%\%instance%\eclipse.exe -data %data%
-configuration %config% %activatedebbug%
if not "%activatedebbug%"==" " pause

REM See the documentation at http://www.eclipse.org/documentation/ for more information
about the command line options.
start %eclipsefolder%\%instance%\eclipse.exe -data %data% -configuration %config%
%activatedebbug%
```

Finalize the package

Once your package is complete, zip each packaging directories.

Important: Do NOT start Eclipse from the packaging directories. When you start Eclipse, it creates installation files that are specific to the computer on which it is running. If you start Eclipse from your packaging directory, these files will be included when you deploy the DITA CMS. To test your package, zip it, then extract it to another directory to test it.

Deploy the package

To deploy the package:

- Simply unzip it on the machines (user workstations and servers) in your deployment, as follows:
 - Unzip the CMS administrator package on all CMS administrator workstations/servers
 - Unzip the regular user package on all other workstations/servers

Note: If you are using update sites to deliver the plug-ins, make sure that you unzip the package to a directory where users have write permissions. This is necessary because Eclipse updates its installation files upon start-up. If users do not have write access to the Eclipse directories, they will not be able to run the DITA CMS.

- Create a shortcut to the DITA CMS. For user on workstations, this shortcut will point to the eclipse executable in the packaging directory. For the remote access solution, this shortcut will point to the DITA CMS startup batch file.

What's next?

The next step is to tell your users how to start the DITA CMS. IXIASOFT has created two template documents that you can adapt and give to your users. You must update the templates to specify the details of your deployment; for example:

- The location of the user workspace

Note: When starting Eclipse, users must specify a workspace. It is preferable to use a local workspace; you can use network workspaces, but performance issues may arise.

- The TEXTML Server hostname, port number, and Content Store that users connect to
- The Output Generator hostname and port number

The templates are the following:

- *Starting the DITA CMS on a user workstation*

This document is available at the following URL:

http://cms.ixiasoft.com/downloads/templates/4.2/Starting_DITA_CMS_4_2_user_workstation.doc

It describes how to start the DITA CMS from a user workstation and perform configuration steps that are required for the first time (for example, how to connect to the right server and Content Store, how to import bookmarks files for update sites, how to access the documentation, etc.).

- *Starting the DITA CMS in a remote solution*

This document is available at the following URL:

http://cms.ixiasoft.com/downloads/templates/4.2/Starting_DITA_CMS_4_2_remote_solution.doc

It describes how to start the DITA CMS in a remote solution and perform configuration steps that are required for the first time (for example, how to connect to the right server and Content Store, how to access the documentation, etc.).

See each template for instructions on how to update the documents before distributing them to your users.

Troubleshooting

This section describes issues that may occur when deploying the Eclipse package and starting the DITA CMS for the first time.

Illegal Key Size error

If the following error is generated when a user first tries to log in to the DITA CMS using Kerberos authentication:

```
java.security.InvalidKeyException:illegal Key Size
```

Then the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files is required and not installed on the client. To solve this issue, see the Java Download site for more information (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).

Users cannot install software using Eclipse plug-ins

If you are using update sites to deliver the plug-ins, make sure that you unzip the package to a directory where users have write permissions. This is necessary because Eclipse updates its installation files upon start-up. If users do not have write access to the Eclipse directories, they will not be able to install the plug-ins.

Eclipse behaves erratically

If a user complains that the Eclipse client behaves erratically, this may indicate that the workspace is corrupted. Ask your users to create a new workspace; if it solves the problem, this confirms that the user workspace is corrupted.

To create a new workspace:

1. From the Eclipse menu, select **File > Switch Workspace. > Other...**
2. Enter a new workspace.

If creating a new workspace solves the problem:

1. Delete the old (corrupted) workspace.

Note: If the workspace contained locked files—which store changes made to a topic and not yet committed to the server—they will be lost if you delete the workspace. To keep these changes, please contact IXIASOFT Support before deleting the workspace. Note that if you decide to delete the workspace without keeping the changes, when users open the DITA CMS in the new workspace, these locked topics will be displayed in purple. Users must revert these topics to the server version by selecting the topics, right-clicking, and choosing **Replace with Server Revision**.

2. Rename the new (working) workspace to the old workspace's name.

Keeping the original name ensures that batch files (to start the DITA CMS or to update plug-ins) will continue to work properly.

3. Ask your users to reconfigure their workspace using the "Configure your workspace" section in the template documents.
4. If you are using update sites, users also need to reimport their bookmarks. They do not need to reinstall the plug-ins.

Updating the DITA CMS Eclipse Client

When new updates of the DITA CMS plug-ins are available, IXIASOFT sends an email with a list of changes.

You should test the new plug-ins before you install them in production, whether you are using dropins or update sites. This ensures that the updates have minimal impact on your current configuration.

Updating the DITA CMS plug-ins when using dropins

This procedure describes how to update the DITA CMS plug-ins when using dropins.

To update the DITA CMS using dropins, you download the plug-ins from the IXIASOFT download site and extract them to the Eclipse *dropins* folder. If you have many versions of the same plug-in in the *dropins* directory, Eclipse loads the latest version.

It is up to you to determine how these plug-ins will be made available to the users. Some options include:

- Automatically pushing the plug-ins to the user workstations or servers. The plug-ins will be loaded the next time users restart their DITA CMS. If you need to make an important update, you could ask users to restart their DITA CMS Eclipse Client right away. Note that you can push the updates to the workstations even if the DITA CMS Eclipse Client is running.
- Putting the plug-ins in a central location and having users manually update them. For example, you could create an *update-cms.bat* script that copies the plug-ins from the central location to the workstations/servers and ask users to run the script as necessary. For example:

Note: The source of this file is available on the IXIASOFT download site at the following URL:

<http://cms.ixiasoft.com/downloads/templates/4.2/updatecms42.bat.txt>

Remove the *.txt* extension to make the file executable.

```
@echo off

REM The source of the dropins must be available on the network and be accessible by users
REM For example: \\acme-server\dita-cms-updates
REM Under this shared source, create a copy of each instance of your installation packages
REM For example:
REM   \\acme-server\dita-cms-updates\cms42_x64
REM   \\acme-server\dita-cms-updates\cms42_x64\eclipse.exe
REM   \\acme-server\dita-cms-updates\cms42_x64\dropins
REM   \\acme-server\dita-cms-updates\cms42_x64\plugins
REM   \\acme-server\dita-cms-updates\cms42_x64\jre
REM   \\acme-server\dita-cms-updates\cms42_admin_x64\
REM Update this installation package as necessary and then tell users to run this batch
```

```

file

REM Save this batch file in the root of each package directory, on the user machines and
on the central location
REM For example:
REM C:\eclipses\cms42_x64\update-cms.bat
REM \\acme-server\dita-cms-updates\cms42_x64\update-cms.bat

REM NOTE: if you use the following command to update the DITA CMS, Eclipse must NOT be
running
REM because this command will try to override files that are in use and cannot be replaced
REM Update the following line for each package directory
robocopy /mir \\acme-server\dita-cms-updates\cms42_x64 c:\eclipses\cms42_x64

REM To update the CMS while Eclipse is running, or if you simply added dropins or updated
the eclipse.ini or start-cms.bat file
REM you can use the following command instead.
REM Update the following line for each package directory
REM robocopy /S /E \\acme-server\dita-cms-updates\cms42 c:\eclipses\cms42_x64

```

This utility replaces an existing installation with the updated installation packages. Save this file in the root of each installation package as well as on the central location. For example:

```

C:\eclipses\cms42_x64\update-cms.bat
C:\eclipses\cms42_admin_x64\update-cms.bat
\\acme-server\dita-cms-updates\cms42_x64\update-cms.bat
\\acme-server\dita-cms-updates\cms42_admin_x64\update-cms.bat

```

Updating the DITA CMS plug-ins using an update site

This procedure describes how to update the DITA CMS plug-ins using an update site.

To update the plug-ins on the update site:

Note: This procedure assumes that you have two sets of update sites: test update sites and production update sites.

1. Get the latest plug-ins from the IXIASOFT download site at the following URL:

<http://cms.ixiasoft.com/downloads/plugins/4.2/>

2. Delete the content of the update site folders.

For example, delete the content of the following folders (keeping the same directory structure):

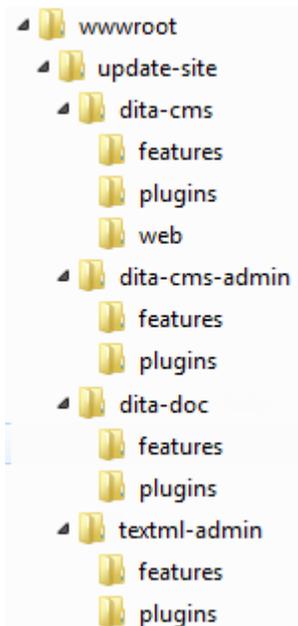
- *C:\wwwroot\update-site\dita-cms*
- *C:\wwwroot\update-site\dita-doc*
- *C:\wwwroot\update-site\dita-cms-admin*
- *C:\wwwroot\update-site\textml-admin*

3. Unzip the content of each update site zip file to the appropriate folder on the test update sites.

For example:

Update site zip file	Directory
<i>dita-cms-<version>.zip</i>	<i>C:\wwwroot\update-site\dita-cms</i>
<i>dita-doc-<version>.zip</i>	<i>C:\wwwroot\update-site\dita-doc</i>
<i>dita-cms-admin-<version>.zip</i>	<i>C:\wwwroot\update-site\dita-cms-admin</i>
<i>textml-admin-<version>.zip</i>	<i>C:\wwwroot\update-site\textml-admin</i>

For example:



4. Test the plug-ins.
5. When satisfied that the plug-ins work as expected, repeat the procedure to install the plug-ins on the production update site.

A

List of DITA CMS plug-ins

The following DITA CMS plug-ins are available for all users:

- **DITA Content Management System:** Main DITA CMS functionality.
- **Collaborative Reviewer:** Allows DITA CMS reviewers to provide feedback on topics and allows authors to consult and revise topics that have been commented.
- **DITA 1.2 Specification:** Provides the rules for publishing DITA content in print, HTML, online Help, and other formats.

The following plug-ins are available for DITA CMS administrators only:

- **DITA CMS Administration:** Provides access to the DITA CMS Administration Console.
- **TEXTML Server Administration:** Allows workstation users to interact with IXIASOFT's TEXTML Server component, typically installed on the server that holds the Content Store.

The following optional plug-ins are also available:

- **oXygen XML Extension (Editor or Author):** Provides the DITA CMS context menus in the oXygen XML Editor installed in Eclipse; a third-party license is required to run

this plug-in. It also installs the *DITA CMS User Guide for oXygen* online help.

- **XMetaL XMAX Extension:** Provides the DITA CMS context menus in the XMAX Editor installed in Eclipse; a third-party license is required to run this plug-in. It also installs the *DITA CMS User Guide for XMAX* online help.
- **HyperSTE:** Activates the HyperSTE functions for XMAX; a third-party license is required to run this plug-in.
- **Dynamic Release Management:** Allows technical publication teams to easily manage the documentation for multiple products across multiple release cycles. This optional module requires a special license. Please contact IXIASOFT Services for more information.

B

DITA CMS Eclipse Client software requirements

The software requirements for the DITA CMS Eclipse Client are as follows:

- Eclipse SDK 4.3 with the DITA CMS plug-ins
- Versions of OS supported:
 - Windows 7 64-bit version is recommended, but the following are supported: Windows Vista and Windows 7 32-bit version
 - Any Linux platform supported by Eclipse SDK 4.3 (32-bit or 64-bit version)
- Oracle Java 1.7.0_51+ (Java 1.8 is not supported)



XMetaL XMAX requirements

The following lists summarizes the most important system requirements for XMetaL XMAX 8.0. See the XMetaL XMAX installation documentation for more information.

- Microsoft Windows Vista, Windows 7, or Windows 8 (32-bit or 64-bit), or Server 2008
- Eclipse SDK 4.3.2, 32-bit only
- An internet connection with port 443 open (or a LAN connection for on-site XMLS configurations)
- Microsoft Internet Explorer, version 6.00.2600.0000 to version 10 with EPM disabled
- Active internet connection that permits an http/https connection to the jsls.xmetal.com site

Known issues during installation

If you get the following message during installation:

```
"Registration failed"
```

Make sure that your computer has a working internet connection and that a firewall is not blocking the TCP/IP port 443. This port (https) is used by XMAX to communicate with an XMetaL license server. Also, make sure that you have unused licenses available to allocate to users. If you have an

XMetaL License server on-site, check with your IT administrator. Otherwise, contact XMetaL Support.

Index

A

authentication schemes 9

B

batch file for starting DITA CMS 34

BIRT 16

update sites 16

bookmarks file 16

C

Citrix farms 34

client installation overview 9

CMS administrator 16

bookmarks file 16

corrupted workspace 37

D

dependencies for plug-ins 19

deploying the Eclipse package 36

DITA CMS Eclipse Client 9

installation overview 9

installing 9

DITA CMS plug-ins 43

Dixia.level 25

Dosgi.bundlefile.limit 25

download Eclipse SDK 22

download site 14, 29, 40

for IXIASOFT dropins 29, 40

for IXIASOFT plug-ins 14

dropins 10, 29, 39

in initial Eclipse package 29

updating 39

E

Eclipse package 20, 36

creating 20

deploying 36

finalizing 36

Eclipse plug-ins 10–12, 14, 19, 29, 39–40

dependencies 19

dropins 10, 29

hybrid solution 12

update sites 11, 14

updating 39–40

Eclipse SDK 20, 22

download 22

eclipse.ini 25, 27, 34

F

finalizing the Eclipse package 36

H

hybrid solution for plug-ins 12

I

Illegal Key Size error 37

initial Eclipse package 20

Installing DITA CMS Eclipse Client 9

IXIASOFT update sites 16

J

Java Runtime Environment, adding to packaging directory 23

JRE, adding to packaging directory 23

K

Kerberos authentication scheme 9, 27

krb5.ini 27

L

local authentication scheme 9

login.conf 27

O

overview 9

DITA CMS Eclipse Client installation 9

oXygen XML 13, 16, 29

drop-ins site 29

update sites 16

P

packaging directory 22

plug-ins 10

Plug-ins 43

list for DITA CMS 43

R

regular user 16

bookmarks file 16

Remote Desktop farms 34
remote solution 34

S

script for updating plug-ins 39
start-cms.bat 34

T

templates 34, 36, 39
 batch file for starting DITA CMS 34
 for starting the DITA CMS (end users) 36
 script for updating plug-ins 39
troubleshooting 37

U

update sites 11, 14, 16, 40
 BIRT update site 16
 bookmarks files 16
 creating 14
 for production 14

update sites (*continued*)
 for tests 14
 IXIASOFT update sites 16
 oXygen update sites 16
 updating plug-ins 40
update-cms.bat 39
updating plug-ins 39–40
 using dropins 39
 using update sites 40

V

virtual machines 34

W

web server 14
workspace, corrupted 37

X

XMetal XMAX 13
XML Editor requirements 13