

**WIND™**

# WIND RIVER®

## DITA USER'S GUIDE

21 February 2020



## **Copyright Notice**

Copyright © 2020 Wind River Systems, Inc.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the prior written permission of Wind River Systems, Inc.

Wind River, Simics, Tornado, and VxWorks are registered trademarks of Wind River Systems, Inc. Helix, Pulsar, Rocket, Titanium Cloud, Titanium Control, Titanium Core, Titanium Edge, Titanium Edge SX, Titanium Server, and the Wind River logo are trademarks of Wind River Systems, Inc. Any third-party trademarks referenced are the property of their respective owners. For further information regarding Wind River trademarks, please see:

[www.windriver.com/company/terms/trademark.html](http://www.windriver.com/company/terms/trademark.html)

This product may include software licensed to Wind River by third parties. Relevant notices (if any) are provided for your product on the Wind River download and installation portal, Wind Share:

<http://windshare.windriver.com>

Wind River may refer to third-party documentation by listing publications or providing links to third-party websites for informational purposes. Wind River accepts no responsibility for the information provided in such third-party documentation.

## **Corporate Headquarters**

Wind River  
500 Wind River Way  
Alameda, CA 94501-1153  
U.S.A.  
Toll free (U.S.A.): +1-800-545-WIND  
Telephone: +1-510-748-4100  
Facsimile: +1-510-749-2010

For additional contact information, see the Wind River website:

[www.windriver.com](http://www.windriver.com)

For information on how to contact Customer Support, see:

[www.windriver.com/support](http://www.windriver.com/support)

*DITA User's Guide, 21 February 2020*

24 February 2020

# **Contents**

<b>1 Ixiasoft CCMS Client Overview .....</b>	<b>1</b>
The Ixiasoft Content Management System .....	1
Understanding Differences Between SVN and the Ixiasoft CCMS .....	2
Ixiasoft CCMS Installation and Configuration .....	3
The DITA Perspective .....	3
Additional Documentation .....	6
<b>2 Logging into a Different Server .....</b>	<b>9</b>
<b>3 Importing Documents into the Ixiasoft CCMS .....</b>	<b>11</b>
About Importing Documents into the Ixiasoft CCMS .....	11
Preparing SVN Documents for Import .....	12
Considerations for Importing Conditionalized Files .....	13
Checking the key2href Results .....	13
Importing Files .....	15
Making Final Changes in the Ixiasoft CCMS .....	16
<b>4 The Document Life Cycle .....</b>	<b>17</b>
The Document Life Cycle .....	17
Cycle and Status Dependencies .....	18
Workflows .....	19
Workflow: Creating New Product Documentation .....	19
Workflow: Review Process .....	20
Workflow: Finalizing your Documentation .....	21
Workflow: Creating a New Lab Module .....	21
Workflow: Creating a New Lecture Module (Placeholder) .....	22
Daily Activities Checklist for Using DITA in the Ixiasoft CCMS .....	23
<b>5 File Management .....</b>	<b>25</b>
About Files in the Ixiasoft CCMS .....	25
Locking Files for Editing .....	26
Change Tracking .....	26
Change Tracking Overview .....	26
Enabling Change Tracking .....	28
Releasing Files to the Database .....	28
Changing File Status .....	30

Assigning Files to Another Person .....	31
Viewing Files in the Todo List .....	31
The Todo List View .....	31
Opening the Todo List View .....	32
Grouping Items in the Todo List View .....	33
Deleting Files .....	33
Restoring Files .....	34
Running the Validate Links Tool .....	34
Product Documentation Release Checklist .....	36
<b>6 Searching in the Ixiasoft CCMS .....</b>	<b>39</b>
The Ixiasoft CCMS Search View .....	40
The Cycles Pane .....	41
The Document Types Pane .....	42
The Search For Pane .....	43
The Limit To Pane .....	44
The Languages Pane .....	45
The Dates Pane .....	46
The Advanced Search Pane .....	46
The Taxonomy Terms Pane .....	48
The Queries Pane .....	48
The Search Results View .....	49
Searching Using a Search String .....	50
Searching by Locked Status .....	51
Searching by Date .....	52
Searching by Label .....	54
Searching by Document ID .....	54
Searching for All References to a Specific Topic .....	55
Searching for Topics that Contain a Specific Element .....	55
Searching for Topics that Contain an Element with Particular Text .....	55
Using Search to Prepare Your Book for Publishing .....	56
Filtering Search Results Based on Text in the Title .....	57
Adding Columns to the Search Results View .....	57
Saving Searches .....	58
Searching Using the AND Operator .....	59
Searching Using the OR Operator .....	60
Making Bulk Changes to XML .....	60
<b>7 Using oXygen .....</b>	<b>65</b>
The oXygen Editor .....	65
Opening the oXygen XML Author Perspective .....	66
Author Mode .....	66

Adding Elements in Author Mode .....	67
Adding Attributes Using the Attributes View .....	67
Adding Attributes Using the Edit Attributes Menu Item .....	68
Adding Elements in Text Mode .....	68
Adding Attributes in Text Mode .....	69
Working in Grid Mode .....	69
Editor Error Feedback .....	70
Shortcut Keys Reference .....	72
<b>8 Maps .....</b>	<b>75</b>
Types of Map Templates .....	75
The DITA Map View .....	76
Creating a New Map .....	77
Opening a Map .....	79
Comparing Maps .....	80
<b>9 Eclipse Help Maps .....</b>	<b>83</b>
Setting Up an Eclipse Help Map .....	83
<b>10 Eclipse Help Submaps .....</b>	<b>87</b>
Setting Up an Eclipse Help Submap in Text Mode .....	87
Adding Links to an Eclipse Help Submap Using the DITA Map View .....	89
Linking to a Bookmap When There is Only One Book in the Plugin .....	90
Adding Subcategories to Your Plugin for Workbench Help .....	91
Linking to a Non-DITA Topic Set .....	92
Linking to a Non-DITA Topic Set that Must Be Wrapped .....	92
Inserting a Plugin for a Sub-Product Inside a Product Plugin .....	93
Linking to a PDF File .....	94
<b>11 Bookmaps .....</b>	<b>97</b>
Adding Product Information to a Bookmap .....	97
Adding Taxonomy Metadata to a Bookmap .....	99
Changing the Bundle Name .....	100
The Bundle Name .....	100
Changing the Bundle Name for a New Document .....	101
Replacing an Existing Doc with an Updated Version .....	102
Replacing Bundles that Contain Edition Numbers .....	103
Changing the Bundle Name Using Conditional Text .....	104
Updating the Metadata for Inter-Book Linking .....	105
Adding an Edition to a Bookmap .....	106
Adding Chapters to Your Bookmap .....	107
Adding Parts to Your Bookmap .....	108

<b>12 Topics Submaps .....</b>	<b>109</b>
Adding Existing Topics to a Map .....	109
Creating Topic Stubs .....	110
Creating a Hierarchy in a Submap Using Topic References .....	113
Creating a Navigation Structure in a Submap Using Topic Headings .....	114
Creating Topic Groups in Submaps .....	115
Using the @chunk Attribute in a Submap .....	117
<b>13 Topics .....</b>	<b>119</b>
Creating Topics in the Ixiasoft CCMS .....	119
Generating Topics from Stubs .....	122
Previewing Content .....	123
<b>14 Concept Topics .....</b>	<b>125</b>
DITA Concept Structure .....	125
Workflow Topics .....	126
<b>15 Task Topics .....</b>	<b>127</b>
Guidelines for Creating Tasks .....	127
DITA Task Structure .....	129
Step Structure .....	131
Prerequisites .....	132
Troubleshooting Information .....	133
Related Links in Task Topics .....	134
Formatting Steps that Involve Commands .....	134
Step Results .....	135
<b>16 Reference Topics .....</b>	<b>137</b>
Guidelines for Creating Reference Topics .....	137
DITA Reference Structure .....	139
<b>17 Referable-Content Topics .....</b>	<b>141</b>
Guidelines for Creating Referable-Content Topics .....	141
Example Referable-Content Topics .....	142
<b>18 Resource Files .....</b>	<b>145</b>
Creating Resource Files .....	145
Editing Resource Files .....	146
Deleting Resource Files .....	147
Importing Resource Files that Were Exported from the Ixiasoft CCMS .....	147
<b>19 Glossaries .....</b>	<b>149</b>

Creating a Glossary Topic .....	149
Adding a Glossary to a Bookmap .....	150
Referencing a Glossary Entry .....	150
<b>20 Short Descriptions .....</b>	<b>153</b>
Guidelines for Writing Short Descriptions .....	153
Examples of Short Descriptions .....	155
<b>21 Inline Elements .....</b>	<b>159</b>
DITA Elements for Inline Content .....	159
Quick Reference Sheet of DITA Elements for Inline Content .....	171
<b>22 Wide Codeblock Elements .....</b>	<b>173</b>
<b>23 Graphics .....</b>	<b>175</b>
Wind River Graphics Standards .....	175
Creating Images in the Ixiasoft CCMS .....	181
Including Images in a Topic from the Ixiasoft CCMS Repository .....	183
Editing Images .....	184
Automated Image Resizing in PDF Output .....	185
Creating Screen Captures .....	185
About Snagit 12 .....	185
Creating Screen Captures Using Snagit 12 (Windows) .....	186
Resizing Screen Captures Using Snagit 12 .....	188
Creating Screen Captures on Linux .....	190
Creating Images Using Visio 2007 .....	191
Connecting to Visio 2007 Using the Remote Desktop Application .....	191
Setting the Text Quality Options .....	191
Creating a Drawing Using a Stencil .....	192
Creating a Drawing Using a Template .....	192
Saving the Image Source .....	193
Saving an Image as a .png File .....	193
Visio Quick Reference .....	194
<b>24 Links .....</b>	<b>199</b>
Updating @scope Attributes in Cross-References Using an XSLT Script .....	199
Creating Links Between Topics in the Same Book .....	201
Creating Links to Targets Within Topics .....	201
Generating ID Attributes .....	202
Creating Links to External Sources .....	203
Creating Links to a Topic in a Different Book .....	203
Creating Links to Documents on the Product Documentation Site .....	205
Creating Links to Hierarchy Locations on the Product Documentation Site .....	206

Creating Links to Videos on the Product Documentation Site .....	207
<b>25 Relationship Tables .....</b>	<b>209</b>
Relationship Tables Overview .....	209
Options for Structuring Relationship Tables .....	210
Models for Relationship Tables .....	211
Opening the ReTable Editing Perspective .....	211
Adding a Relationship Table to a Map .....	213
Editing Relationship Tables .....	214
Best Practices for Relationship Tables .....	216
<b>26 Tables .....</b>	<b>217</b>
General Guidelines .....	217
Types of Tables .....	218
Simple Tables .....	218
Complex Tables .....	218
Choice Tables .....	219
Properties Tables .....	220
Creating Simple Tables with the Insert Table Dialog Box .....	221
Creating Complex Tables with the Insert Table Dialog Box .....	222
Creating Choice Tables with the Insert Choice Table Dialog Box .....	223
Resizing Columns in Simple Tables, Choice Tables, or Property Tables .....	224
Resizing Columns in Complex Tables .....	225
Spanning Columns in a Complex Table .....	225
Spanning Rows in a Complex Table .....	227
Splitting Cells in a Spanned Row or Column .....	228
The Table Toolbar Reference .....	229
<b>27 Indexes .....</b>	<b>231</b>
About Indexes .....	231
Adding an Index .....	231
Nesting Index Elements .....	232
Additional Index Elements .....	233
<b>28 Reusing Content .....</b>	<b>235</b>
Content Reuse .....	235
Including a Topic in Multiple Books .....	236
Including a Topic Multiple Times in the Same Book .....	236
Reusing Elements from Referable-Content Topics .....	237
Conditional Content .....	240
Conditional Content Overview .....	240
Guidelines for Using Conditional Content .....	241
Conditionalizing Topic Content .....	243

Conditionalizing Maps .....	245
Creating a Ditaval File .....	246
Finding the ID of a ditaval File .....	249
Generating Filtered Output .....	249
Conditional Dependencies Can Prevent Publish .....	251
Keyrefs/Variables .....	251
Keywords/Variables Overview .....	251
Guidelines for Using Keywords/Variables .....	252
Creating a Key Map .....	253
Adding a Key Map to Your Bookmap .....	254
Adding a Keyword/Variable to Your Content .....	255
Keyword Examples .....	255
<b>29 Generating Output .....</b>	<b>257</b>
Running Generate Output .....	257
Wind River Supported Output Formats .....	259
<b>30 Delivering Documentation .....</b>	<b>261</b>
Release Management Practices for Product Documentation Delivery (Placeholder) .....	261
What to Deliver (Placeholder) .....	261
Where to Deliver (Placeholder) .....	261
Delivering Product Documentation (Placeholder) .....	261
Release Management Practices for Customer Training Documentation Delivery .....	262
What to Deliver .....	262
Where to Deliver .....	262
Delivering Customer Training Documentation .....	264
<b>31 Cloning .....</b>	<b>267</b>
Cloning Best Practices .....	267
Cloning Files .....	268
<b>32 Publishing .....</b>	<b>269</b>
Publishing Overview .....	269
Validating Links .....	270
Changing Resource Status .....	272
Publishing a Map .....	272
Regenerating Content from a Published Map .....	274
<b>33 Branching .....</b>	<b>275</b>
When to Branch .....	275
What to Branch .....	276
Creating a New Branch .....	278
Branching a Bookmap or Submap .....	279

Branching a Topic .....	280
Branching an Image .....	280
Branching a Conref .....	281
Branching a Referenced Resource .....	282
<b>34 Snapshots .....</b>	<b>283</b>
Snapshot Overview .....	283
Creating a Snapshot .....	284
Searching for a Snapshot .....	285
<b>35 The WIP Site .....</b>	<b>287</b>
About the WIP Site .....	287
Posting a Local Build to WIP .....	289
Customizing the TOC by Editing the index.php File .....	290
Adding Custom Content Using a custom.html.txt File .....	291
<b>36 Troubleshooting .....</b>	<b>293</b>
Isolating the Reason for Build Failure .....	293
Reviewing the Error Logs .....	294
Common Error Messages Reference .....	295
Finding Out-of-Scope Links .....	297

# 1

## *Ixiasoft CCMS Client Overview*

[The Ixiasoft Content Management System](#) 1

[Understanding Differences Between SVN and the Ixiasoft CCMS](#) 2

[Ixiasoft CCMS Installation and Configuration](#) 3

[The DITA Perspective](#) 3

[Additional Documentation](#) 6

### **The Ixiasoft Content Management System**

The Ixiasoft component content management system (CCMS) is used to store and publish DITA topics, maps, images, and resources.

Unlike familiar file system-based documents and topic sets, the Ixiasoft CCMS stores files in a flat structure. Instead of using the file system to organize and locate individual files, the Ixiasoft CCMS depends on a search function and on the metadata associated with each file.

You can find maps and topics that are included in a book or Eclipse map by browsing through the higher-level map. You can also search the database, filtering your search in many ways; for example:

- the status of the document in the production cycle
- the type of document (topic, map, image, resource)
- a string in the document
- the locked state of the document
- the topic metadata

### **High-Level Architecture**

Access to the document base is provided by one of two clients: a local client or a remote desktop (RDP) client.

The DITA perspective, the Eclipse perspective where you interact with the document base, is the same for both clients. If you are local to the server, the local client is the best option. If you are remote, the best client depends on what you are doing.

Local Client	RDP Client
Runs on your local machine.	Runs on the remote Ixiasoft CCMS server.
Faster for tasks that do not access the document base.	Faster if you are remote from the server for tasks that require many server accesses, such as searching the document base and generating output.

The client connects to a server and an output generator. The server provides access to the documentation repository. The output generator allows you to generate the final documentation output.

There are two instances each of the server and output generator: production and development.

The production server is where you do the majority of your work and is where you generate output for customer facing documentation.

The development server and output generator contain recent additions and changes to the Ixiasoft CCMS that are not fully tested and, therefore, should not be used to develop customer-facing documentation. However, you can use the development server as a "sandbox".

## Understanding Differences Between SVN and the Ixiasoft CCMS

SVN is a file management system, while the Ixiasoft CCMS is a topic and map management system.

SVN	Ixiasoft CCMS
File structure; locate topics by structure or use <b>grep</b>	No file structure; locate topics by search
The file name, topic ID, and title have a defined relationship	The topic has a unique ID no matter what the title is; the title can change without changing the ID
Human-readable file names	File name is determined by the topic ID; consists of a long random string
No structured workflow	Structured workflow
Publish using tags; if it builds, it can be published	Publish using tool; requirements are more stringent than for building and depend on document cycles and status

## Ixiasoft CCMS Installation and Configuration

Before you can author your topics, you must install and configure either a local CCMS client or a remote desktop (RDP) CCMS client.

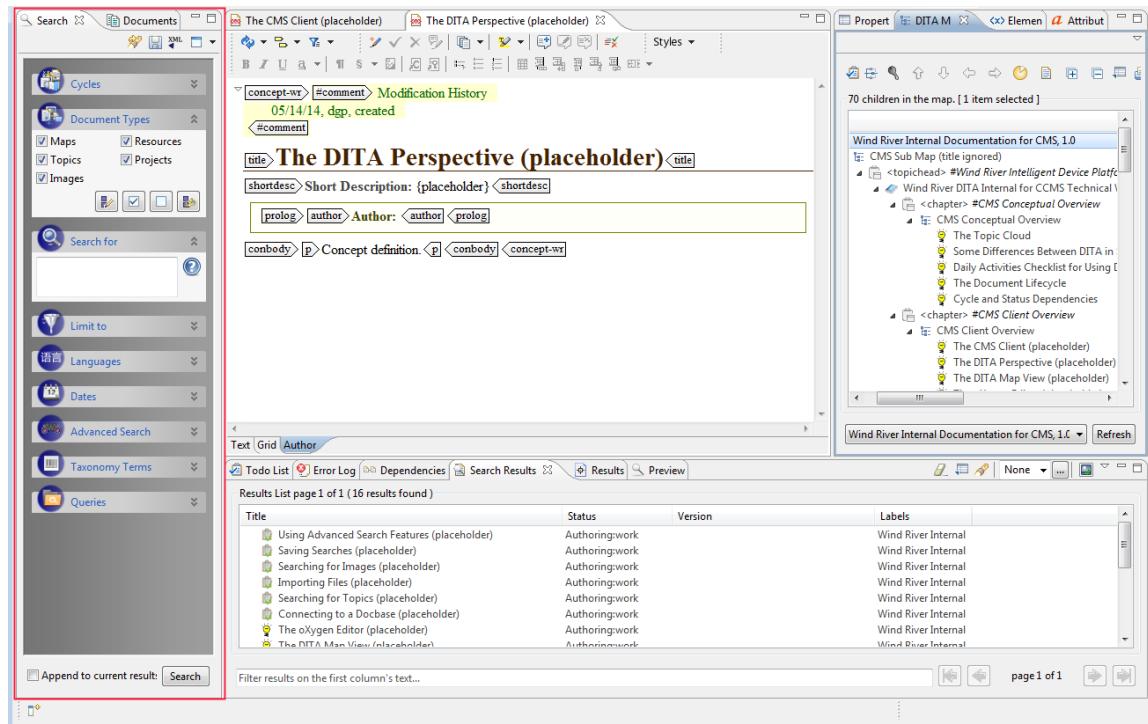
Installation and configuration instructions are provided in the *Wind River CMS Installation and Configuration Guide*.

## The DITA Perspective

The DITA perspective is the Eclipse perspective that supports the Ixiasoft CCMS.

The DITA Perspective consists of four main panes each of which contains multiple views. Certain views are present by default. You can open additional views using **Window > Show View**. You can move a view to a different pane than the default by drag-and-drop.

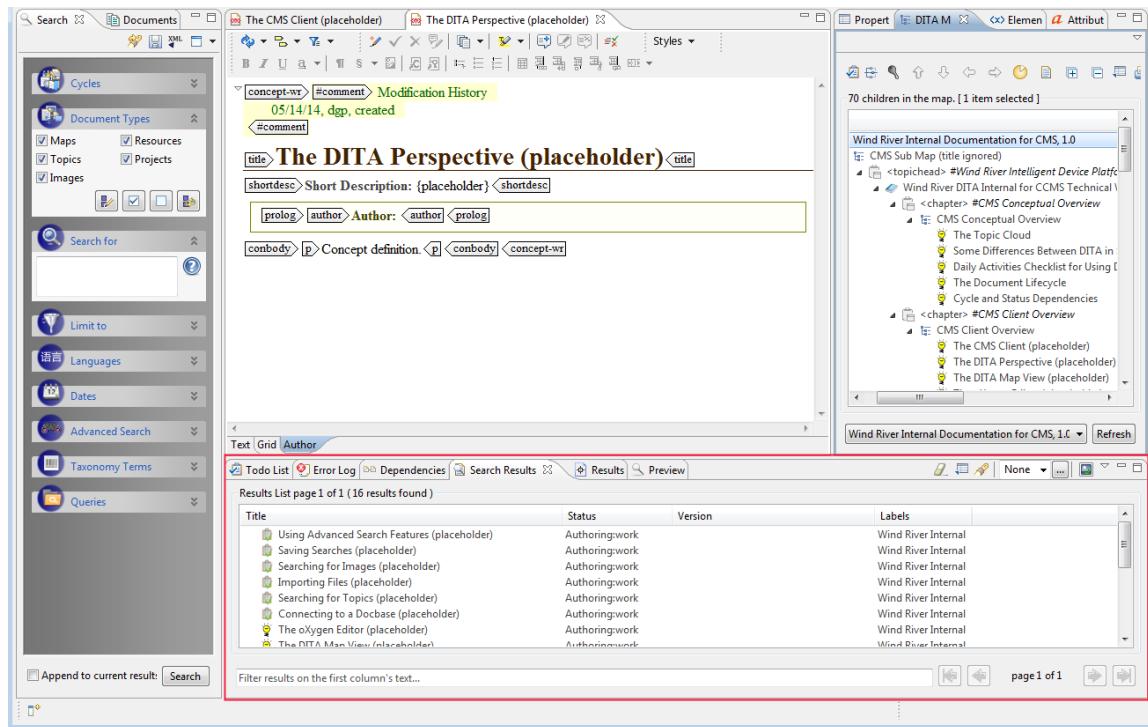
### The Search View



The Search view appears in the left-hand pane of the DITA perspective by default. The Search view allows you to locate DITA documents of all types and to filter your results. For more information, see [The Ixiasoft CCMS Search View](#) on page 40.

The same pane also includes the Documents view by default. This view maintains lists of recent documents you have accessed and recent operations you have performed. You can also place documents on a Favorites list for easy access.

### The Search Results View



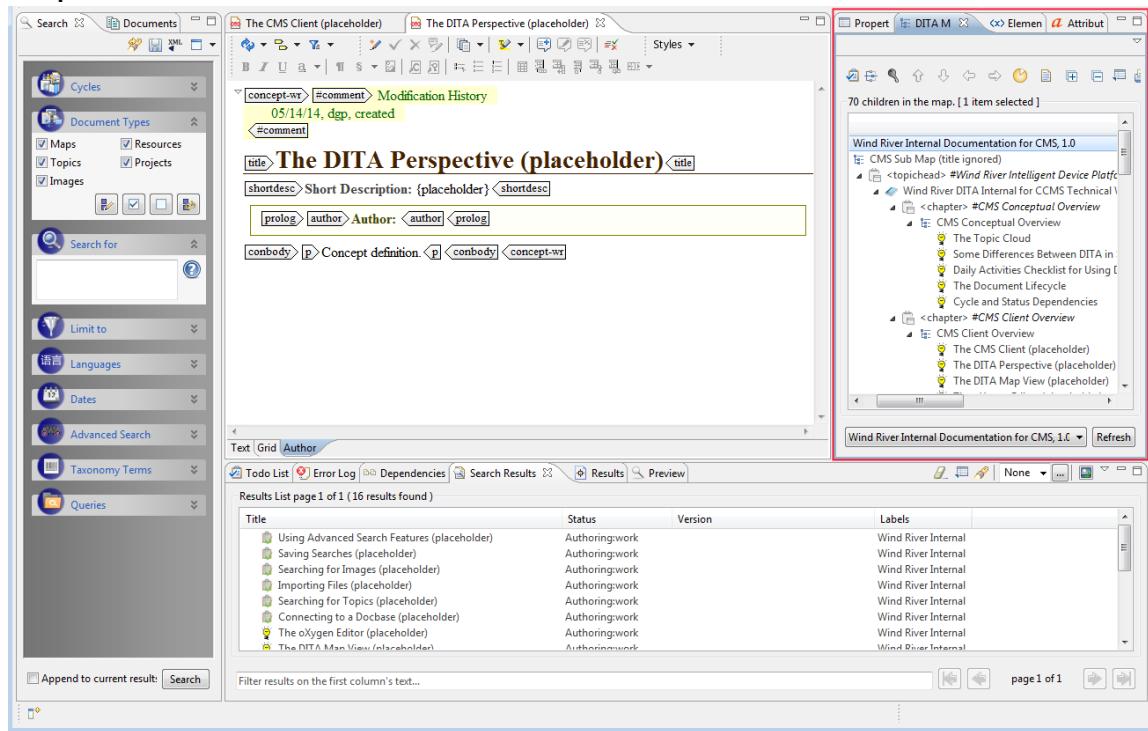
The Search Results view appears in the bottom window of the DITA perspective by default. This key view allows you to lock, open, and release documents as well as perform other operations such as cloning and labeling documents.

The same pane also includes the following views by default:

- Todo List
- Error Log
- Dependencies
- Results

Other views including the Preview view open in this pane by default.

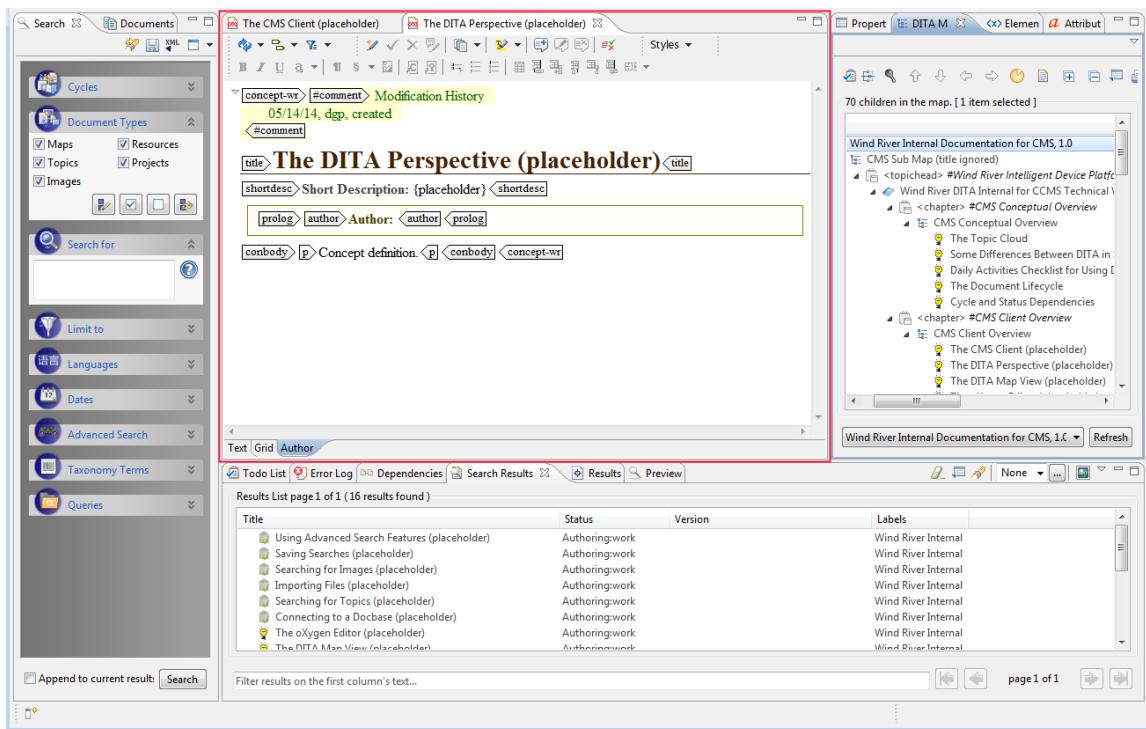
## The DITA Map View



The DITA Map view appears in the upper-right corner of the DITA perspective by default. This key view shows the contents of a DITA map in hierarchical form. It allows you to lock, open, and release documents as well as perform other operations such as cloning and labeling documents.

The same pane also includes the Properties view by default. This view shows the properties of the document selected in the DITA Map view.

## The Editor View



The Editor view opens in the top center of the DITA perspective by default. This view is where you edit maps and topics. Multiple documents can be open in this view; each appears on a separate tab.

## Additional Documentation

The Ixiasoft and Eclipse websites provide additional documentation about the Ixiasoft CCMS and Eclipse Workbench.

### Ixiasoft Documentation

The latest documentation for the Ixiasoft CCMS is located at:

<https://www.ixiasoft.com/documentation-guide/>

The CCMS Desktop User Guide is located at:

[https://www.ixiasoft.com/documentation/IXIASOFT\\_CCMS/6.0/User\\_Guides\\_Advanced\\_User\\_Standard/en/per1389985945656.html](https://www.ixiasoft.com/documentation/IXIASOFT_CCMS/6.0/User_Guides_Advanced_User_Standard/en/per1389985945656.html)

### Eclipse Documentation

**NOTE:**

As recommended by Ixiasoft, Wind River is using Eclipse 4.3 (Kepler). Eclipse documentation is located at:

<http://www.eclipse.org/documentation/>

Some documents, including the *Workbench User Guide*, are available from the **Help > Help Contents** menu.

Some helpful topics in the *Workbench User Guide* are as follows:

- For information on configuring Eclipse, see **Workbench User Guide > Tasks > Customizing the Workbench**.
- For tips and tricks, see **Workbench User Guide > Tips and tricks**.
- For information on setting Eclipse preferences, see **Workbench User Guide > Reference > Preferences**.



# 2

## *Logging into a Different Server*

Typically, most of your work in the Ixiasoft CCMS is done on the production server. However, occasionally you may need work on the development server. Changing servers involves changing the Ixiasoft CCMS configuration values in the Preferences dialog box.

### **Procedure**

1. On the Ixiasoft CCMS Client toolbar, click the red or green ball.

The ball is red if you are not currently logged into a server. The ball is green if you are already logged in.

The Preferences dialog opens with CMS Configuration selected.

2. Fill in the User Information section.

Domain	<b>Corp</b>
User	Your Windows user name
Password	Your Windows password

3. Fill in the TEXTML Server Connection section.

The settings are different for the production and development server.

<b>Setting</b>	<b>Production Server</b>	<b>Development Server</b>
Hostname	<b>ala-ixia-cms</b>	<b>ala-ixia-cms-t</b>
Port	<b>2500</b>	<b>2500</b>
Document base	<b>prod-cms60</b>	<b>dev-cms60</b>

4. Fill in the Output Generator section.

<b>Setting</b>	<b>Production Server</b>	<b>Development Server</b>
Hostname	<b>ala-ixia-gen</b>	<b>ala-ixia-gen-t</b>
Port	<b>1560</b>	<b>1560</b>

Setting	Production Server	Development Server
Monitor Port	0	0

5. Click **Apply**.



**NOTE:** You must click **Apply** before you click **Apply and Close** or the information you entered is lost and the login does not complete.

After a few seconds, the process completes and the progress dialog closes.

6. Click **Apply and Close**.

# 3

## *Importing Documents into the Ixiasoft CCMS*

[About Importing Documents into the Ixiasoft CCMS](#) 11

[Preparing SVN Documents for Import](#) 12

[Considerations for Importing Conditionalized Files](#) 13

[Checking the key2href Results](#) 13

[Importing Files](#) 15

[Making Final Changes in the Ixiasoft CCMS](#) 16

### **About Importing Documents into the Ixiasoft CCMS**

You can import any type of XML document. XML documents created in SVN can be imported into the Ixiasoft CCMS with minor changes.

You can import individual documents (topics or images) into the Ixiasoft CCMS. You can also import maps, which import all the children of the map as well.

In SVN, links are created using the `@keyref` attribute. In the Ixiasoft CCMS, links must use the `@href` attribute. The `key2href` script automatically changes these attributes.

## Preparing SVN Documents for Import

Before importing SVN documents into the Ixiasoft CCMS, you must remove keyrefs and create an import map if necessary.

### Procedure

1. If needed, create a top-level import map for your documents.

You need to create a map in cases similar to the following:

- You are importing multiple books that do not belong to plugin files.
- You have multiple plugin files for your product documentation.

Include links to any required topics that are not included in other top-level maps. For example:

- Files in the **z-not-used** folder that you need to import can be linked directly from the import map.
- Some projects have other maps in lower level folders that are built separately. For example, in Linux, there are maps to generate placeholder documents. You can add these to the import map to avoid doing a separate import.

However, these files must have unique names; they cannot have the same names as the actual document files. If you use the same files, you should import the files once and recreate the second map in the Ixiasoft CCMS.

2. Add titles to all ditamaps that do not have them.

If you do not add titles before importing to the Ixiasoft CCMS, you will not be able to identify untitled maps in either the DITA Map view or the Search Results view.

3. Delete all **temp** folders from your project.

The **temp** folders contain copies of topics with the same topic ID as the original, which breaks the **key2href** script.

4. Open a command window at the top-level folder of your project and run **key2href**.

The script creates a new folder called **output**. If it already existed, the script deletes it. This means that build artifacts are removed before they can be imported into the Ixiasoft CCMS.

5. [Check the key2href results.](#)

6. Copy the **images** and **graphics** folders into the appropriate locations in the **output** folder.

7. Test your results by building one or more of the maps in the **output** folder.

#### Related Links

[Considerations for Importing Conditionalized Files](#) on page 13

Conditionalized documents in SVN include files not found in other documents. Special steps must be taken to address these files.

## Considerations for Importing Conditionalized Files

Conditionalized documents in SVN include files not found in other documents. Special steps must be taken to address these files.

When planning your import for document sets that contain multiple variants, consider the following:

- If you use the same topic in multiple maps, you must import all the associated maps at the same time. If you do not, multiple copies of the same topic are created in the Ixiasoft CCMS with unique topic IDs.
- If your document set uses **global-vars.ditamap**, be sure it is included. If the file is in the same folder as your plugin maps, add it to the import map.
- Some files used in SVN are not required in the Ixiasoft CCMS and therefore are not processed by the **key2href** script. For example, **.ditaval** and **local.properties** files are ignored. This means they are not copied to the **output** folder.

**NOTE:** In order to test your **key2href** output by building it, you must manually copy these files into the **output** folder and then remove them before importing to the Ixiasoft CCMS.

## Checking the key2href Results

The **log/key2href.log** file provides troubleshooting information.

#### Procedure

1. Check the script output status.

```
displaying errors, if any. Line numbers are from log/key2href.log:  
.end of errors
```

Whether or not any errors appear before **end of errors**, continue with the next step. Some errors are missed by the script output that appear in the log.

2. Open **log/key2href.log**.

The last few lines of the file are similar to the following:

```
[xslt] - Map: wr_idp_programmers_guide\introduction\introduction.ditamap
[xslt]   - Link: c-idp-about-wr-idp.dita
[xslt]   - Link: c-idp-idp-user-roles.dita
[xslt]   - Link: c-idp-where-to-find-info.dita
[xslt]   - Link: t-idp-accessing-documentation.dita
[xslt] - Map: wr_idp_programmers_guide\mcafee-embedded-control\mcafee-embedded-
control.ditamap
[xslt] C:\svn\DIRA-trunk\toolkit\non-DITA-tools\bin\key2href\key2href.xsl:280:
Fatal
Error! A sequence of more than one item is not allowed as the first argument of
tokenize() ("wr_idp_programmers_guide\key-i...", "wr_idp_security_guide\key-
idps...")
[xslt] Failed to process C:\svn\DIRA-trunk\toolkit\non-DITA-tools\bin\key2href\
\dummy.xml
```

The final line indicates the failure; the previous line (wrapped in the output) includes the source of the problem:

```
Fatal Error! A sequence of more than one item is not allowed as the first argument
of tokenize()
```

This example shows the most common type of failure and indicates that two files in your project have the same ID.

3. Determine which file has a duplicate topic name.

Each map is processed as it is reached by the **key2href** script.

First the script prints a line with the name of the map. Then it processes each **<topicref>**. When it finds the associated file, it prints that topic's path relative to this map.

In the example output, the previous map linked to several topics.

```
[xslt] - Map: wr_idp_programmers_guide\introduction\introduction.ditamap
[xslt]   - Link: c-idp-about-wr-idp.dita
[xslt]   - Link: c-idp-idp-user-roles.dita
[xslt]   - Link: c-idp-where-to-find-info.dita
[xslt]   - Link: t-idp-accessing-documentation.dita
```

The next map is listed, but no associated topics.

```
[xslt] - Map: wr_idp_programmers_guide\mcafee-embedded-control\mcafee-embedded-
control.ditamap
```

Therefore, the duplicate is probably the key of the first **<topicref>** in that map.

The **key2href** script extracted the **@key** from the **@keyref** attribute and tried to find the name of the topic associated with that **@key**. In this case, it found two files. One of those is a topic in another book, but one is a topic that is in this book, probably in this folder.

4. Change the topic ID wherever it appears.

- in this topic
- in the **<topicref>** in the map
- in the **@keyref** in this book's key map
- in any other links in this book that reference this topic

To find any other files in the book that link to this topic, use oXygen's **Find/Replace in Files** command. Access the command from the **Find** menu or from the right-click menu in

the DITA Map Manager. It allows you to specify a folder and search all the files in that folder, as well as optionally searching all the files in its sub-folders.

5. If your **log/key2href.log** file indicates a different error, contact the Infrastructure Group for assistance.

## Importing Files

After you have prepared your documents for import, you use the Ixiasoft CCMS import tool to bring documents into the document base.

### Procedure

1. Log in to the Ixiasoft CCMS in the Ixiasoft CCMS Client.

2. Select **File > Import**.

The Import dialog box appears and displays folders on the left.

3. Open the **IXIASOFT CCMS** folder.

4. Select **Import Maps** and click **Next**.

The Import Maps dialog box appears.

5. Browse to your output folder and select the map to import.

This is the import map you created or the single top-level map in your project.

6. Select the language, image, and label options.

---

Default Language English

Image Type Screen Capture

Image Format Low Res

Labels Select an appropriate label for your project. If there is none, ask for help or create it yourself.

---

7. Click **Finish**.

The defaults in the remaining dialog panes are acceptable without change.

8. Test one or more bookmaps by generating output in the Ixiasoft CCMS.

The maps should build with no changes.

## Making Final Changes in the Ixiasoft CCMS

After you import your documents, you must add the plugin name to your Eclipse Help maps because the Ixiasoft CCMS cannot get the name from the folder name.

### Procedure

1. Add an `<osgiManifest>` element to each top-level Eclipse Help map.

Add the `<osgiManifest>` element at the end, just before the closing `<eclipse-plugin-wr>` element.

```
<osgiManifest id="com.windriver.ide.docFIXME">
    <manifestMeta/>
</osgiManifest>
</eclipse-plugin-wr>
```

2. Replace the `@id` attribute value with the correct name of the plugin folder.

This is the same as the filename of the plugin file in SVN, minus the `.ditamap` extension.

3. Test the plugin by building Wind River Help.

# 4

## *The Document Life Cycle*

[The Document Life Cycle 17](#)

[Cycle and Status Dependencies 18](#)

[Workflows 19](#)

[Daily Activities Checklist for Using DITA in the Ixiasoft CCMS 23](#)

### **The Document Life Cycle**

The Ixiasoft CCMS provides a structured workflow for managing documentation projects from authoring through localization.

Writing structured topics and organizing them in maps allows you to reuse topics in new locations. It also makes keeping track of the status of each document more difficult. The Ixiasoft CCMS has cycles defined with relationships and prerequisites for the stages of each cycle.

The cycles Wind River has defined are:

#### Authoring Cycle

Each image, topic, and map goes through a cycle of creation, review, editing, and completion.

#### Published Cycle

When all the documents in a topic set are complete, the topic set is ready for publication. Publication labels all documents with a version label to show what release they were included in. The label also allows you to return to a particular state easily, even after some topics have changed.

#### Localization Cycle

The localization cycle allows you to track the status of each topic, map, and image through the translation process. It also allows you to track which items have changed, so only modified items must be retranslated.

Within the Authoring Cycle, various statuses are defined. A document must be in a specified status in order to proceed to the next one.

#### Work

The document is being worked on. A document can move to the Work status from any other status.

#### Review

The document has been submitted for review. A document moves from Work to Review status, and then back to Work.

#### Contribute

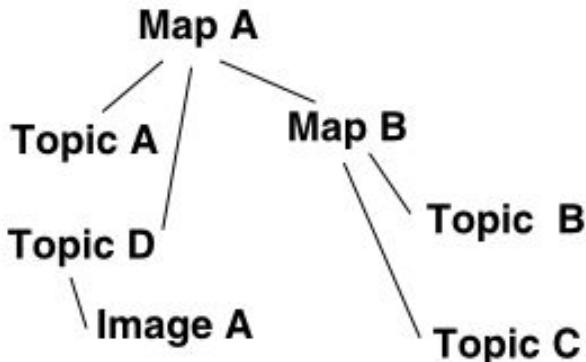
Input from another writer or subject matter expert is needed. A document moves from Work to Contribute status, and then back to Work.

#### Done

The document is ready for publication. A document moves to Done from Work.

## Cycle and Status Dependencies

Each document moves through defined states that have specified prerequisites. A document cannot move to the next status unless all its dependents have reached that status.



The example shows that Map A includes Topic A and Topic D, as well as Map B. Map B Includes Topic B and Topic C. Topic D includes Image A.

In order to move a topic or map to a more complete status, all topics, maps, and images it contains must be in that status. For example, in the image above, all maps, topics and images are in the Authoring:work status.

1. In order for Topic D to move to Authoring:done, Image A must first move to Authoring:done.
2. Map B cannot move to Authoring:done until both Topic B and Topic C are in Authoring:done.
3. Map A cannot move to Authoring:done until Topic A, Topic D, and Map B are all in Authoring:done.

4. Finally, Map A and the topics and images it contains cannot be moved to the Published cycle until Map A is in the Authoring:done status, which means that all the documents in Map A must be in Authoring:done as well.

New topics, maps, and images are created in Authoring:work. Topics, maps, and images imported from SVN are saved in the Ixiasoft CCMS in Authoring:work.

## Workflows

### Workflow: Creating New Product Documentation

Creating new product documentation consists of creating Eclipse Help maps (if necessary), a bookmap, the DITA maps that make up the chapters of the book, and the actual topics themselves.

The basic workflow is as follows:

1. [Create the Eclipse Help map.](#)

You use an Eclipse Help map to generate a Wind River Help (Eclipse) plugin from your document set. You only need this map if your documentation will become part of a Wind River Help plugin.

2. [Set up the Eclipse Help map.](#)

After you create an Eclipse Help map, you must customize it in order to generate combined PDF and HTML for your document set. Building this map also produces output that is suitable for a Wind River Help (Workbench) plugin.

3. [Create the Eclipse Help Submap.](#)

The Eclipse Help Submap is an intermediary map between your Eclipse Help map and the content of your project. It specifies all the books included in the plugin.

4. [Set up the Eclipse Help Submap](#) and add it to your Eclipse Help map.

After you create an Eclipse Help Submap, you must customize it to specify the appropriate anchor ID and product name and to reference all the documents in your document set. Editing the map in Text mode is typically the easiest way to set up your map.

5. Add the Eclipse Help Submap to your Eclipse Help map.

6. [Create your bookmap.](#)

The bookmap defines your book or document deliverable.

7. Set up your bookmap.

Setting up the bookmap consists of the following tasks:

- [adding product information](#)
- [adding taxonomy metadata](#)
- [changing the bundle name](#)
- [updating the metadata for inter-book linking](#)

- [adding an edition number \(optional\)](#)
8. [Create a DITA map](#) for each chapter in your book.  
The chapter `.ditamap` file defines the chapter structure.
  9. [Create your topics](#) and add them to the appropriate DITA map..
  10. Add content to your topics.
    - Add body (block) elements.
    - Add inline elements.
    - Add links.
    - [Add graphics](#).
    - [Add tables](#).
    - [Add an index](#).

## Workflow: Review Process

The process of submitting your documents for review depends on the type of review: a subject matter expert (SME) review or a peer review.

### SME Review

1. Generate output in one of the following formats:

Review Method	Output Type
Work In Progress (WIP) site	HTML and PDF
Product Documentation staging site	HTML
simple PDF review	PDF

2. Change the status of the topics or maps to Authoring:review.  
For more information, see [Changing File Status](#) on page 30.
3. After the review is complete, change the status of the topics or maps back to Authoring:work.
4. Add review comments to your topics.

### Peer Review

A peer review is a review of your documentation by another writer or information architect. The review is done in the Ixiasoft CCMS.

1. The writer changes the status of the topics or maps to Authoring:review.  
For more information, see [Changing File Status](#) on page 30.
2. The writer then assigns the topics or maps to the appropriate reviewer or information architect and notifies them that the content is ready for review.  
For more information, see [Assigning Files to Another Person](#) on page 31.
3. The reviewer or information architect reviews the topics or maps and adds input using the Track Changes feature.

4. Once the review is complete, the reviewer or information architect changes the status back to Authoring:work and notifies the writer that updates are available for review/acceptance.
5. The writer accepts or rejects the changes.

## Workflow: Finalizing your Documentation

After your content is complete, you must finalize the documentation, post it to the appropriate sites, and publish it in the Ixiasoft CCMS.

1. [Validate links](#) and fix any out-of-scope links.
2. Change the status of your images, topics, and maps to Authoring:done.  
For more information, see [Changing File Status](#) on page 30.
3. [Generate output](#).
4. Upload your document to the Product Documentation staging site and check the output.
5. Upload your document to the Product Documentation production site and check the output.
6. [Publish](#) your document in the Ixiasoft CCMS.

## Workflow: Creating a New Lab Module

Each lab module (that is, a chapter in a lab book) consists of a chapter ditamap and its topics (an *Overview* concept topic and nested child task topics).

When you want to add a new lab chapter to a customer training course book, you must create:

- a new chapter **.ditamap** file  
The chapter **.ditamap** file defines the chapter structure - the introductory overview and the tasks the student will perform.
- an *Overview* concept topic

The overview topic presents the lab objective, a note that indicates the typical time required to complete the lab, and any additional background information the student needs to know (such as prerequisites, technical background, special technical details specific to the lab environment, or a preview list of the high level tasks the student will perform).

- a set of task topics (the sections within the lab)

Each task topic describes the steps a student must perform to complete a high level task.

The workflow is:

1. [Create your bookmap](#).  
The bookmap defines your book or document deliverable.
2. [Create a DITA map](#) for each chapter in your book.  
The chapter **.ditamap** file defines the chapter structure - the introductory overview and the tasks the student will perform.
3. [Create your topics](#) and add them to the appropriate DITA map.  
You must create the following:

- An *Overview* topic that presents the lab objective, a note that indicates the typical time required to complete the lab, and any additional background information the student needs to know (such as prerequisites, technical background, special technical details specific to the lab environment, or a preview list of the high level tasks the student will perform).
  - A set of task topics (the sections within the lab). Each task topic describes the steps a student must perform to complete a high level task.
4. Add content to your topics.
    - Add body (block) elements.
    - Add inline elements.
    - Add links.
    - Add graphics.
    - Add tables.
  5. Conduct an editorial and DITA code review, and resolve all changes and issues.
  6. Generate draft output for technical review and testing.
    - Move topics to the Authoring:review state.
    - Generate a draft PDF for review.
    - Put each lab chapter PDF in the **/Lab\_docs** folder of the appropriate module in the **Learning Modules** repository.
  7. Incorporate technical review and test comments in your topics.
  8. Finalize your content.
  9. Generate output:
    - Generate a chapter PDF from each chapter ditamap.

Put each lab chapter PDF in the **/Lab\_docs** folder of the appropriate module in the **Learning Modules** repository.
    - Generate the lab book from the bookmap.
    - Use the ats-bb tool to build a lab book, then compare the ats-bb lab book to the Ixiasoft CCMS-generated lab book, and verify that the substantial content is the same (expect non-substantive differences, like formatting).

If there are substantive differences, resolve them.
  10. Release the lab book, in the appropriate release repository on the WRU SVN server.
  11. [Publish](#) your document in the Ixiasoft CCMS.

## Workflow: Creating a New Lecture Module (Placeholder)

Concept definition.

## Daily Activities Checklist for Using DITA in the Ixiasoft CCMS

The content management system is a repository that provides tools for creating DITA topics, combining them into documents, managing revisions, and publishing and archiving.

Unlike the SVN repository we used before, the Ixiasoft CCMS does not have a local repository. This simplifies your interaction with the repository. However, it means that if you leave a topic locked (checked out), no one else can make changes to that topic.

1. Log into the DITA document base.
2. Search for existing maps and topics you want to work with.
3. Lock any file you want to change. You can lock multiple files at one time.
4. Open each file in the editor and make your changes. Add a note to the Modification History in the topic or map for significant changes.
5. Release the lock on the file. You must provide a comment. You can release multiple files with the same comment in the same action.
6. Create new topics and maps. New topics and maps are locked by you when you create them.
7. Add content to the topics and maps and add the new topics to a map.
8. Release the lock on the new files.

In the Release dialog box, you can select **Release and keep locked**. This capability can be useful if you want to release content that you are still working on to see how it builds; it saves releasing and then locking again. Generating output operates only on released documents.

9. Validate links.

The Validate Links tool locates any links whose target is outside the scope of the current map. It is best practice to validate links on a regular basis.

10. Generate output and review error logs.

It is best practice to generate output and review the error logs regularly. Doing so makes troubleshooting issues much easier. You can concentrate on just those files that have changed since the last successful build.

11. Publish documents at release time.



# 5

## *File Management*

<a href="#">About Files in the Ixiasoft CCMS</a>	25
<a href="#">Locking Files for Editing</a>	26
<a href="#">Change Tracking</a>	26
<a href="#">Releasing Files to the Database</a>	28
<a href="#">Changing File Status</a>	30
<a href="#">Assigning Files to Another Person</a>	31
<a href="#">Viewing Files in the Todo List</a>	31
<a href="#">Deleting Files</a>	33
<a href="#">Restoring Files</a>	34
<a href="#">Running the Validate Links Tool</a>	34
<a href="#">Product Documentation Release Checklist</a>	36

### **About Files in the Ixiasoft CCMS**

Each map, topic, image, and resource in the Ixiasoft CCMS exists as a file.

You can use the term *file* to refer collectively to multiple items of different types in the Ixiasoft CCMS.

Each file has a unique name, which is generated by the Ixiasoft CCMS and which is the same as the file ID. However, in working with files, you will use the title to identify it rather than the file name or ID. The Ixiasoft CCMS tracks the ID for you, which allows you to change the title freely as necessary.

## Locking Files for Editing

In order to make any change to a file, you must lock the file first. When the file is locked, no one else can make changes to that file.

Actions that require a file to be locked include:

- editing text in the file
- adding a topic to a map file (which modifies text in the map)
- adding a submap to a map file (which modifies text in the map)

### Procedure

1. Locate the correct file in the DITA Map view or by searching.
2. Right-click on the file and select **Lock** from the drop-down list.

The file name appears in bold. When the file is selected, the text is black. When it is not selected, the text is blue.

## Change Tracking

### Change Tracking Overview

The Track Changes feature allows you to keep track of the changes you or reviewers make to a document. You can then choose to accept or reject those changes.

You should use the Track Changes feature in the following situations:

- when making any significant updates to an existing topic; such adding new paragraphs or steps
- when making any significant updates to an existing map
- when creating new content from external sources
- during reorganization efforts, especially where content is reworked into more than one book or additional topics

You can access the change tracking options by selecting **Edit > Review**, or by using the change tracking toolbar.

The change tracking toolbar is located at the top of the Editor view:



The options are as follows:



### Track Changes

Allows you to toggle change tracking mode on or off. When change tracking is on, a blue square appears around the icon:



### Accept Changes

This option is only available when change tracking has been enabled. It accepts the change located at the cursor position.



### Reject Changes

This option is only available when change tracking has been enabled. It rejects the change located at the cursor position.



### Comment Change

This option is only available when change tracking has been enabled. It allows you to add a comment to a change.



### Track Changes Drop-Down List

Allows you to switch between the following modes:

- **View All Changes/Comments** - This is the default mode. It shows all tracked changes.
- **View Only Changes/Comments By** - This mode allows you to limit the tracked changes shown to those that were made by a particular person.
- **View Final** - This mode provides a preview of what the topic would look like if all the changes are accepted.
- **View Original** - This mode provides a preview of what the topic would look like if all the changes are rejected.



### Highlight

Allows you to highlight text or remove highlights from text. The drop-down list allows you to select the color of the highlight. If you select **No color (erase)** you can erase a highlight.



### Add Comment

Allows you to add a comment for a selected fragment of text. After you add the comment, the fragment of text is highlighted in the document and you can view the comment by hovering over the highlighted content.



### Edit Comment

Allows you to edit a comment.



### Remove Comment

Allows you to remove a comment.



### Manage Reviews

Allows you to view all the comments for a topic in the Review view. When you click on an item in the Review view, the cursor is inserted where the change was made in the topic. You can accept or reject the changes for each item in the list.

---

## Enabling Change Tracking

You can enable change tracking to keep track of the changes you or reviewers make to a document.

### Procedure

1. Lock and open your topic or map.
2. Select **Edit > Review > Track Changes**.

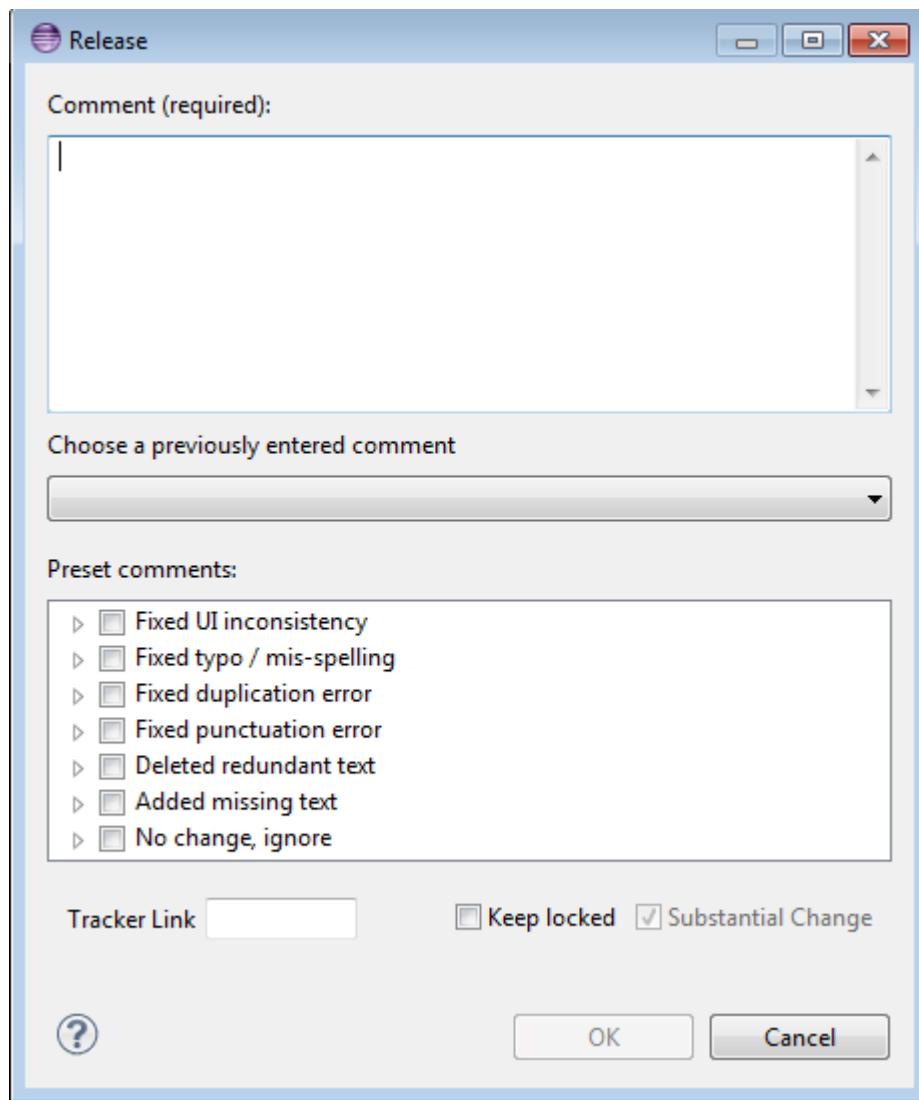
## Releasing Files to the Database

Until you release the lock on a file, the changes you make are only local. When you release a file, the changes are saved to the server and the file is freed so other users can edit it.

### Procedure

1. Right-click the file and select **Release**.

The Release dialog box opens.



**2. Enter a comment.**

A comment is required. You can type a new comment, select a previously entered comment, or select one of the preset comments. If your change is small, clear the **Substantial Change** selection if it is active.

**3. Click OK.**

The file is updated in the Ixiasoft CCMS database. Also, the file name no longer appears in bold blue or black text in the DITA Map or Search Results views.

## Changing File Status

The file status allows you to track the progress of your document. Specific statuses are required for certain actions.

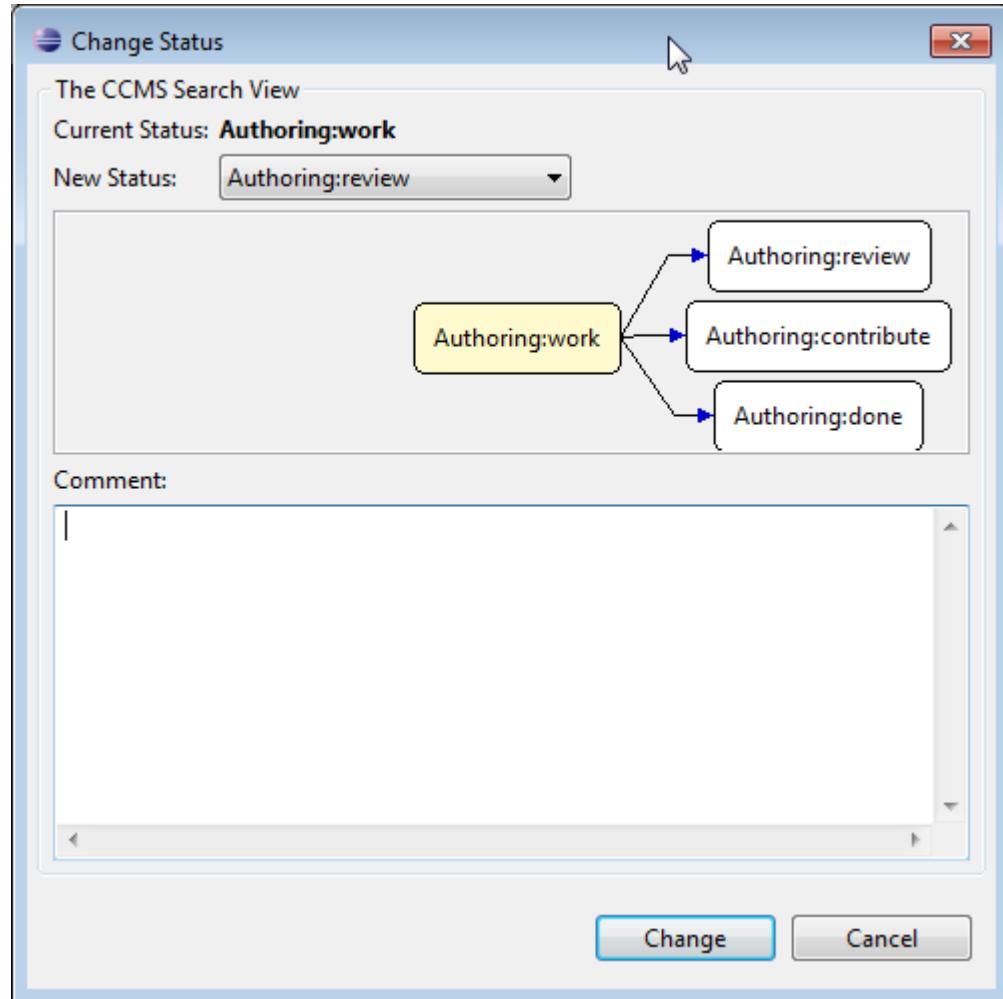
The Ixiasoft CCMS enforces a workflow that Information Development has established. For more information, see:

- [The Document Life Cycle](#) on page 17
- [Cycle and Status Dependencies](#) on page 18

### Procedure

1. Locate and select one or more files.
2. Right-click and select **Change Status**.

The Change Status dialog box appears.



3. Select the new status.

You can use the drop-down menu or click the status in the flow chart.

4. Click **Change**.

---

**NOTE:** A comment is not currently required.

---

If successful:

All file statuses are updated.

If not successful:

The error window shows why the status could not be updated. If any of the files you selected are not in the correct status, you cannot move them to the next status.

## Assigning Files to Another Person

You must assign files to others as your file progresses through the different phases of the document life cycle. For example, when a topic is ready for a peer review, it is assigned to a reviewer.

### Procedure

1. Perform a search to find the files you want to reassign.
2. Right-click the file and select **Assign to**.
3. In the Assignments dialog box, use the Filter section to list assignees by project or by group.  
Assignees are listed under the roles to which they have been assigned. If you are listing assignees by project, you can limit the scope to a particular project or to just your projects.
4. Select the appropriate assignee and click **OK**.

## Viewing Files in the Todo List

### The Todo List View

The Todo List view allows you to view the files that are assigned to you. It also displays the due dates for the file and other people who are assigned to it (such as a reviewer or information architect).

You can limit the display of the Todo List to view files that are at specific points on your timeline. These points are: Incoming, Active, and Retained.

### Incoming

Indicates that this file will be assigned to you to work on at some point in the documentation life cycle. For example, you are assigned as the Information Architect for a map and will be reviewing it after it has been created by the writer. If the writer is still working on creating the map, it will show up in your Incoming list.

### Active

Indicates that you are actively working on a file.

### Retained

Indicates that someone else is currently assigned to the file, but at some point it will be assigned back to you. For example, you have finished work on a file and assigned it to a reviewer. While the reviewer is actively reviewing the file, it will appear in your Retained list. Once the review is finished, the file will be assigned back to you. It will then, again, show up in your Active list.

Since the items in the Todo List view depend on the status of the file and the file assignment, it is important for you to consistently move the file to the appropriate status during the documentation life cycle and make the proper assignments. For more information, see:

- [The Document Life Cycle](#) on page 17
- [Changing File Status](#) on page 30
- [Assigning Files to Another Person](#) on page 31

Once you have completed work on a file, if you change the status to Authoring:done, it will drop off your Todo List.



**NOTE:** The Todo List view does not refresh automatically. To refresh your list of assignments, click the blue Refresh Todo List icon.

---

If you are a project manager, you can view the Todo List of all the team members of the projects that you manage. This allows you to view the status of the documents and manage workloads.

## Opening the Todo List View

If the Todo List view is not already open, you must open it to view the files that are assigned to you.

### Procedure

1. Select **Window > Show View > Other**.
2. Expand **IXIASOFT CCMS - General** and select **Todo List**.

## Grouping Items in the Todo List View

You can group the items listed in the Todo Listview to better manage the items to which you are assigned.

### Procedure

Select one of the following options from the drop-down list on the upper-right corner of the Todo List view.

Option	Description
<b>None</b>	Displays an alphabetical list of all the documents assigned to you.
<b>Page</b>	Groups items by page.
<b>Role</b>	Groups items according to role; for example, writer, reviewer, information architect, and so forth.
<b>Timeline</b>	Groups items according to their position on the timeline; for example, Incoming, Active, and Retained.
<b>Status</b>	Groups items according to their status; for example, Authoring:work or Authoring:review.
<b>Due Date</b>	Groups items according to their due date.
<b>Object Type</b>	Groups items according to their type; for example, concept-wr, task-wr, bookmap-wr, and so forth.

## Deleting Files

The Ixiasoft CCMS maintains the files you are working on in an active area of the repository. The delete function allows you to remove unnecessary files (maps, topics, images, and so forth) from the active area, without removing them completely from the Ixiasoft CCMS.

### Prerequisites

- The file must be unlocked.
- The file cannot be referenced by another active map or topic. Use the Dependencies view to locate topics or maps that reference the file.

### Procedure

1. Do a search for your file.
2. Right-click the file and select **Delete**.



---

**NOTE:** The delete function is not available in the DITA Map view.

---

3. Click **Yes**.

The file is removed from the active area of the repository.

If you accidentally deleted a file or wish to restore a deleted file, use the restore function.

## Restoring Files

The restore function allows you recover files that you may have accidentally deleted from the active area of the Ixiasoft CCMS repository.

### Procedure

1. Do a search for your file, with **Deleted** selected in the Cycles pane of the Search view.
2. Right-click the file and select **Restore**.

The file is restored to the active area of the Ixiasoft CCMS repository.

## Running the Validate Links Tool

The Validate Links tool locates any links whose target is outside the scope of the current map. It is best practice to validate links on a regular basis.

You can validate links at any level of your topic set hierarchy, from the topic to the Eclipse Help map. Most commonly you will validate links in a bookmap or an Eclipse Help map.



---

**NOTE:** You cannot publish a document that contains invalid links.

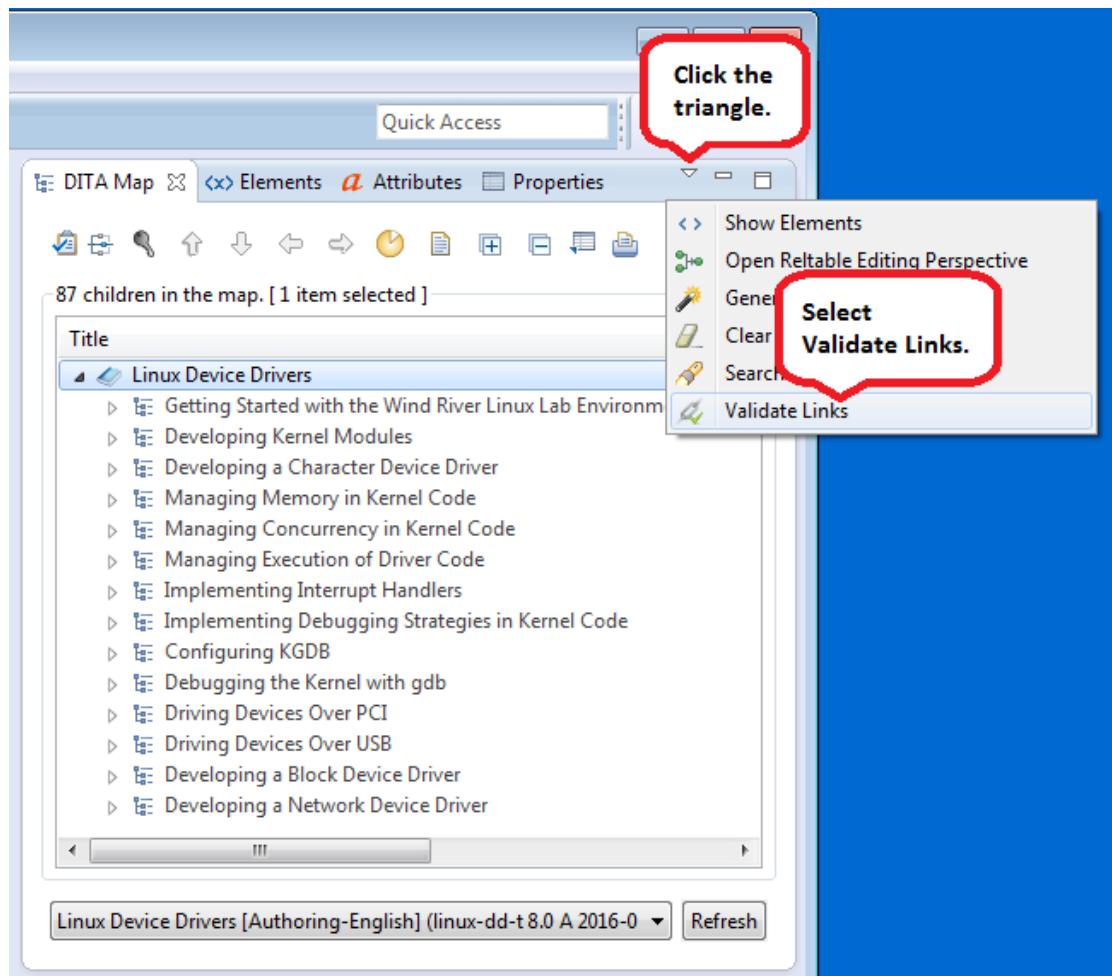
---

### Procedure

1. Open the map you want to publish in the DITA Maps view.
2. Click on the top-level document you wish to validate.

For product documentation this is either an Eclipse Help map or a bookmap. For training documents this is usually a bookmap.

- Click on the triangular menu icon at the upper right of the view and select **Validate Links** from the drop-down list.



If the validation succeeds, a confirmation window appears.

If the validation fails, an error window appears.

Detailed Information		
Level	Title	Description
⚠	Building Advantech UTX-3115 Boards [rec1394031140025]	This reference [/content/authoring/rec1394031140025.xml] is out of the map scope.
⚠	Deploying Quark Boards Using a Script [dia1394827580320]	This reference [/content/authoring/dia1394827580320.dita] is out of the map scope.
⚠	Building Platform Projects for Quark Boards [rec1394031139822]	This reference [/content/authoring/rec1394031139822.xml] is out of the map scope.
⚠	Deploying Advantech UTX-3115 Boards Using a Script [rec1394031147583]	This reference [/content/authoring/rec1394031147583.xml] is out of the map scope.
⚠	Performing a Secure Boot on Cross Hill and Clanton Hill Boards [rec1394031147896]	This reference [/content/authoring/rec1394031147896.xml] is out of the map scope.
⚠	Deploying Advantech UTX-3115 Boards Manually [rec1394031147740]	This reference [/content/authoring/rec1394031147740.xml] is out of the map scope.
⚠	Deploying Quark Boards Using a vfat-Formatted USB Drive [dia1394827472515]	This reference [/content/authoring/dia1394827472515.dita] is out of the map scope.
⚠	Performing a Verified Boot [rec1394031148099]	This reference [/content/authoring/rec1394031148099.xml] is out of the map scope.
⚠	McAfee Embedded Control [rec1394031140369]	This reference [/content/authoring/rec1394031140369.xml] is out of the map scope.
⚠	Setting Up a Zigbee Network [rec1394031143720]	This reference [/content/authoring/rec1394031143720.xml] is out of the map scope.
⚠	Performing a Secure Boot on Advantech UTX-3115 Using UEFI [dia1395696238114]	This reference [/content/authoring/dia1395696238114.dita] is out of the map scope.
⚠	Preparing USB Boot Media [rec1394031147286]	This reference [/content/authoring/rec1394031147286.xml] is out of the map scope.

## Postrequisites

If the validation fails, you must find and fix the out-of-scope links. For more information, see [Finding Out-of-Scope Links](#) on page 297.

# Product Documentation Release Checklist

Maintaining a checklist helps you ensure your documentation is complete before you release it to customers and that it is archived appropriately before work is started on a new version of the document.

Done	Checklist Item
<input type="checkbox"/>	Have all review comments been incorporated into the topics?
<input type="checkbox"/>	If the document is being authored by multiple writers, is all work is finished?
<input type="checkbox"/>	Is the taxonomy in the bookmap file correct? Taxonomy metadata is necessary for classification and search on the Product Documentation site. You need to add terms from the following taxonomies: <ul style="list-style-type: none"><li>• Products</li><li>• Category</li><li>• Content Type</li><li>• Format</li><li>• Visibility</li></ul> For more information about adding taxonomies to your documentation, see <a href="#">Adding Taxonomy Metadata to a Bookmap</a> on page 99.
<input type="checkbox"/>	Are the values of the <mainbooktitle> and <vrm> elements correct?
<input type="checkbox"/>	Have <othermeta> elements to change the bundle name and ensure proper inter-book linking been added to your bookmap? For example, the following metadata sets the bundle name for the <i>Wind River Linux Getting Started, LTS 18</i> : <pre>&lt;othermeta name="bundle" content="wind_river_linux_getting_started_lts_18"/&gt;</pre> For more information about bundle names, see <a href="#">The Bundle Name</a> on page 100. The following example sets the product version to Linux LTS 18 to ensure that inter-book links go to the proper version of the document: <pre>&lt;othermeta name="facets" content="version=os_linux_lts_18"/&gt;</pre> For more information about metadata for inter-book linking, see <a href="#">Updating the Metadata for Inter-Book Linking</a> on page 105
<input type="checkbox"/>	Do all your <xref> elements contain appropriate @scope attributes and link text?

Done	Checklist Item
	<p>For more information, see <a href="#">Updating @scope Attributes in Cross-References Using an XSLT Script</a> on page 199.</p>
<input type="checkbox"/>	<p>Have all your CID and content-lookup links been updated to the current format? CID links are links to a particular product page or category on the Product Documentation site. Content-lookup links are links to the "top" page of a document or video on the Product Documentation site. The previous format used does not work for the current version of the Product Documentation site.</p> <p>For more information, see:</p> <ul style="list-style-type: none"> <li>• <a href="#">Creating Links to Hierarchy Locations on the Product Documentation Site</a> on page 206</li> <li>• <a href="#">Creating Links to Documents on the Product Documentation Site</a> on page 205</li> </ul>
<input type="checkbox"/>	<p>Remove all items using the <code>&lt;draft-comment&gt;</code> element and any unnecessary XML comments.</p>
<input type="checkbox"/>	<p>Validate links at the bookmap level and fix any out-of-scope link issues. You used to validate links at the Eclipse Help map level. However, unless you are producing documentation for Eclipse Help, you must now validate links at the bookmap level.</p> <p>For more information, see <a href="#">Running the Validate Links Tool</a> on page 34.</p>
<input type="checkbox"/>	<p>Change the status of your images, topics, maps, and resource files to the Authoring:done state.</p>
<input type="checkbox"/>	<p>Generate output using the <b>Zoomin-SFTP-staging</b> transform.</p>
<input type="checkbox"/>	<p>Check the output on the Product Documentation staging site:</p> <ul style="list-style-type: none"> <li>• Does the document show up in the appropriate category?</li> <li>• Are all chapters/topics present?</li> <li>• Is the style/formatting correct?</li> <li>• Do your links work?</li> </ul> <p>If there are errors, correct them, regenerate, and check again on the staging site.</p>
<input type="checkbox"/>	<p>Generate output using the <b>Zoomin-SFTP-production</b> transform.</p>
<input type="checkbox"/>	<p>Check the output on the Product Documentation production site:</p> <ul style="list-style-type: none"> <li>• Does the document show up in the appropriate category?</li> <li>• Are all chapter/topics present?</li> <li>• Is the style/formatting correct?</li> <li>• Do your links work?</li> </ul> <p>If there are errors, correct them, regenerate, and check again on both the staging site and production site.</p>
<input type="checkbox"/>	<p>Publish your bookmap.</p>



# 6

## *Searching in the Ixiasoft CCMS*

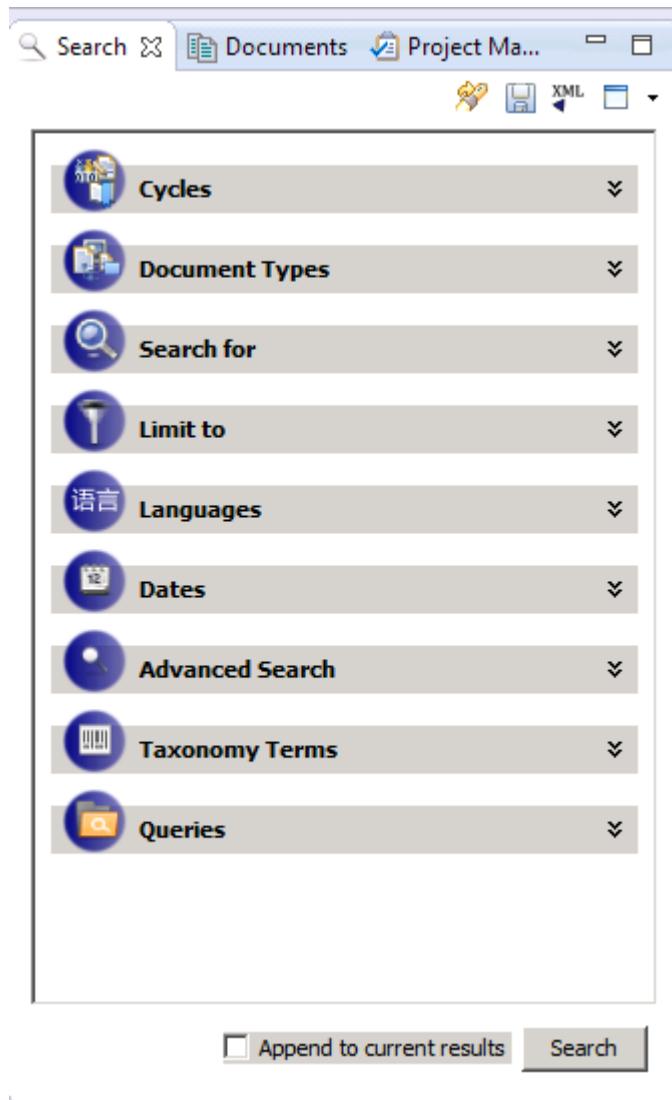
<b>The Ixiasoft CCMS Search View</b>	<b>40</b>
<b>The Search Results View</b>	<b>49</b>
<b>Searching Using a Search String</b>	<b>50</b>
<b>Searching by Locked Status</b>	<b>51</b>
<b>Searching by Date</b>	<b>52</b>
<b>Searching by Label</b>	<b>54</b>
<b>Searching by Document ID</b>	<b>54</b>
<b>Searching for All References to a Specific Topic</b>	<b>55</b>
<b>Searching for Topics that Contain a Specific Element</b>	<b>55</b>
<b>Searching for Topics that Contain an Element with Particular Text</b>	<b>55</b>
<b>Using Search to Prepare Your Book for Publishing</b>	<b>56</b>
<b>Filtering Search Results Based on Text in the Title</b>	<b>57</b>
<b>Adding Columns to the Search Results View</b>	<b>57</b>
<b>Saving Searches</b>	<b>58</b>
<b>Searching Using the AND Operator</b>	<b>59</b>
<b>Searching Using the OR Operator</b>	<b>60</b>
<b>Making Bulk Changes to XML</b>	<b>60</b>

## The Ixiasoft CCMS Search View

The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

By default, the Search view appears as a tab on the left side of the DITA perspective.

The Search view is divided into several panes. Each pane lets you specify search criteria. You can expand or collapse the panes to hide the ones you are not currently using.



You can add search results to the current search by selecting **Append to current results** at the bottom of the view.

#### Related Links

[The Cycles Pane](#) on page 41

The Cycles pane allows you to limit your search to documents in a particular cycle in the workflow.

[The Document Types Pane](#) on page 42

The Document Types pane allows you to limit your search to one or more document types.

[The Search For Pane](#) on page 43

The Search for pane allows you to limit your search to documents containing a particular string.

[The Limit To Pane](#) on page 44

The Limit to pane allows you to limit your search in various ways. Particularly useful are files that are locked by you or others and dependencies of the current DITA Map view.

[The Languages Pane](#) on page 45

The languages pane allows you to search for documents authored in different languages.

[The Dates Pane](#) on page 46

The Dates pane allows you to search for documents created, modified, or locked on a specific date or during a range of dates.

[The Advanced Search Pane](#) on page 46

The Advanced Search pane allows you to specify complex searches using **and** and **or** operators.

[The Taxonomy Terms Pane](#) on page 48

The Taxonomy Terms pane allows you to search for items containing specific taxonomy terms.

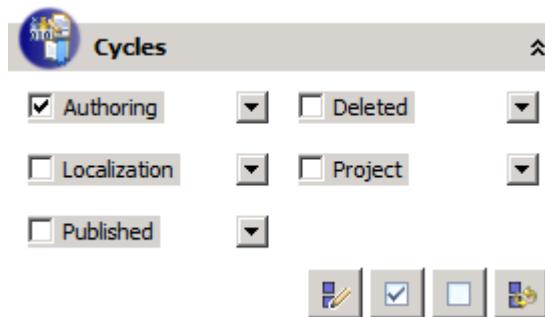
[The Queries Pane](#) on page 48

The Queries pane records your search history and other information so you can repeat a specific search.

## The Cycles Pane

The Cycles pane allows you to limit your search to documents in a particular cycle in the workflow.

By default, all the states in the cycle selected are searched. For example, if you select **Authoring** and click the drop-down arrow to the right of **Authoring**, you will see that all states in the Authoring cycle are selected (such as, Authoring:work, Authoring:done, and so forth).



The buttons at the bottom of the pane allow you to do the following:

Action	Button
Set current settings as default settings for the pane.	
Select all options.	
Clear all options.	
Reset to default settings.	

#### Related Links

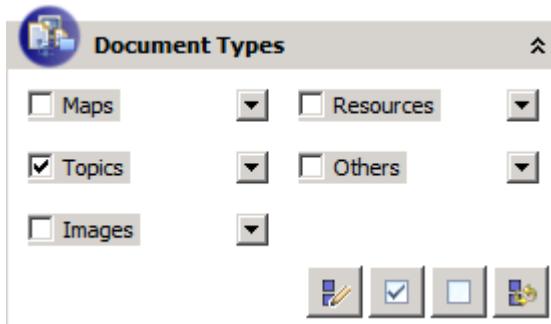
[The Ixiasoft CCMS Search View](#) on page 40

The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

### The Document Types Pane

The Document Types pane allows you to limit your search to one or more document types.

By default, all subtypes for the selected document type will be searched. For example, if you select **Topics** and click the drop-down arrow to the right of **Topics**, you will see that all topic types are selected (such as, concept-wr, task-wr, and so forth).



The buttons at the bottom of the pane allow you to do the following:

Action	Button
Set current settings as default settings for the pane.	
Select all options.	
Clear all options.	

Action	Button
Reset to default settings.	

### Related Links

[The Ixiasoft CCMS Search View](#) on page 40

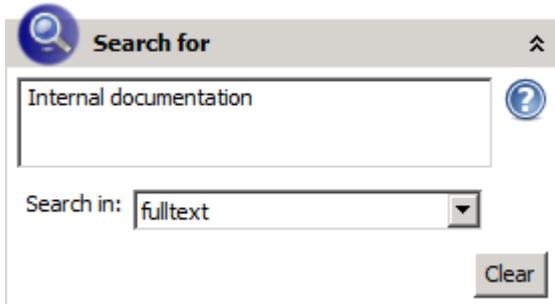
The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

### The Search For Pane

The Search for pane allows you to limit your search to documents containing a particular string.

By default, the full text of the document will be searched for the specified string. You can also search a specific element, index, or attribute by selecting a different value from the **Search in** drop-down list.

Search is not case-sensitive. You can use wildcards to refine your search by clicking the question mark button for a list of search operators and their descriptions.



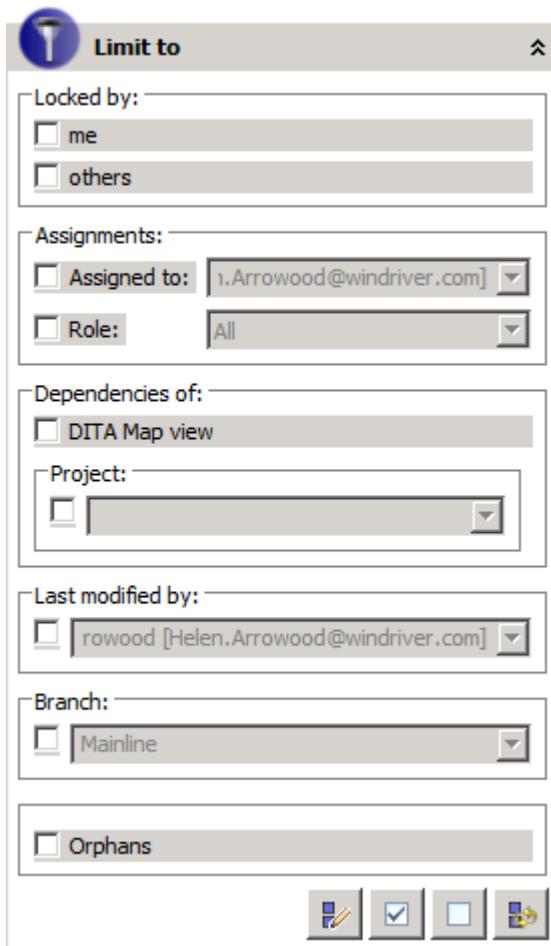
### Related Links

[The Ixiasoft CCMS Search View](#) on page 40

The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

### The Limit To Pane

The Limit to pane allows you to limit your search in various ways. Particularly useful are files that are locked by you or others and dependencies of the current DITA Map view.



#### Related Links

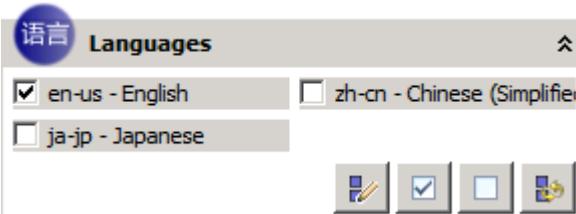
[The Ixiasoft CCMS Search View](#) on page 40

The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

#### The Languages Pane

The languages pane allows you to search for documents authored in different languages.

You will usually leave English as the only language checked.



#### Related Links

[The Ixiasoft CCMS Search View](#) on page 40

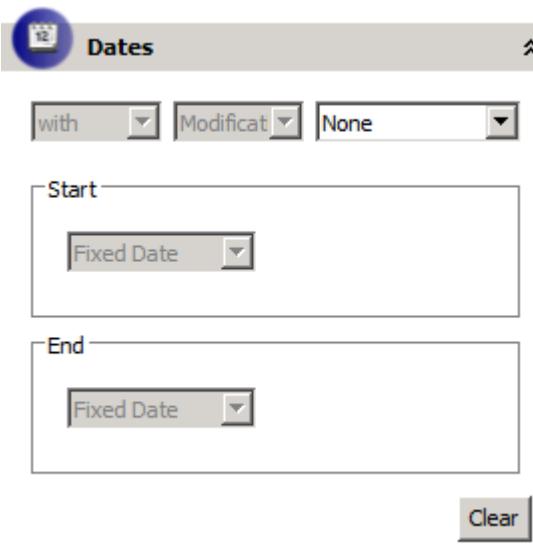
The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

#### The Dates Pane

The Dates pane allows you to search for documents created, modified, or locked on a specific date or during a range of dates.

You can also search for all documents that were *not* created on a specific date.

Click **Clear** to clear the search.



#### Related Links

[The Ixiasoft CCMS Search View](#) on page 40

The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

#### The Advanced Search Pane

The Advanced Search pane allows you to specify complex searches using **and** and **or** operators.

The **Operator**, **Type**, **Name**, and **Value** fields provide drop-down lists of available choices.



**NOTE:** You must first click in the column before the drop-down list appears.

Operator	Type	Name	Value
and			

Clear

The first column specifies a search operator. By default, the first row always starts with the **and** operator.

On subsequent rows, you can select one of the following from the **Operator** drop-down list:

- **and** - the search result must contain the specified value
- **or** - the search result may contain the specified value
- **without** - the search result must not contain the specified value

The buttons at the bottom of the pane allow you to move, add, or delete rows in the search table.

---

→ **NOTE:** Usually you will use the Advanced Search pane in conjunction with other filters. Be sure to clear any unwanted filters before searching.

---

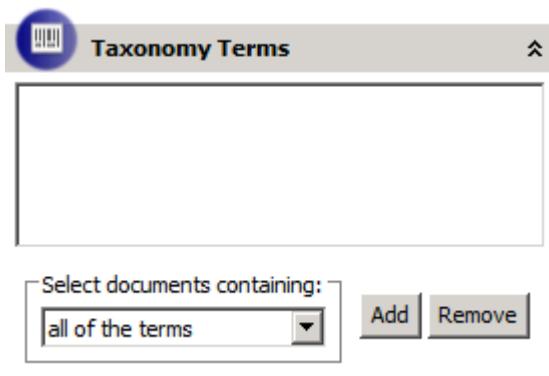
#### Related Links

[The Ixiasoft CCMS Search View](#) on page 40

The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

## The Taxonomy Terms Pane

The Taxonomy Terms pane allows you to search for items containing specific taxonomy terms.



Clicking **Add** allows you to select one or more taxonomy terms from a tree view.

You can search for topics that contain all the terms or topics that contain any of the terms.

#### Related Links

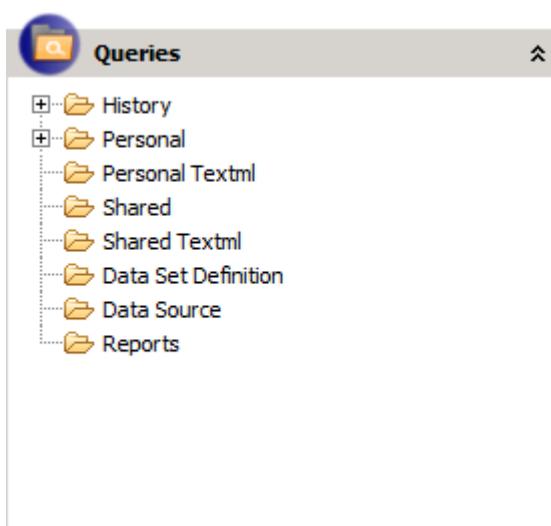
[The Ixiasoft CCMS Search View](#) on page 40

The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

## The Queries Pane

The Queries pane records your search history and other information so you can repeat a specific search.

You can also save custom queries and share them with other Ixiasoft CCMS users.



### Related Links

[The Ixiasoft CCMS Search View](#) on page 40

The Search view is the primary tool used to locate items in the Ixiasoft CCMS. The Search view provides multiple filters to help you locate the correct items.

## The Search Results View

When you search for something in the Search view, the results appear in the Search Results view.

By default, the Search Results view opens below the editor.

Title	Type	Id	Status	Version
Diagnosing Initial Connectivity Failures on Windows	task-wr	qti1476302478948	Authoring:work	HDC 2.1 2016-11-07 (October UI GA)
Retrieving Files from Devices	task-wr	icw1483479499991	Authoring:work	HDC 2.2 2017-02-10 (HDC_DEV_R2 C)
Updating the Agent on Windows from Release 2.2 to 2.3.0	task-wr	ntn1488462924669	Authoring:work	HDC 2.2 2017-03-08 (HDC_DEV_R4 A)
Updating the Agent on Wind River Linux and IDP XT from Release 2.1	task-wr	sis1488567594759	Authoring:work	HDC 2.2 2017-03-08 (HDC_DEV_R4 A)
Agent Changes in this Release	concept-wr	jke1459341557003	Authoring:work	HDC 2.0 2016-04-06; HDC 2.0.1 2016
iot.cfg	reference-wr	qel1480972751049	Authoring:work	HDC 2.2 2016-12-13 (GA); HDC 2.2

The buttons and drop-down lists on the upper right of the Search Results view provide additional options for refining and working with your search results.

Search Results Button	Description
	Clear all items from the search results list.

Search Results Button	Description
	Export the search results list and save it in a file outside the repository. You can save the results in the following formats:
<ul style="list-style-type: none"><li>• .tsv</li><li>• .xml</li><li>• .xls</li></ul>	
	Search the content of documents listed in the search results list, including XML elements and attributes.
	Create and sort by groupings. Groupings allow you to organize your search results.
	For more information about using groupings, see the <i>CCMS Desktop User Guide</i> located online at:
	<a href="https://www.ixiasoft.com/documentation/IXIASOFT_CCMS/6.0/User_Guides_Advanced_User_Standard/en/per1389985945656.html">https://www.ixiasoft.com/documentation/IXIASOFT_CCMS/6.0/User_Guides_Advanced_User_Standard/en/per1389985945656.html</a>
	Enable or disable thumbnails display.

## Searching Using a Search String

You can use the Search for pane to limit your search to items that contain a specific string.

### Procedure

1. In the Document Types pane, select the desired document type.
2. In the Search for pane, select **fulltext** from the **Search in** drop-down list.
3. Type the desired search string in the **Search for** text box.

Search is not case-sensitive. You can use wildcards to refine your search by clicking the question mark button for a list of search operators and their descriptions.



**NOTE:** If you search for a string with spaces (for example, controller configuration), you may not get all results. This is because the XML may contain line breaks and multiple spaces, which are not recognized by the search. To ensure that you find all occurrences, use a regular expression; for example:

controller\\n\* \*configuration

This finds the word controller, followed by newline (0 or more), followed by space (0 or more).

---

4. Click **Search**.

The search results appear in the Search Results view.



**NOTE:** The string does not always appear in the title.

---

## Searching by Locked Status

You can use the Limit to pane to limit your search to items that are locked by you or by others.

### Procedure

1. In the Document Types pane, select the desired document type.
2. In the Limit to pane, select the desired option in the Locked by section.

Option	Description
me	Limits search results to items locked by you.
others	Limits search results to items locked by others.

3. Click **Search**.

The search results appear in the Search Results view. In this example, the results include all topics that are currently locked by me.

Results List page 1 of 1 (3 results found)	
Title	Status
Searching for Images (placeholder)	Authoring:work
Searching for Topics (placeholder)	Authoring:work
Searching for Maps	Authoring:work

## Searching by Date

You can use the Dates pane to limit your search to items created, locked, or modified on a particular date or during a date range. You can also search for items that were *not* created, locked, or modified on a certain date or during a date range.

### Procedure

1. In the Document Types pane, select the desired document type.
2. In the Dates pane, select an option from the far right drop-down list.



**NOTE:** To search by date, you must select an item from this drop-down list first. If **None** is selected, all other options are grayed out.

Option	Description
On Date	Searches for items created, locked or modified on a particular date.
Before Date	Searches for items created, locked or modified before a particular date.
Since Date	Searches for items created, locked or modified after a particular date.
Between Dates	Searches for items created, locked or modified during a date range.

3. Select one of the following options from the first drop-down list:

Option	Description
<b>with</b>	Limits search to items created, locked, or modified on a particular date or during a date range.
<b>without</b>	Limits search to items <i>not</i> created, locked, or modified on a particular date or during a date range.

4. Select the desired type of date from the second drop-down list:

- **Modification Date**
- **Locked Date**
- **Creation Date**

5. Select whether to search by fixed date or relative date.

Option	Description
<b>Fixed Date</b>	Searches for items created, locked, or modified on an exact date; for example, 4/28/2017.
<b>Relative Date</b>	Searches for items created, locked, or modified on a day relative to today; for example, yesterday or last week.

6. If you selected **Fixed Date**, select the date.

7. If you selected **Relative Date**, select an option from the drop-down list on the far right of the Start section.

Select **custom** to create a custom relative date.

8. Select if your date is inclusive or exclusive.



**NOTE:** This step does not apply if you selected **On Date** in Step 2.

9. If you selected **Between Dates** in Step 2, specify your end date.

The procedure is the same as Steps 5 through 8, above.

- 10. Click Search.**

## Searching by Label

You can use the Search for pane to limit your search to items that have a particular label.

### Procedure

1. Select the desired document type from the Document Types pane.
2. In the Search for pane, select **Label (Word)** from the **Search in** drop-down list.
3. Type the desired label in the **Search for** text box.
4. Click **Search**.

## Searching by Document ID

If you know a document ID, you can use the Search for pane to locate it.

### Procedure

1. In the Document Types pane, select the desired document type.
2. Type the document ID in the Search for text box.

---

 **NOTE:** Use just the ID; do not include the file extension.

---

3. Click **Search**.

---

 **NOTE:** This method locates only the topic with the specified ID. To locate all topics that refer to a specific ID, see [Searching for All References to a Specific Topic](#) on page 55.

---

## Searching for All References to a Specific Topic

Where topics may be shared between multiple topic sets, it is important to know if a topic is used or referenced in multiple locations. You can use the Search view to search for all references to a specific topic.

### Procedure

1. In the Document Types pane, select the desired document type.
2. In the Search for pane, select **href** from the **Search in** drop-down list.
3. Locate the topic you are interested in, right-click it, and select **Copy > Copy Reference**.
4. Paste the reference in the **Search for** text box.
5. Click **Search**.

## Searching for Topics that Contain a Specific Element

You can limit your search results to topics that contain a specific element.

For example, you may want to find all topics in your bookmap that contain the **<draft-comment>** element, so you can make sure all comments are addressed before completing your document.

### Procedure

1. In the Document Types pane, select **Topics**.
2. In the Search for pane, select the desired element from the **Search in** drop-down list.
3. Leave the **Search for** text box empty.
4. Click **Search**.

## Searching for Topics that Contain an Element with Particular Text

You can limit your search results to just those topics that contain an element with particular text.

For example, if you want to find a topic containing a **<draft-comment>** element with a defect number in its text.

#### Procedure

1. In the Document Types pane, select **Topics**.
2. In the Search for pane, select the desired element from the **Search in** drop-down list.
3. Type the desired text in the **Search for** text box.
4. Click **Search**.

## Using Search to Prepare Your Book for Publishing

Before you can publish a map, all maps and topics included in the map must have a status of **Authoring:done**. You can use the Search view to limit your search to just those items that need to be changed.

---

 **NOTE:**

- You cannot move any map to the **Authoring:done** status until all topics it contains are in the **Authoring:done** status.
  - You cannot move any topic to the **Authoring:done** status until all images and any other objects it references are in the **Authoring:done** status.
- 

#### Procedure

1. Make sure your top-level book map is open in the DITA Map editor.
2. In the Search view, clear all the selections the Cycles pane.
3. Select **work** from the drop-down list to the right of **Authoring**.  
This limits the search to items that have a status of **Authoring:work**.
4. In the Document Types pane, select only **Images**.
5. In the Dependencies of section of the Limit to pane, select **DITA Map view**.  
This limits your search to just the items included in your top-level map.
6. Click **Search**.
7. In the Search Results view, select all the items in the results list.
8. Right-click the selected items and select **Change Status**.
9. In the Change Status dialog box, select **Authoring:done** and click **Change**.
10. Repeat Steps 4 through 9 twice, first selecting **Topics**, then selecting **Maps** from the Document Types pane.

## Filtering Search Results Based on Text in the Title

You can limit the number of items displayed in the Search Results view by filtering the results based on text contained in the title.

### Procedure

1. Perform your search.
2. In the Search Results view, type the desired text in the text box located below the search results.

The items displayed are reduced to just those items that contain the text in their title. For example, Search Results view in the following image shows search results that have been limited to just those files that contain the term *kernel* in their titles:

The screenshot shows the Ixiasoft CCMS interface with the 'Search Results' tab selected in the top navigation bar. Below the navigation bar, a search bar contains the text 'kernel'. The main area displays a table titled 'Results List page 1 of 42 (100/4,149 results found)'. The table has three columns: 'Title', 'Type', and 'Id'. The data in the table is as follows:

Title	Type	Id
Example: Patching the Kernel With SCC Files	task-wr	mmo1403548767054
Kernel Patching Overview	concept-wr	mmo1403548767754
Options for Saving Kernel Modifications	concept-wr	ejn1509467967830
Platform Kernel Versions by Product Release	reference-wr	mmo1403548594011
Working with -dirty Kernel Strings	task-wr	mmo1403548840403
Options for Saving Kernel Modifications	concept-wr	lpf1509467946970
Example: Patching the Kernel	task-wr	tyh1509452879113

### Postrequisites

Make sure to clear the text from the text box in the Search Results view before performing another search. If you do not clear this box, you may get undesired results from your next search.

## Adding Columns to the Search Results View

You can add columns to the Search Results view to help you sort and refine your results.

### Procedure

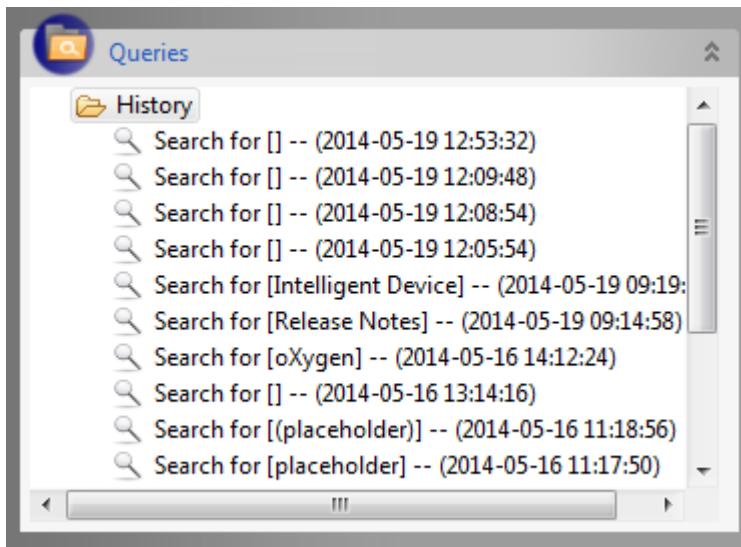
1. Right-click anywhere in the title bar and select the desired item from the drop-down list.  
Items already appearing in the title bar are indicated with a check mark.

2. Drag and drop the new column to the desired position on the title bar.

## Saving Searches

You can save your searches by saving the search terms as a personal query. You can also share your queries with others.

The History folder in the Queries pane of the Search view automatically saves your most recent searches. However, if you did not use a search string in your search configuration, you cannot see what the search terms were in a particular search.

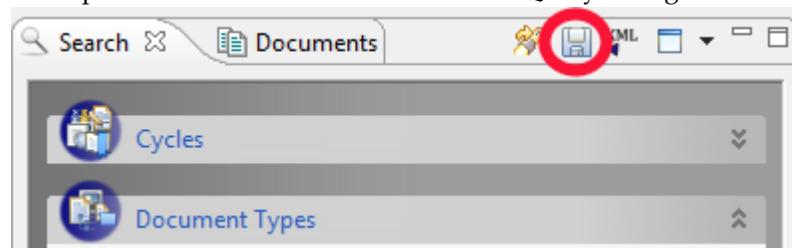


Notice that searches that used non-string filters only show **Search for [ ]**.

You can save a search configuration under a recognizable name. The saved configuration appears in the Personal folder in the Queries pane of the Search view.

### Procedure

1. Configure a search in the Search view.
2. Save the search as a personal query by selecting the **Save Query** button and entering a descriptive name for the search in the Save Query dialog box.



3. Use the saved query.
  - a) In the Queries pane, expand **Personal**.

- b) Double-click on the query you want to run.  
The search parameters are filled from the saved query.
  - c) Click **Search**.
4. Share your query with others.
- a) Right-click your query and select **Share queryname**.
  - b) In the Share query with dialog box, select the roles and groups you want to share the query with.
  - c) Click **OK**.

## Searching Using the AND Operator

You can reduce the search results by using the Advanced Search pane to specify additional search terms.

For example, if you want to search for topics that contain both a **<draft-comment>** element and an **@href** attribute, you would do the following:

### Procedure

1. On the first line in the Advanced Search pane, select **index** from the **Type** drop-down list.

 **NOTE:** You must first click in the column before the drop-down list appears.

2. Select **Contains element** from the **Name** drop-down list.
3. Select **draft-comment** from the **Value** drop-down list.
4. Click the **Add a new line** button.  
The **and** operator is selected by default.
5. Select **attributes** from the **Type** drop-down list.
6. Select **href** from the **Name** drop-down list.
7. Leave the **Value** field blank.
8. Click **Search**.

## Searching Using the OR Operator

Sometimes you need to expand rather than narrow a search. One way to do this is to use the **OR** operator. This is an inclusive **OR**.

For example, to expand the search to all documents that include either a `<draft-comment>` element or an `@href` attribute, you would do the following:

### Procedure

1. On the first line in the Advanced Search pane, select **index** from the **Type** drop-down list.

 **NOTE:** You must first click in the column before the drop-down list appears.

2. Select **Contains element** from the **Name** drop-down list.
3. Select **draft-comment** from the **Value** drop-down list.
4. Click the **Add a new line** button.
5. Select **or** from the **Operator** drop-down list.
6. Select **attributes** from the **Type** drop-down list.
7. Select **href** from the **Name** drop-down list.
8. Leave the **Value** field blank.
9. Click **Search**.

## Making Bulk Changes to XML

You can search and replace XML text strings within the context of a map.

### Prerequisites

Documents must be locked in order for the replace to succeed. You can lock the documents in advance or allow the Ixiasoft CCMS to lock them during the process.

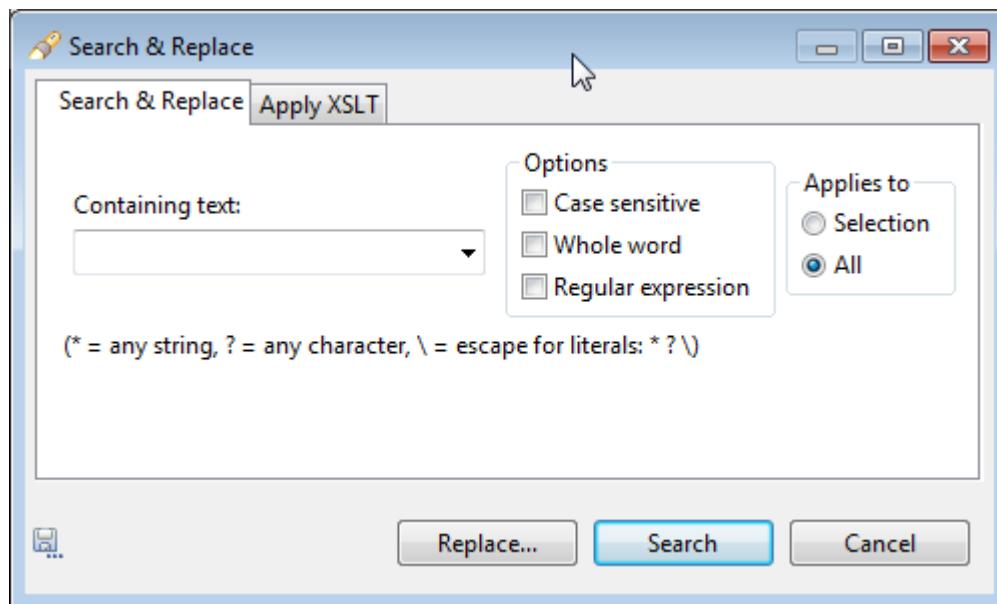
 **CAUTION:** This powerful feature can alter the XML code in your documents. Use it with care to avoid making your documents invalid.

### Procedure

1. Open a map in the DITA Map view.

2. To limit your search, select specific topics. Otherwise, all topics are searched.
3. Click on the triangular menu icon at the upper right of the view and select **Search and Replace**.

The Search and Replace dialog box appears.



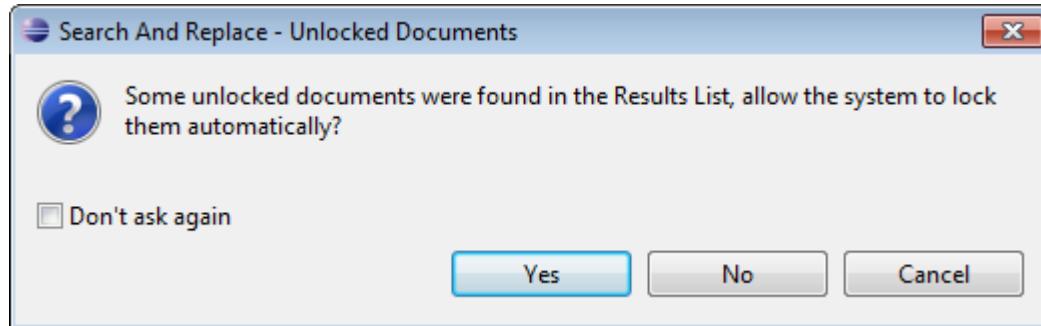
4. Select options.

Name	Meaning
Containing text	<p>The string you are searching for. The string can contain wildcards but not operators. The drop-down list shows recently used text or expressions.</p> <p><b>NOTE:</b> If you search for a string with spaces (for example, controller configuration), you may not get all results. This is because the XML may contain line breaks and multiple spaces, which are not recognized by the search. To ensure that you find all occurrences, use a regular expression; for example:</p> <p>controller\n* *configuration</p> <p>This finds the word controller, followed by newline (0 or more), followed by space (0 or more).</p>
Case sensitive	Finds only text that matches the case of your search string.
Whole word	Finds only whole words. For example, the string <b>cat</b> does not find <b>caterpillar</b> in the text.
Regular expression	Causes your string to be interpreted as a regular expression. If this option is not selected, the [ ] characters are interpreted as wildcards.

Name	Meaning
Applies to	If you have selected specific files for a limited search, select <b>Selection</b> .

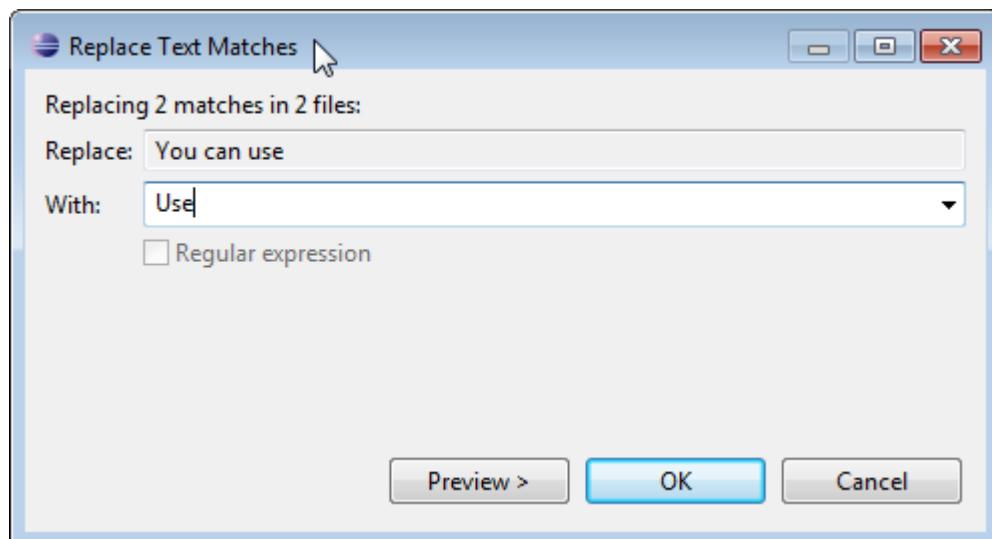
5. Click **Replace**.

If any of the target documents is unlocked, the Unlocked Documents confirmation window appears. Otherwise, the Replace Text Matches dialog box appears.



6. If the Unlocked Documents confirmation window appears, click **Yes** to continue with the replace operation.

The Replace Text Matches dialog box appears.



7. Enter your replacement text in the **With** field or select it from the drop-down list.

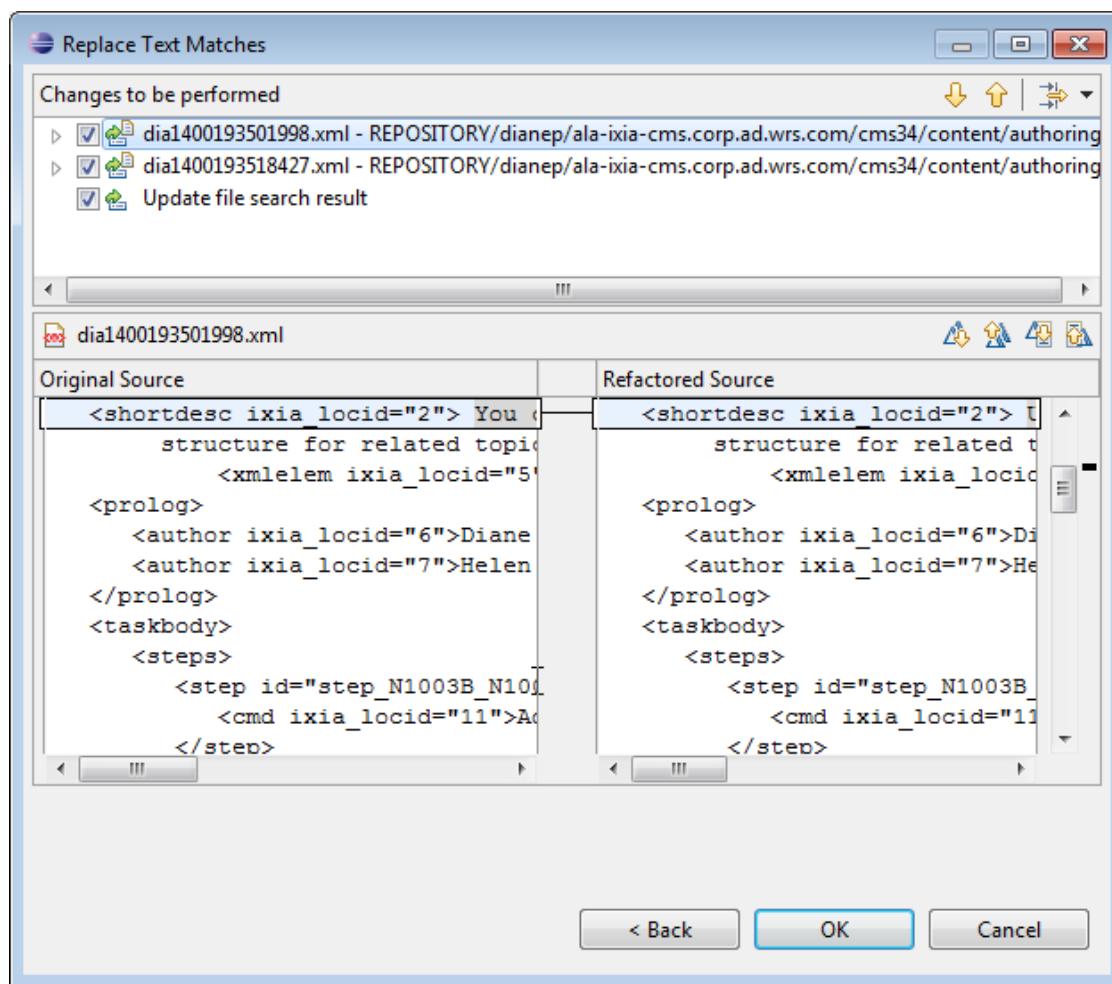
**Regular expression** is selected if you chose this option for your search.

8. Click **Preview**.



**CAUTION:** Do not click **OK**. Making changes without reviewing them can make damaging changes and send them to the repository.

The Replace Text Matches dialog box appears.



9. Confirm each of the changes to be performed.
  - a) Highlight the item in the Changes to be performed pane.
  - b) Compare the highlighted contents of Original Source and Refactored Source.
  - c) Decide to make the change or not.
    - To make the change, do nothing and continue to the next item.
    - To cancel the change, clear the checkbox for that item.

**10. Click OK.**

The replacement action is carried out on the selected items and the documents are saved and unlocked.



# 7

## *Using oXygen*

<a href="#">The oXygen Editor</a>	65
<a href="#">Opening the oXygen XML Author Perspective</a>	66
<a href="#">Author Mode</a>	66
<a href="#">Adding Elements in Author Mode</a>	67
<a href="#">Adding Attributes Using the Attributes View</a>	67
<a href="#">Adding Attributes Using the Edit Attributes Menu Item</a>	68
<a href="#">Adding Elements in Text Mode</a>	68
<a href="#">Adding Attributes in Text Mode</a>	69
<a href="#">Working in Grid Mode</a>	69
<a href="#">Editor Error Feedback</a>	70
<a href="#">Shortcut Keys Reference</a>	72

### **The oXygen Editor**

The oXygen editor is your main work area for writing and editing topics and customizing maps.

The default editor used by Wind River Technical Publications is oXygen. When you configure the Ixiasoft CCMS you configure oXygen to be the default editor. The *Ixiasoft CCMS User Guide for oXygen* is available from the Help menu and contains general instructions on using oXygen in the Ixiasoft CCMS environment.

## Opening the oXygen XML Author Perspective

If you are more familiar with the standalone version of oXygen, you may prefer to work in the oXygen XML Author perspective.

 **NOTE:** Many of the Ixiasoft CCMS options, including the Search view, are not available in this perspective.

### Procedure

1. Click **Window > Open Perspective > Other**.
2. In the dialog box, click **oXygen XML Author**.
3. Click **OK**.

The oXygen XML Author perspective appears in the upper right-hand corner.

## Author Mode

Author mode uses a CSS style sheet to render the XML tags as formatting, making the content easier to read and edit.

 **NOTE:** The style sheet used to display content in Author mode is not the one that is used in building the document. Author mode does not reflect the final formatting; check the formatting by generating output.

The easiest way to work with Author mode is to set your oXygen preferences to display Full Tags. The tags appear inline as flags, which are easier to understand than raw XML, but also easy to navigate.

You can edit in author mode as long as you place the cursor in the correct container element. The flags help you to know what container the cursor is in.

The bar at the top of the editor window also indicates the element where the cursor is currently located. The element containing each element is also shown, back to the root element.

## Adding Elements in Author Mode

You can add DITA elements to your topic by several different methods in Author mode.

### Procedure

1. Place the cursor in the location you want to add the element or highlight the text you want to tag with an inline element.
2. Add the element in one of the following ways:

Options	Description
Using the Elements view:	Double-click the element name in the Elements view or highlight it and press <b>ENTER</b> .
Using the drop-down menu:	Press <b>CTRL+SPACE</b> or <b>ENTER</b> and select the element you want from the drop-down list. If you start typing the element name, non-matching elements drop off the list.

All tags that you can legally place at the cursor location appear in the Elements tab and in the drop-down list. If the tag you need does not appear, try moving the cursor to a different location.

3. [Identify and correct errors.](#)

You can enter some kinds of incorrect markup in author mode. If you make an error, it is marked in two ways:

- A red bar appears in the status bar to the right of the editor scroll bar. You can navigate to the error by double-clicking the red bar and cycle through the errors using the up and down errors at the bottom of the status bar.
- A squiggly red line underlines the incorrect text and tag.

Both the bar and the line disappear when you correct the error.

## Adding Attributes Using the Attributes View

You can add DITA attributes to your elements using the Attributes view in Author mode. This method is particularly helpful if you are adding several attributes to the same element.

### Procedure

1. Select the element (or insert your cursor inside the element).
2. In the Attributes view, double-click in the Value column next to the desired attribute.



**NOTE:** Be careful when you double-click in the Value column. Even if you do not select or type a value, oXygen adds the attribute to your topic. In some cases, an empty value to an attribute will cause your topic to become invalid.

If this happens, you need to remove the attribute by right-clicking it and selecting **Remove** or by manually deleting it in Text mode.

- 
3. Type the value of the attribute or select a value from the drop-down list.

## Adding Attributes Using the Edit Attributes Menu Item

You can add DITA attributes to your elements using the **Edit Attributes** menu item. This is the quickest method if you are adding a single attribute to an element.

### Procedure

1. Right-click on the element tag and select **Edit Attributes**.
2. Select the attribute name from the drop-down list.
3. Type the value of the attribute or select a value from the drop-down list.
4. Click **OK**.

## Adding Elements in Text Mode

You can add DITA elements to your topic manually by typing the appropriate markup in Text mode.

Text mode is the standard, raw XML view. It is more difficult to see your content in text mode, but it is the view that shows the XML tags explicitly and is therefore the view to use for resolving XML tagging problems.

### Procedure

1. Place your cursor where you want to add the element.
2. Type < and select the element you want from the drop-down list.

If you continue typing the element name, non-matching elements drop off the list. Double-click the element in the list to add it or highlight the element and press **ENTER**.

The editor enters both starting and closing tags. In some cases, it also enters additional required elements, such as an **<li>** element between the starting and ending tags of a **<ul>** element.

If you highlight an element (including the angle brackets) and delete it, the editor deletes the associated closing tag.

If you highlight just the name of the element (not including the angle brackets) and modify it, the editor changes the associated closing tag to match.

**3. Add content.**

Type your content between the appropriate tags. White space greater than one space is ignored except for code.

**4. Identify errors.**

You can enter incorrect markup in text mode. If you make an error, it is marked in two ways:

- A red bar appears in the status bar to the right of the editor scroll bar. You can navigate to the error by double-clicking the red bar and cycle through the errors using the up and down errors at the bottom of the status bar.
- A squiggly red line underlines the incorrect text and tag.

Both the bar and the line disappear when you correct the error.

## **Adding Attributes in Text Mode**

You can add attributes to your elements manually by typing the appropriate markup in Text mode.

Text mode is the standard, raw XML view. It is more difficult to see your content in text mode, but it is the view that shows the XML tags explicitly and is therefore the view to use for resolving XML tagging problems.

### **Procedure**

**1. Insert your cursor in the starting tag for element you want to add an attribute to.**

The cursor must be between the last letter of the name and the angle bracket >.

**2. Press the spacebar and select the desired attribute from the drop-down list.**

If you start typing the attribute name, non-matching elements drop off the list. Double-click the attribute in the list to add it or highlight the attribute and press **ENTER**.

**3. Type the attribute value between the quotation marks or select it from the drop-down list.**

## **Working in Grid Mode**

Grid mode allows you to edit XML in a structured grid of nested tables.

In Grid mode, the XML tags are rendered as table cells. Grid mode is particularly useful for restructuring and sorting documents.

### Procedure

- Edit in grid mode by editing the table as you would an Excel spreadsheet.  
Highlight a cell, right-click it, and select the appropriate action from the drop-down menu.
- Expand or hide elements using the arrow buttons at the left of each element that contains other elements.

## Editor Error Feedback

The oXygen editor provides multiple indicators of incorrect XML, including the red bars to the right of the editor window, red-squiggle underlining of errors, an error description at the bottom of the window, and pop-up error messages.

In some cases in Author mode, oXygen does not allow you to enter a particular element at the location you choose; in this case, you will receive a pop-up error message. In other cases, you can enter the element in either Text or Author mode but additional elements are required; in this case, oXygen marks the error in the error bar and in the editor pane, and provides an error message, both at the bottom of the editor window and as a tooltip when you float over the error marker. The error feedback clears when you correct the error.

### The Error Tray

The error tray is located at the right side of the editor window.

The block at the top of the tray is red when you have XML errors and green when your XML is correct. When the block is red, red bars also appear in the tray.

- Double-clicking on a red bar places the cursor on the error.
- Using the up and down arrows at the bottom of the tray cycles through the errors.
- Hovering over a red bar opens an error pop-up and causes the error message to appear in the message tray at the bottom of the window.

The bar disappears when you correct the error.

### Error Underlining

Incorrect XML is marked with a red-squiggle underline. This differs from the standard red underline that indicates a spelling error if you have activated continuous spell-checking. Hovering over text with the red-squiggle underline opens an error pop-up. Clicking on the error text causes the error message to appear in the message tray at the bottom of the window.

The underline disappears when you correct the error.

### The Message Tray

The message tray is located at the bottom of the editor window above the **Text/Grid/Author** tabs. It displays an error message when you click on error text or on a red bar in the error tray. That message appears until you click on a different error.

## The Error Pop-up

A pop-up error message appears when you hover over either text marked as an error or the red error bar in the error tray. Hovering does not change the focus for the message in the message tray.

When you hover over non-error text, a pop-up message appears that explains the element you are hovering over.

## Example Error Messages

- **The element xxx is not allowed here.**

The error also includes the following statement: "To insert this element here, disable the 'Allow only insertion of valid elements and attributes' option.

- **Unexpected element xxx . The content of the parent element type must match..."**

The error message lists the permitted elements from the Information Model. For example:

"(prereq?,context?,section\*,(steps | steps-unordered)?,result?,example?,postreq?)"

prereq?

The `<prereq>` element is permitted but not required here. You may include more than one `<prereq>` element and the `<prereq>` elements may occur in any order before `<steps>`.

context?

The `<context>` element is permitted but not required here. You may include more than one `<context>` element and the `<context>` elements may occur in any order before `<steps>`.

section\*

One or more `<section>` elements are permitted but not required here. They must follow any `<prereq>` and `<context>` elements.

(steps | steps-unordered)?

Either the `<steps>` or `<steps-unordered>` elements are permitted but not required here.

You can only use one type of step in a topic.

result?

The `<result>` element is permitted but not required here. It must follow any step elements.

example?

The `<example>` element is permitted but not required here. It must be last unless you include a `<postreq>` element.

postreq?

The `<postreq>` element is permitted but not required here. It must be the last element.

## Shortcut Keys Reference

The Ixiasoft CCMS and the oXygen editor have a rich set of keyboard functionality. The shortcut keys help provide an easier and usually quicker method of navigating and editing your files.

### Adding Elements

Shortcut Key	Description
<b>CTRL+SPACE</b>	Add an element.
or	
<b>ENTER</b>	Type the first few letters of the desired element or use the up or down arrows to select the desired element.

### Editing Files

Shortcut Key	Description
<b>CTRL+SHIFT+Q</b>	Check spelling.
<b>ALT+SHIFT+,</b>	Delete the current element. Removes the tags from around the text.
<b>CTRL+DELETE</b>	Delete the next word.
<b>CTRL+BACKSPACE</b>	Delete the previous word.
<b>CTRL+F</b>	Find/replace text in the current file.
<b>ALT+DOWN ARROW</b>	Move lines down.
<b>ALT+UP ARROW</b>	Move lines up.
<b>ALT+SHIFT+R</b>	Rename an element. You can use the radio buttons in the Rename dialog box to rename just the current element, sibling elements with the same name, or all elements with the same name. This can be particularly helpful when manually converting from SPFE.
<b>ALT+SHIFT+/-</b>	Surround the selected text with the last element tag you used.

### Managing Files

Shortcut Key	Description
<b>CTRL+F4</b>	Close the current file.
or	

Shortcut Key	Description
<b>CTRL+W</b>	
<b>CTRL+SHIFT+F4</b>	Close all files open in the editor.
or	
<b>CTRL+SHIFT+W</b>	
<b>CTRL+ALT+I</b>	Create a new image.
<b>CTRL+ALT+M</b>	Create a new map.
<b>CTRL+ALT+T</b>	Create a new topic.
<b>CTRL+S</b>	Save the current file.
<b>CTRL+SHIFT+S</b>	Save all open files.

#### Navigating Within a File

Shortcut Key	Description
<b>CTRL+RIGHT ARROW</b>	Move cursor to the start of the next word.
<b>CTRL+LEFT ARROW</b>	Move cursor to the start of the previous word.
<b>CTRL+DOWN ARROW</b>	Scroll one line down.
<b>CTRL+UP ARROW</b>	Scroll one line up.
<b>CTRL+A</b>	Select all.
<b>CTRL+SHIFT+RIGHT ARROW</b>	Select the next word.
<b>CTRL+SHIFT+LEFT ARROW</b>	Select the previous word.
<b>SHIFT+TAB</b>	In a table, move the cursor to the previous cell.
<b>TAB</b>	In a table, move the cursor to the next cell.



# 8

## Maps

[Types of Map Templates](#) 75

[The DITA Map View](#) 76

[Creating a New Map](#) 77

[Opening a Map](#) 79

[Comparing Maps](#) 80

### Types of Map Templates

Wind River provides customized templates for both product documentation and training documentation. You can use these templates to create maps in DITA that allow you not only to organize your topics into the final deliverables but also to generate output.

Map Type	Description
Product Doc - Bookmap for Books with Parts	<p>Use a bookmap to organize your topics at the document level (book level). The parts bookmap template includes examples of entries for a parts-based book.</p> <p><b>NOTE:</b> You can generate PDF output from a map at this level, but you cannot generate combined HTML and PDF output. You must use an Eclipse Help map and Eclipse Help Submap in combination with a bookmap if you want to generate HTML output for your document without posting it to the Product Documentation site.</p>

Map Type	Description
Product Doc - Bookmap	<p>Use a bookmap to organize your topics at the document level (book level). The bookmap template includes examples of topics organized into chapters.</p> <p><b>NOTE:</b> You can generate PDF output from a map at this level, but you cannot generate combined HTML and PDF output. You must use an Eclipse Help map and Eclipse Help Submap in combination with a bookmap if you want to generate HTML output for your document without posting it to the Product Documentation site.</p>
Product Doc - Eclipse Help Map for Product Name	Use an Eclipse Help Map to generate the HTML and PDF and provide the required Eclipse information for product documentation that integrates with the Workbench help system.
Product Doc - Eclipse Help Submap for Product Name	An Eclipse Help Submap is an intermediary map between your Eclipse Help Map and the bookmap for your document.
Product Doc - Topics Submap	<p>Use a topics submap to organize your topics at the chapter level. Your bookmap can refer to these maps to pull in topics at a chapter level.</p> <p>You can also use submaps to create specialized maps, such as relationship table maps.</p>
Training - Lab Book	Use this map as your top-level map when creating a lab book for Wind River instructor-led training (ILT).
Training - Lab Submap (Chapter)	<p>Use a submap to organize your topics at the module level (book chapter, or individual lab) for a Wind River training module.</p> <p><b>NOTE:</b> You must complete the metadata as appropriate for the lab you are authoring. If the metadata is incomplete or wrong, the transformation and other book building tools will not function correctly.</p>

## The DITA Map View

The DITA map view provides a hierarchical overview of your topic set. It is one key way of locating content in the Ixiasoft CCMS.

The hierarchical view shows all the children of the map you open, including submaps and topics. It also shows hierarchical elements such as <chapter> elements. Items that have children can be expanded to show the children or closed. Each child is marked with an icon that shows what kind of document it is:

- bookmap
- DITA map
- concept topic
- task topic
- reference topic

The icons show an additional lock symbol when the document is locked, and the document title appears in bold; the text is black if a topic is selected in the map and blue otherwise. By default only the Title and Status columns appear. You can customize the columns by right-clicking and selecting or unselecting columns and by dragging them to new positions.

The screenshot shows the DITA Map view with the following details:

**Toolbar:** Properties, DITA Map, Elements, Attributes.

**Message Bar:** 70 children in the map. [1 item selected].

**Table Headers:** Title, Status, Audience, Labels.

**Table Data:**

Title	Status	Audience	Labels
<topichead> #Wind River Intelligent Device Platform XT, 2.0			
Wind River DITA Internal for CCMS Technical Writer's Guide	⚠️ Authoring:work	Wind River Internal	
<chapter> #CMS Conceptual Overview			
<chapter> #CMS Client Overview			
<chapter> #Authoring Environment Setup for the CMS			
<chapter> #Searching in the CMS			
<chapter> #Working with Files			
<chapter> #Working with Maps			
Working with Maps	⚠️ Authoring:work	Wind River Internal	
DITA Map Configuration	⚠️ Authoring:work	Wind River Internal	
Creating a New Map in the CCMS	⚠️ Authoring:work	Wind River Internal	
Adding Topics to a Map	⚠️ Authoring:work	Wind River Internal	
Customizing a Map	⚠️ Authoring:work	Wind River Internal	
<topicgroup> #Bookmaps			
Adding a Registered Trademark to Title Pages	⚠️ Authoring:work	Wind River Internal	
Adding Copyright Information	⚠️ Authoring:work svn; ccms	Wind River Internal	
Adding a Table of Contents	⚠️ Authoring:work svn; ccms	Wind River Internal	
<b>Adding Chapters to Your Bookmap</b>	⚠️ Authoring:work	Wind River Internal	
<b>Adding Parts to Your Bookmap</b>	⚠️ Authoring:work	Wind River Internal	
<topicgroup> #DITA Maps			

**Bottom Bar:** Wind River Internal Documentation for CMS, 1.0 [Authoring-English] [dia1399929353839], Refresh.

At the bottom of the view, the drop-down list contains maps you have opened previously. To change to a map you previously viewed, select that map in the drop-down list and then click Refresh.

In the above example, *Wind River DITA Internal for CCMS Technical Writer's Guide* is the bookmap. *Working with Maps* is an example of a topics submap (chapter map). Most chapters are closed but *Working with Maps* is expanded. *Adding Copyright Information* and *Adding a Table of Contents* both have the Audience attribute set, meaning they are used to create output for different audiences. *Adding Chapters to Your Bookmap* is locked and *Adding Parts to Your Bookmap* is selected.

## Creating a New Map

You can create new maps using customized map templates for both product documentation and training documentation.

For information about the different map templates, see [Types of Map Templates](#) on page 75.

## Procedure

1. Select IXIASOFT CCMS > Create Map.
2. Type the title of your map in the dialog.



**NOTE:** For Eclipse Help maps and Eclipse Help Submaps, it is best practice to include the type of map and the product name in the title; for example:

- Eclipse Help Map for Helix Cockpit
- Eclipse Help Submap for Helix Cockpit

3. Select the appropriate map template:
  - Product Doc - Bookmap for Books with Parts
  - Product Doc - Bookmap
  - Product Doc - Eclipse Help Map
  - Product Doc - Eclipse Help Submap
  - Product Doc - Topics Submap
  - Training - Lab Book
  - Training - Lab Submap (Chapter)
4. Select any of the following options:

Option	Description
Open map in DITA view	The new map opens in the DITA Map view, replacing any map currently open in the view.
Add to current map as sub-map	The new map is added to the bottom of the map that is currently open in the DITA Map view.
<b>NOTE:</b> This option is only available if the map in the DITA Map view is locked.	

5. Click **Create**.

## Opening a Map

By default, when you create a new map, the Ixiasoft CCMS saves the map but does not open it.

However; if you selected **Open map in DITA Map view** when you created the map, the map opens and replaces the current map in the DITA Map view.

If your map is not open, you can open and edit it in several ways.

### Procedure

1. Use the **Search** tab to locate the map.
2. Open the map using one of the following methods:

Editor/View	Method
<b>DITA Map view</b>	The DITA Map view is the default map editor for the Ixiasoft CCMS. You can edit the map by using the toolbar or by dragging and dropping topics.  To open the map in the DITA Map view, double-click the map in the Search Results view.
<b>Generic Text Editor</b>	To open the map in a plain text editor, right-click the map in the Search Results view and select <b>Open With &gt; Generic Text Editor</b> .
<b>DITA Map editor</b>	The DITA Map editor is identical to the DITA Map view, except it opens in the editor pane.  This feature comes in handy if you want to see two maps side-by-side (one in the DITA Map editor and one in the DITA Map view).  To open the map in the DITA Map editor, right-click the map in the Search Results view and select <b>Open With &gt; DITA Map Editor</b> .
<b>Text editor</b>	To open the map in a plain text editor, right-click the map in the Search Results view and select <b>Open With &gt; Text Editor</b> .

---

Editor/View	Method
Oxygen DITA Map editor	The Oxygen DITA Map editor allows you to edit the map in several modes: Author mode, Text mode, and Grid mode.  To open the map in the Oxygen DITA Map editor, right-click the map in the Search Results view and select <b>Open With &gt; Oxygen DITA Map Editor</b> .

---

## Comparing Maps

The Ixiasoft CCMS provides a **Compare With** tool that allows you to compare topic sets at the TOC level by comparing their maps.

One case where you may want to compare maps is when you have two topic sets that include some of the same topics. Comparing the maps allows you to see that the correct topics are included in each map.

You may also want to compare a published map with a map in the authoring state to see what has changed.

---

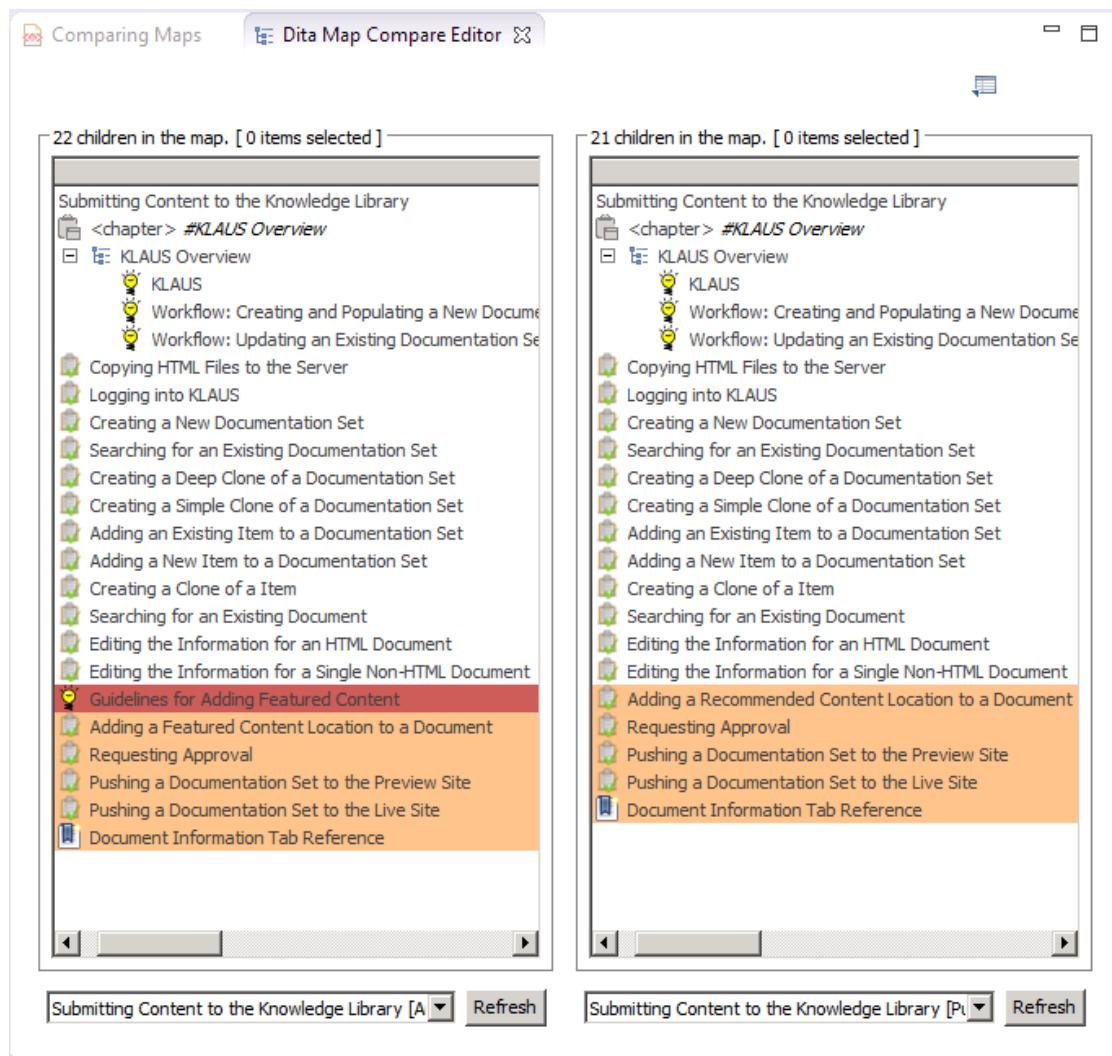
 **NOTE:** You can use the same procedure to compare any two objects of the same type except images. Images cannot be compared.

---

### Procedure

1. Locate the two maps.  
You need to find search terms that will display both maps in the Search Results view.
2. Highlight the two maps in the Search Results view.
3. Right-click on the highlighted maps and select **Compare With > Compare With Each Other**.

Both maps open in the Dita Map Compare Editor tab in the DITA Map Editor.



4. Add additional columns to the title bar.

The Dita Map Compare Editor tab displays columns similar to those in the Search Results view. You can add additional columns to aid in your comparison.

In each of the maps, right-click anywhere in the title bar and select the desired item.



**NOTE:** You may have to maximize the editor and adjust the width of the columns to see all the information.

For example, if you are comparing a published map to a map in the authoring state, you would need to add the following:

- **Authoring Revision**
- **Revision**

You would then compare the values in the Authoring Revision column for the published map to the values in the Revision column of the map in the authoring state.

The screenshot shows two windows of the DITA Map Compare Editor. Both windows have a title bar "Comparing Maps" and a tab "Dita Map Compare Editor".

**Left Window (Authoring State):**

Table Title: [22 children in the map, 0 items selected]

Title	Status	ID	Revision
Submitting Content to the Knowledge Library	Authoring:done	abq1472491791063	15
└ Chapter > #KLAUS Overview			
└ T: KLAUS Overview	Authoring:done	arb1472674452636	3
└ Workflow: Creating and Populating a New Documentation Set	Authoring:done	vgn1472662086283	7
└ Workflow: Updating an Existing Documentation Set	Authoring:done	bdf1472657510596	7
└ Copying a Map Set to the Server	Authoring:done	cwq14726575105976	8
└ Logging into KLAUS	Authoring:done	ndp14726575105972	8
└ Creating a New Documentation Set	Authoring:done	ams14726575105955	5
└ Searching for an Existing Documentation Set	Authoring:done	uyt14726530910655	9
└ Creating a Deep Clone of a Documentation Set	Authoring:done	lyt14726532275405	12
└ Creating a Simple Clone of a Documentation Set	Authoring:done	vma14726532218671	9
└ Adding a New Item to a Documentation Set	Authoring:done	vr114726532219513	9
└ Adding a New Item to a Documentation Set	Authoring:done	rsv147265322194528	9
└ Creating a Clone of a Thread	Authoring:done	ctf147265322194525	10
└ Searching for an Existing Document	Authoring:done	kgr147265322194588	5
└ Editing the Information for an HTML Document	Authoring:done	lgr147265322194586	9
└ Editing the Information for a Single Non-HTML Document	Authoring:done	yrx1472653221945148	9
└ Document Information Tab Reference	Authoring:done	pre1473178699532	8
└ Adding a Featured Content Location to a Document	Authoring:done	zrv1472653221945149	5
└ Requesting Approval	Authoring:done	ogv1472653221945228	6
└ Pushing a Documentation Set to the Preview Site	Authoring:done	gnv1472653221945238	7
└ Pushing a Documentation Set to the Live Site	Authoring:done	pbt14726532219452630	5
└ Document Information Tab Reference	Authoring:done	ptt1473177438873	7

**Right Window (Published State):**

Table Title: [21 children in the map, 0 items selected]

Title	Status	ID	Authoring Revision
Submitting Content to the Knowledge Library	Published:done	abq1472491791063	15
└ Chapter > #KLAUS Overview			
└ T: KLAUS Overview	Published:done	arb1472674452636	2
└ Workflow: Creating and Populating a New Documentation Set	Published:done	vgn1472662086283	7
└ Workflow: Updating an Existing Documentation Set	Published:done	bdf1472657510596	7
└ Copying a Map Set to the Server	Published:done	cwq14726575105976	6
└ Logging into KLAUS	Published:done	ndp14726575105972	8
└ Creating a New Documentation Set	Published:done	uyt14726530910655	5
└ Searching for an Existing Documentation Set	Published:done	lyt14726532275405	12
└ Creating a Deep Clone of a Documentation Set	Published:done	vma14726532218671	9
└ Creating a Simple Clone of a Documentation Set	Published:done	vr114726532219513	9
└ Adding a New Item to a Documentation Set	Published:done	rsv147265322194528	9
└ Adding a New Item to a Documentation Set	Published:done	kyv147265322194525	10
└ Creating a Clone of a Thread	Published:done	ocf1473353604988	5
└ Searching for an Existing Document	Published:done	kgr147265322194588	9
└ Editing the Information for an HTML Document	Published:done	lgr147265322194586	9
└ Editing the Information for a Single Non-HTML Document	Published:done	yrx1472653221945148	9
└ Document Information Tab Reference	Published:done	pre1473178699532	8
└ Adding a Featured Content Location to a Document	Published:done	zrv1472653221945149	5
└ Requesting Approval	Published:done	ogv1472653221945228	6
└ Pushing a Documentation Set to the Preview Site	Published:done	gnv1472653221945238	7
└ Pushing a Documentation Set to the Live Site	Published:done	pbt14726532219452630	5
└ Document Information Tab Reference	Published:done	ptt1473177438873	7

Both windows have a status bar at the bottom:

Submitting Content to the Knowledge Library [Authoring:English] (CMS 1.0 2017-01-20; CMS 1.0 2017-02-10; CMS 1.0 2017-03-27; CMS 1.0) Refresh

Submitting Content to the Knowledge Library [Published:English] (CMS 1.0 2017-01-20) Refresh

# Eclipse Help Maps

[Setting Up an Eclipse Help Map](#) 83

## Setting Up an Eclipse Help Map

After you create an Eclipse Help map, you must customize it in order to generate combined PDF and HTML for your document set. Building this map also produces output that is suitable for a Wind River Help (Workbench) plugin.

### Procedure

1. Lock your map.
2. Open the map in the oXygen DITA Map editor.
3. Select the **Text** tab in the lower left corner of the editor.
4. Ensure the title conforms to standards.

It is best practice to include the type of map and the product name in the title; for example, *Eclipse Help Map for Helix Cockpit*.

5. Update the value of the **<pluginname>** element.  
Replace the *fullProductName* placeholder with the full name of your product.
6. Update the **<vrm>** element.
  - a) Update the **@version**, **@release**, and **@modification** attributes.



---

**NOTE:** The **@version**, **@release**, and **@modification** attributes are used differently than they are in a bookmap.

---

The **@version**, **@release**, and **@modification** attributes in the help map correspond to the major, minor, and patch digits in the release number of your product. For example, if you

have a release number of 1.2.3, set the **@version** attribute to 1, the **@release** attribute to 2, and the **@modification** attribute to 3.



**NOTE:** When you save your map, oXygen automatically changes the order of the items in the **<vrmlist>** element to be in alphabetical order. Make sure you are adding the values to the correct attribute.

For example:

```
<vrmlist>
  <vrm version="1" release="2" modification="3"/>
</vrmlist>
```

- b) Remove any attribute you are not using.

For example, if your release number is only two digits, remove the modification attribute.

However, the version attribute is a required attribute. If your product does not have a version number, enter a space in the version attribute in place of a number and delete the release and modification attributes.

- If your product has a fourth digit in the release number, add it to the **<qualifier>** element in the next step.

7. If your product has a fourth-digit in the release number, or if your product is delivered as part of Workbench 4 Help, remove the comment markup (`<!--` and `-->`) from around the **<qualifier>** element and update the value.

Option	Description
<b>Product has a fourth-digit release number</b>	Update the value of the <b>&lt;qualifier&gt;</b> element to match the fourth digit of the release number.
<b>Product is delivered as part of Workbench 4 Help</b>	The <b>&lt;qualifier&gt;</b> element must end with <code>_qualifier</code> ; for example, <code>linux_qualifier</code> .

8. Add an **@href** attribute in the **<tocref>** element to point to your Eclipse Help Submap.

For example:

```
<tocref href="ekn1446152592007.ditamap"/>
```

9. Update the **@id** attribute in the **<osgiManifest>** element.
  - a) To form the correct name for the plugin, replace the *FIXME* placeholder with the name of the product followed by the product's version number.

For Example, for Wind River Intelligent Device Platform XT, 2.0:

```
<osgiManifest id="com.windriver.ide.doc.wr_intelligent_device_platform_XT_2.0">
```

9. b) If your product is a "tooling" product, update the **@id** attribute by deleting the version number and the underscore that precedes it.

A tooling product is a product whose Eclipse plugin name includes the word "tooling." For example, Wind River System Viewer belongs in the

`com.windriver.ide.doc.wr_system_viewer_tooling_3.3` plugin, and is therefore considered a tooling product.

Most Wind River products do not fall under this category. Their topic IDs match the file name, which includes the product release number. However, the topic ID for tooling products should not include the release number. Only one version of a tooling product can appear in an install. Wind River Help uses the identical ID values in different revisions to identify that they are the same product. Thus, the latest version of the product replaces the previous version in the install.

10. Save the file.



# 10

## *Eclipse Help Submaps*

<a href="#">Setting Up an Eclipse Help Submap in Text Mode</a>	87
<a href="#">Adding Links to an Eclipse Help Submap Using the DITA Map View</a>	89
<a href="#">Linking to a Bookmap When There is Only One Book in the Plugin</a>	90
<a href="#">Adding Subcategories to Your Plugin for Workbench Help</a>	91
<a href="#">Linking to a Non-DITA Topic Set</a>	92
<a href="#">Linking to a Non-DITA Topic Set that Must Be Wrapped</a>	92
<a href="#">Inserting a Plugin for a Sub-Product Inside a Product Plugin</a>	93
<a href="#">Linking to a PDF File</a>	94

### **Setting Up an Eclipse Help Submap in Text Mode**

After you create an Eclipse Help Submap, you must customize it to specify the appropriate anchor ID and product name and to reference all the documents in your document set. Editing the map in Text mode is typically the easiest way to set up your map.

#### **Procedure**

1. Lock your map.
2. Open your map in the oXygen DITA Map editor.
3. Select the **Text** tab in the lower left corner of the editor.
4. Ensure the title conforms to standards.

It is best practice to include the type of map and the product name in the title; for example, *Eclipse Help Submap for Helix Cockpit*.

5. Replace the *anchorId* placeholder in the @anchorref attribute with an appropriate value.

Does Documentation Integrate with Workbench Help?	Method
<b>Yes</b>	Use the value for your product from the list of anchor IDs at: <a href="https://jive.windriver.com/docs/DOC-41488">https://jive.windriver.com/docs/DOC-41488</a> For example, for Wind River Linux:

```
<eclipse-sub-map-wr anchorref="../
com.windriver.ide.doc.globals/
toc.xml#wr_linux"
id="auto-generated-topic-id"
xml:lang="en-us">
```

**NOTE:** If an anchor ID for your product does not exist, file a Jira ticket for the Infrastructure team (TECHPUBINF).

No	Use the value of <b>bottomline</b> . For example:
	<pre>&lt;eclipse-sub-map-wr anchorref="../ com.windriver.ide.doc.globals/ toc.xml#bottomline" id="auto-generated-topic-id" xml:lang="en-us"&gt;</pre>

6. Update the <navtitle> element.

Replace the *Product Name and Version* placeholder with the full name and release number of your product.

For example, for Wind River Helix Cockpit, 1.0:

```
<topichead>
  <topicmeta>
    <navtitle>Wind River Helix Cockpit, 1.0</navtitle>
  </topicmeta>
```

7. Add links to all the documents in your documentation set after the following line:

```
<anchor id="top"/>
```

The easiest way to do this is by dragging-and-dropping the book maps into the submap in the DITA Map view. For more information, see [Adding Links to an Eclipse Help Submap Using the DITA Map View](#) on page 89.

However, you can also add the links manually in Text mode. The most common format you will use is as follows:

```
<topichead>
  <topicmeta>
    <navtitle>Product Name Programmer's Guide, N.N</navtitle>
  </topicmeta>
  <mapref href="elm123456789.ditamap" format="ditamap"/>
</topichead>
```

For more complex cases where the linking from this Eclipse submap must be done in a described, specific way, see the following:

- [Linking to a Bookmap When There is Only One Book in the Plugin](#) on page 90
- [Adding Subcategories to Your Plugin for Workbench Help](#) on page 91
- [Linking to a Non-DITA Topic Set](#) on page 92
- [Linking to a Non-DITA Topic Set that Must Be Wrapped](#) on page 92
- [Inserting a Plugin for a Sub-Product Inside a Product Plugin](#) on page 93
- [Linking to a PDF File](#) on page 94

You will most likely not need these cases.

8. Save the file.

## Adding Links to an Eclipse Help Submap Using the DITA Map View

The easiest way to add links to an Eclipse Help Submap it to use the DITA Map view to drag and drop the bookmaps into your submap.

### Procedure

1. Use the **Search** view to locate the map.
2. Right-click on the map in the **Search Results** view and select **Lock**.  
 **NOTE:** If the map is already locked (appears in bold blue text in the **Search Results** tab, skip this step.)
3. Double-click the map in the **Search Results** view.  
The map opens in the DITA Map view.
4. Locate the bookmaps using the Search view.
5. Drag and drop the bookmap from the Search Results view into your submap in the DITA Map editor.  
Links to the bookmaps appear at the bottom of the submap.
6. Open the map in the oXygen DITA Map editor in Text mode.

7. Move the links after the following line:

```
<anchor id="top"/>
```

8. Format the links.

The most common format you will use is as follows:

```
<topichead>
  <topicmeta>
    <navtitle>Product Name Programmer's Guide, N.N</navtitle>
  </topicmeta>
  <mapref href="elm123456789.ditamap" format="ditamap"/>
</topichead>
```

## Linking to a Bookmap When There is Only One Book in the Plugin

When you only have one book in your Workbench Help plugin, you must modify your Eclipse Help Submap to prevent duplicate titles from showing up in Workbench Help.

### Procedure

1. Lock your map.
2. Open your map in the oXygen DITA Map editor.
3. Select the **Text** tab in the lower left corner of the editor.
4. Change the value of the first **<navtitle>** element in the map to the name of your document.  
This is normally the product name and version, but to prevent unnecessary nesting and duplicate titles you must change it.
5. Add a **<mapref>** element between the following lines:

```
<anchor id="top"/>
<anchor id="bottom"/>
```

Do not add a **<topichead>** element around your **<mapref>** element.

6. Save and release the file.

### Example

```
<topichead ixia_locid="2">
  <topicmeta>
    <navtitle ixia_locid="3">Wind River Technical Writer's Guide, 1.0</navtitle>
  </topicmeta>
  <anchor id="top"/>
  <mapref href="elm123456789.ditamap" format="ditamap"/>
  <anchor id="bottom"/>
</topichead>
```

## Adding Subcategories to Your Plugin for Workbench Help

If you require subcategories in your plugin for Workbench Help, you must add a navigation title for each subcategory and links to the documents in the subcategory.

### Procedure

1. Lock your map.
2. Open your map in the oXygen DITA Map editor.
3. Select the Text tab in the lower left corner of the editor.
4. Add a `<topichead>` element.

**NOTE:** The `<topichead>` element may appear before, after, or between the `<topichead>` elements containing links to other topic sets, but it must be between the following lines:

```
<anchor id="top"/>
<anchor id="bottom"/>
```

5. Add a `<topicmeta>` element inside the `<topichead>` element
6. Add a `<navtitle>` element inside the `<topicmeta>` element.

```
<topichead>
  <topicmeta>
    <navtitle>Getting Started</navtitle>
  </topicmeta>
```

7. Add a `<mapref>` element after the `<topicmeta>` element.
8. Add an `@href` attribute pointing to the bookmap for the subcategory.

```
<topichead>
  <topicmeta>
    <navtitle>Getting Started</navtitle>
  </topicmeta>
  <mapref href="CMS-ID.ditamap"/>
</topichead>
```

9. Save and release the file.

## Linking to a Non-DITA Topic Set

To link to a non-DITA topic set, you must add a <navref> element to your Eclipse Help Submap.

### Procedure

1. Lock your map.
2. Open your map in the oXygen DITA Map editor.
3. Select the Text tab in the lower left corner of the editor.
4. Add a <navref> element.



**NOTE:** The <navref> element may appear before, after, or between the <topichead> elements containing links to other topic sets, but it must be between the following lines:

```
<anchor id="top"/>  
<anchor id="bottom"/>
```

5. Add a @mapref attribute to the <navref> element.

The format is as follows:

```
<navref mapref="vxworks_mils_xml_config_ref/html/toc.ditamap"/>
```

The .ditamap extension is turned into .xml in an output XML file and a toc element is added to plugin.xml.

6. Save and release the file.

## Linking to a Non-DITA Topic Set that Must Be Wrapped

If you are linking to a non-DITA topic set that has more than one element at the top or more than one topic set at the top level, you must wrap the topic set and provide a navigation title.

### Procedure

1. Lock your map.
2. Open your map in the oXygen DITA Map editor.
3. Select the Text tab in the lower left corner of the editor.
4. Add a <tocref> element.



**NOTE:** The `<tocref>` element may appear before, after, or between the `<topichead>` elements containing links to other topic sets, but it must be between the following lines:

```
<anchor id="top"/>  
<anchor id="bottom"/>
```

- a) Add an `@href` attribute and set its value to point to the title page of the document.
- b) Add a `@scope` attribute and set its value to **external**.
- c) Add a `@format` attribute and set its value to **html**.

```
<topicref href="wr_diab_compiler_users_guide_mips/html/title.html" scope="external"  
format="html">  
</topicref>
```

5. Add a `<navref>` element inside the `<topicref>` element.
6. Add a `@mapref` attribute to the `<navref>` element.

The format is as follows:

```
<navref mapref="vxworks_mils_xml_config_ref/html/toc.ditamap"/>
```

The `.ditamap` extension is turned into `.xml` in an output XML file and a `toc` element is added to `plugin.xml`.

7. Save and release the file.

#### Example

```
<topicref href="wr_diab_compiler_users_guide_mips/html/title.html" scope="external"  
format="html">  
    <topicmeta>  
        <navtitle>Wind River Diab Compiler C Library Reference, 5.9.3</navtitle>  
    </topicmeta>  
    <navref mapref="wr_diab_compiler_c_lib_reference/html/toc.ditamap"/>  
</topicref>
```

## Inserting a Plugin for a Sub-Product Inside a Product Plugin

To insert a sub-product plugin inside a Workbench Help plugin, you must define a new anchor.

#### Procedure

1. Lock your map.
2. Open your map in the oXygen DITA Map editor.
3. Select the **Text** tab in the lower left corner of the editor.
4. Add an `<anchor>` element.



**NOTE:** The `<anchor>` element may appear before, after, or between the `<topichead>` elements containing links to other topic sets, but it must be between the following lines:

```
<anchor id="top"/>  
<anchor id="bottom"/>
```

- 
5. Add an `@id` attribute to the `<anchor>` element and set an appropriate value.

The anchor ID must be unique within the ID values in the Eclipse Help submap. It should be an underscore-separated ID that is descriptive of the content that belongs in the plugin.

For example, for Wind River Linux Carrier Grade Profile:

```
<anchor id="wr_linux_cgpp"/>
```

An anchor is created in the output XML file.

6. Save and release the file.

## Linking to a PDF File

To link to a PDF file, you must provide the relative path to the PDF file in the output folder and a navigation title.

### Procedure

1. Lock your map.
2. Open your map in the oXygen DITA Map editor.
3. Select the **Text** tab in the lower left corner of the editor.
4. Add a `<topicref>` element to your map.



**NOTE:** The `<topicref>` element may appear before, after, or between the `<topichead>` elements containing links to other topic sets, but it must be between the following lines:

```
<anchor id="top"/>  
<anchor id="bottom"/>
```

- 
- a) Add an `@href` attribute and set its value to point to the relative path to the PDF file in the output folder.
  - b) Add a `@scope` attribute and set its value to **external**.

- c) Add a **@format** attribute and set its value to **pdf**.

```
<topicref href="wr_diab_compiler_dmake/dmake_4.1.pdf" format="pdf" scope="external">  
</topicref>
```

5. Add a **<topicmeta>** element inside the **<topicref>** element.
6. Add a **<navtitle>** element inside the **<topicmeta>** element and add the title of the PDF document.
7. Save and release the file.

#### Example

```
<topicref href="wr_diab_compiler_dmake/dmake_4.1.pdf" format="pdf" scope="external">  
    <topicmeta>  
        <navtitle>dmake Command Reference, 4.1</navtitle>  
    </topicmeta>  
</topicref>
```

#### Postrequisites

You must manually copy the PDF to the output folder.



# 11

## *Bookmaps*

<a href="#">Adding Product Information to a Bookmap</a>	97
<a href="#">Adding Taxonomy Metadata to a Bookmap</a>	99
<a href="#">Changing the Bundle Name</a>	100
<a href="#">Updating the Metadata for Inter-Book Linking</a>	105
<a href="#">Adding an Edition to a Bookmap</a>	106
<a href="#">Adding Chapters to Your Bookmap</a>	107
<a href="#">Adding Parts to Your Bookmap</a>	108

### **Adding Product Information to a Bookmap**

The product information is combined with the title of the book to create the title displayed in the HTML and PDF output. It can also be used for search.

#### **Procedure**

1. Add a `<bookmeta>` element to your bookmap.

The `<bookmeta>` element is a container element for information about the publication. The `<bookmeta>` element must follow the `<booktitle>` element.

For example:

```
<booktitle>
    <mainbooktitle>Wind River Firewall Programmer's Guide</mainbooktitle>
</booktitle>
<bookmeta></bookmeta>
```

2. Add a `<prodinfo>` element inside the `<bookmeta>` element.

The **<prodinfo>** element is the container element for the **<prodname>** and **<vrmlist>** elements.

```
<booktitle>
    <mainbooktitle>Wind River Firewall Programmer's Guide</mainbooktitle>
</booktitle>
<bookmeta>
    <prodinfo></prodinfo>
</bookmeta>
```

3. Add a **<prodname>** element inside the **<prodinfo>** element and add the name of the product.

The product name must exactly match the first part of the **<mainbooktitle>**, but without the documentation type.

```
<booktitle>
    <mainbooktitle>Wind River Firewall Programmer's Guide</mainbooktitle>
</booktitle>
<bookmeta>
    <prodinfo>
        <prodname>Wind River Firewall</prodname>
    </prodinfo>
</bookmeta>
```

4. Add a **<vrmlist>** element inside the **<prodinfo>** element.

```
<booktitle>
    <mainbooktitle>Wind River Firewall Programmer's Guide</mainbooktitle>
</booktitle>
<bookmeta>
    <prodinfo>
        <prodname>Wind River Firewall</prodname>
        <vrmlist></vrmlist>
    </prodinfo>
</bookmeta>
```

5. Add a **<vrm>** element to the **<vrmlist>** element.

6. Add **@version** and **@release** attributes to the **<vrm>** element.

The **@version** attribute is required by the DTD, but since Wind River uses release numbers, we have chosen to use the **@release** attribute instead. The **@version** attribute must be present, but empty.

For example:

```
<booktitle>
    <mainbooktitle>Wind River Firewall Programmer's Guide, 6.9</mainbooktitle>
</booktitle>
<bookmeta>
    <prodinfo>
        <prodname>Wind River Firewall</prodname>
        <vrmlist>
            <vrm version="" release="6.9"/>
        </vrmlist>
    </prodinfo>
</bookmeta>
```

## Adding Taxonomy Metadata to a Bookmap

Taxonomy metadata is necessary for classification and search on the Product Documentation site. The Taxonomy Terms view in the Ixiasoft CCMS client lets you add taxonomy data to bookmaps that are going to be imported to the Product Documentation site.

In addition to the product/version taxonomies, you must ensure that all appropriate taxonomies (such as visibility) are applied to your bookmap. You can apply taxonomies to more than one bookmap at a time.

When you add taxonomies, the taxonomy tool adds several <category> elements to the <bookmeta> element of your bookmap.

---

 **NOTE:** Do not manually modify these elements. If you need to add or delete a taxonomy from your bookmap, use the taxonomy tool.

---

### Procedure

1. If it is not already open, open the Taxonomy Terms view.

Select **Window > Show View > Other > IXIASOFT CCMS-General > Taxonomy Terms**.

2. Locate the desired bookmap or bookmaps.

For example, perform a search, open a map, or display them in a view. You can add the same metadata to more than one bookmap at a time, so if you have several bookmaps in your documentation set, search for the Eclipse Help map that contains them.

3. Verify that your bookmap is in the Authoring:work state.

---

 **NOTE:** The bookmap does not need to be locked.

---

4. Drag the bookmaps into the Documents pane in the Taxonomy Terms view.

5. In the Taxonomy pane of the Taxonomy Terms view, expand the list of terms to find the values you want to add to your bookmap and select one or more taxonomy terms.

To select more than one term, use **CTRL+CLICK**.

You need to add terms from the following taxonomies:

Taxonomy	Description	Examples
Products	Specifies which product, product version, and category the document belongs to.	<ul style="list-style-type: none"><li>• Wind River Linux &gt; Wind River Linux LTS 18 &gt; Getting Started</li><li>• VxWorks &gt; VxWorks 7 SR0600 &gt; Networking &gt; DHCP</li></ul>

Taxonomy	Description	Examples
Category	Specifies the category for the content.	<ul style="list-style-type: none"><li>• Getting Started</li><li>• Release Notes</li><li>• Connectivity</li><li>• Kernel &amp; Application Development</li></ul>
Content Type	Specifies if the content is featured content or discontinued content.	<ul style="list-style-type: none"><li>• Featured Content</li><li>• Discontinued Content</li></ul>
Format	Specifies the type of document.	<ul style="list-style-type: none"><li>• App Note</li><li>• Document</li><li>• Tutorial</li><li>• Video</li></ul>
Visibility	Specifies if the document is customer visible or not.	<ul style="list-style-type: none"><li>• External</li><li>• Internal</li></ul>

6. Right-click the term(s) and select **Add Term(s) to All Documents**.
7. If a message appears requesting to lock the documents, click **OK**.
8. Review the changes made to your bookmaps and then release them.

## Changing the Bundle Name

### The Bundle Name

On the Product Documentation site, the file name is also referred to as the bundle name. By default, the bundle name is the bookmap ID.

Since the bookmap ID may not change from one version of a book to another, when the document is imported to the Product Documentation site, the newer version may overwrite the older version.

For example, you finish work on version 1.0 of Product X. You then upload it to the Product Documentation site and publish the book in the Ixiasoft CCMS. Since the bookmap ID does not change unless you branch, when you continue working on the mainline trunk in the Ixiasoft CCMS for version 2.0 of Product X, the IDs for both versions are the same. If you then upload version 2.0 to the Product Documentation site, the 2.0 version will overwrite the 1.0 version.

To prevent this, you must add an `<othermeta>` element to your bookmap that changes the bundle name. Changing the bundle name also creates human-readable URLs on the Product Documentation site.

The syntax is as follows:

```
<othermeta name="bundle" content="bundleName"/>
```

Where *bundleName* is document title, including release number, using the following rules.

- it must not contain special characters
- spaces in the title must be replaced by underscores
- periods in the version number must be replaced by underscores
- do not include edition numbers

If you want your document to replace a document that already exists on the Product Documentation site, the bundle name must exactly match the bundle name of the document on the Product Documentation site. For more information, see [Replacing an Existing Doc with an Updated Version](#) on page 102.

If your bookmap uses conditional text to create several variations of the document, you must use conditional text and add an `<othermeta>` element for each variation of the document. For more information, see [Changing the Bundle Name Using Conditional Text](#) on page 104.

## Changing the Bundle Name for a New Document

For a new document, or a document that is not replacing a document already posted to the Product Documentation site, you must add a bundle name that consists of the document title, including the release number.

---

**NOTE:** If you are making updates to a document and want it to replace the version already on the Product Documentation site, see [Replacing an Existing Doc with an Updated Version](#) on page 102.

If your bookmap uses conditional text to create several variations of the document, see [Changing the Bundle Name Using Conditional Text](#) on page 104.

---

### Procedure

1. Add an `<othermeta>` element to your bookmap between the `<prodinfo>` element and the `<bookid>` element.
2. Set the `@name` attribute to **bundle**.
3. Set the `@content` attribute to the document title, including release number, using the following rules:
  - it must not contain special characters
  - spaces in the title must be replaced by underscores
  - periods in the version number must be replaced by underscores
  - do not include edition numbers

For example, the `<othermeta>` entry for the *Wind River Titanium Cloud Introduction, 17.06* guide is as follows:

```
<prodinfo>
  <prodnname>Wind River Titanium Cloud</prodnname>
  <vrmlist>
    <vrm release="17.06" version=" " />
  </vrmlist>
</prodinfo>
<othermeta name="bundle" content="wind_river_titanium_cloud_introduction_17_06"/>
<bookid>
  <edition>Patch 0001</edition>
</bookid>
```

## Replacing an Existing Doc with an Updated Version

If you want your updated document to replace a document that already exists on the Product Documentation site, the bundle name must match the bundle name of the existing document exactly.

When some of the content was migrated from the old Knowledge Library to the new Product Documentation site, the bundle names did not conform to the standard; for example:

Wind\_River\_Introduction\_to\_Titanium\_Cloud\_17.06\_1

Even though it does not conform to the standard, the bundle name in your updated document must still match this bundle name.

**NOTE:** An exception to this rule is when the bundle name contains an edition number. For more information, see [Replacing Bundles that Contain Edition Numbers](#) on page 103.

### Procedure

1. Add an `<othermeta>` element to your bookmap between the `<prodinfo>` element and the `<bookid>` element.
2. Set the `@name` attribute to **bundle**.
3. Find the bundle name of the existing document:
  - a) On the Product Documentation site, click on your user name and select **Admin**.
  - b) Click on the **Publications** tile.
  - c) Do a search for your document, using the document title.
  - d) Copy the file name.

The file name is the same thing as the bundle name.

Workbench 4 Getting Started, Edition 4  
File name: [HTML-Workbench\\_4\\_Getting\\_Startet\\_Edition\\_4\\_1](#)

4. Set the @content attribute to the document title, by pasting in the file name you copied in the previous step.

## Replacing Bundles that Contain Edition Numbers

When content was migrated from the old Knowledge Library (KL) to the new Product Documentation site, some of the bundle names contained edition numbers. Since the edition number is likely to change, and the bundle name becomes part of the URL on the Product Documentation site, we do not want to include it in the bundle name.

If the bundle name for the document you want to replace on the Product Documentation site contains an edition number, keeping the same bundle name can be confusing. For example, a document migrated from the old KL may have a bundle name similar to the following:

**VxWorks\_653\_Platform\_Release\_Notes\_3.0.1.1\_Edition\_3\_1**

If you update the above document to edition 4, and use the same bundle name, it would be confusing for the customer to be looking at edition 4 of the document, but have a URL showing **Edition\_3\_1**.

In this case, you would create a bundle name that does not include the edition number. Because the bundle names are different, the updated version will not replace the old version on the Product Documentation site. Once you push your document and check it in both staging and production, you must manually delete the old version of the document.

### Procedure

1. Add an **<othermeta>** element to your bookmap between the **<prodinfo>** element and the **<bookid>** element.
2. Set the @name attribute to **bundle**.
3. Set the @content attribute to the document title, including release number, using the following rules.
  - it must not contain special characters
  - spaces in the title must be replaced by underscores
  - periods in the version number must be replaced by underscores
  - do not include the edition number

For example, the **<othermeta>** entry for the *Wind River Titanium Cloud Introduction, 17.06* guide is as follows:

```
<prodinfo>
    <prodname>Wind River Titanium Cloud</prodname>
    <vrmlist>
        <vrm release="17.06" version=" " />
    </vrmlist>
</prodinfo>
<othermeta name="bundle" content="wind_river_titanium_cloud_introduction_17_06"/>
<bookid>
    <edition>Patch 0001</edition>
</bookid>
```

## Postrequisites

Once you push your document and check it in both staging and production, you must manually delete the old version of the document.

## Changing the Bundle Name Using Conditional Text

If you have a document where there is one source document in DITA that is used to create several variations of the same guide, you need to use conditions to create a different bundle name for each variation of your document.

### Procedure

1. Add an `<othermeta>` element to your bookmap for each variation of the document.  
The `<othermeta>` elements must be between the `<prodinfo>` element and the `<bookid>` element.
2. For each `<othermeta>` element, set the `@name` attribute to **bundle**.
3. For each `<othermeta>` element, set the `@content` attribute to the document title, including release number.
4. Set the appropriate conditions.

For example:

```
<othermeta product="tis_cpe_sx" name="bundle"
content="wind_river_titanium_server_installation_for_simplex_systems_16_10"/>
<othermeta product="tis_cpe_dx" name="bundle"
content="wind_river_titanium_server_installation_for_duplex_systems_16_10"/>
<othermeta product="tis_ceph" name="bundle"
content="wind_river_titanium_server_installation_for_systems_with_dedicated_storage_
16_10"/>
<othermeta product="tis_lvm" name="bundle"
content="wind_river_titanium_server_installation_for_systems_with_controller_storage
_16_10"/>
```

## Postrequisites

If you are creating a new document that is not replacing an existing document on the Product Documentation site, you are finished.

However, if you are updating a document that you want to replace an existing document on the Product Documentation site, there is an issue.

By default, Zoomin uses the ID of the bookmap as the filename when processing files. When a conditionalized document is processed, during the transform, the ID of the ditaval file is appended to the end of the bookmap ID to prevent one version of the doc from overwriting another. For example:

Document	File Name
<i>Wind River Titanium Cloud Installation for Duplex Systems</i>	<code>rwg1463671602892_zjo1490647488025</code>
<i>Wind River Titanium Cloud Installation for Simplex Systems</i>	<code>rwg1463671602892_vud1490647488368</code>

When you include an <othermeta> element to change the bundle name, the value you add in the @content attribute replaces the bookmap ID. However, the ditaval ID is still appended to the filename during transform. For example:

Document	File Name
<i>Wind River Titanium Cloud Installation for Duplex Systems</i>	<code>wind_river_titanium_cloud_installation_for_duplex_systems_18_03_zjo1490647488025</code>
<i>Wind River Titanium Cloud Installation for Simplex Systems</i>	<code>wind_river_titanium_cloud_installation_for_simplex_systems_18_03_vud1490647488368</code>

Even if you make the bundle name in your bookmap match the bundle name on the Product Documentation site, it will not be the same when you push the doc to the staging or production because the transforms add the ditaval ID. Therefore, the document you just created will not overwrite the older version. Once you push your document and check it in both staging and production, you must manually delete the old version of the document.

## Updating the Metadata for Inter-Book Linking

To ensure that inter-book links work properly, you must add an <othermeta> element to your bookmap that points to the proper version on the Product Documentation site.

 **NOTE:** You must also update the @scope attribute for your cross-references. For more information, see [Updating @scope Attributes in Cross-References Using an XSLT Script](#) on page 199.

### Procedure

1. Add an <othermeta> element between the <prodinfo> element and the <bookid> element.
2. Set the @name attribute to **facets**.
3. Locate your product version in the Taxonomy pane of the Taxonomy Terms view and note the value in the Vocabulary Term Value column for your product version.

 **NOTE:** You do not want the category, only the product version; for example, VxWorks 7 SR0540, Linux LTS 18, and so forth.

4. Set the value of the @content attribute to the Vocabulary Term Value for your product version.

For example, for a document included in the VxWorks 7 SR0540 release, the <othermeta> element would be as follows:

```
<bookmeta>
  <prodinfo>
    <prodname>VxWorks 7 SR0540</prodname>
    <vrmlist>
      <vrm version="3" release="0"/>
    </vrmlist>
  </prodinfo>
  <othermeta name="facets" content="version=os_vxworks_7_sr0540"/>
  <bookid>
    <edition>Edition 2</edition>
  </bookid>
```

## Adding an Edition to a Bookmap

The Wind River standard includes an edition number for all editions of a topic set except the first.

### Procedure

1. Add a <bookid> element to your map, after the <prodinfo> element.

An <edition> element is automatically added for you.

```
<bookmeta>
  <prodinfo>
    <prodname>Product Name such as VxWorks</prodname>
    <vrmlist>
      <vrm version="" release="0.0"/>
    </vrmlist>
  </prodinfo>
  <bookid>
    <edition></edition>
  </bookid>
```

2. Update the edition number.

```
<bookmeta>
  <prodinfo>
    <prodname>Product Name such as VxWorks</prodname>
    <vrmlist>
      <vrm version="" release="0.0"/>
    </vrmlist>
  </prodinfo>
  <bookid>
    <edition>Edition 2</edition>
  </bookid>
```

## Adding Chapters to Your Bookmap

To get your PDF output to generate properly you must add chapters to your bookmap in different ways, according to your content.

### Procedure

Add a `<chapter>` element to your bookmap using one of the following methods:

Type of Chapter	Method
<b>Multi-topic chapters</b>	<p>Use the <code>&lt;topicmeta&gt;</code> and <code>&lt;navtitle&gt;</code> elements to add a title to your topic set.</p> <p>Add a <code>&lt;mapref&gt;</code> element containing a <code>@format</code> attribute set to <b>ditamap</b>.</p> <p>Add the map name including the <b>.ditamap</b> extension to the <code>@href</code> attribute.</p> <p>For example:</p>

```
<chapter>
    <topicmeta>
        <navtitle>CAN Bus Sample Application</navtitle>
    </topicmeta>
    <mapref format="ditamap"
href="yco1466106499351.ditamap"/>
</chapter>
```

**Multi-topic chapters where the DITA map for the chapter consists of a set of nested `<topicref>` elements as the ONLY item**

For example, if your DITA map looks similar to the following:

```
<map-wr id="xth1467124390476" xml:lang="en-us">
    <title ixia_locid="1">Accessing Telemetry Data Through Web
Applications</title>
    <topicref href="itn1467124645835.xml" ixia_locid="5">
        <topicref href="jxt1467124731693.xml" ixia_locid="6"/>
        <topicref href="gag1467124761026.xml" ixia_locid="7"/>
        <topicref href="ttc1467124784306.xml" ixia_locid="8"/>
        <topicref href="khz1467124814367.xml" ixia_locid="9"/>
    </topicref>
</map-wr>
```

Add an `@href` attribute inside the `<chapter>` element, set the `@format` attribute to **ditamap**.

Do not include a navtitle.

For example:

```
<chapter href="xth1467124390476.ditamap" format="ditamap"/>
```

---

Type of Chapter	Method
Single-topic chapters	Add an @href attribute directly in the <chapter> element. Do not include a navtitle. For example:

```
<chapter href="ekn1234567890.xml"/>
```

---

## Adding Parts to Your Bookmap

To configure a book to use parts, you must surround each group of chapters with a <part> element.

### Procedure

1. Add a <part> element to your map.
2. Add <topicmeta> and <navtitle> elements to include the part title.
3. Add <chapter> and <mapref> elements for each chapter of your book.

### Example

```
<part>
  <topicmeta>
    <navtitle>Getting Started</navtitle>
  </topicmeta>
  <chapter>
    <topicmeta>
      <navtitle>Ixiasoft CCMS Client Overview</navtitle>
    </topicmeta>
    <mapref format="ditamap" href="dia1396907958793.ditamap"/>
  </chapter>
  <chapter>
    <topicmeta>
      <navtitle>Authoring Environment Setup for the Ixiasoft CCMS</navtitle>
    </topicmeta>
    <mapref format="ditamap" href="dia1396908861015.ditamap"/>
  </chapter>
</part>
```

# 12

## *Topics Submaps*

<a href="#">Adding Existing Topics to a Map</a>	109
<a href="#">Creating Topic Stubs</a>	110
<a href="#">Creating a Hierarchy in a Submap Using Topic References</a>	113
<a href="#">Creating a Navigation Structure in a Submap Using Topic Headings</a>	114
<a href="#">Creating Topic Groups in Submaps</a>	115
<a href="#">Using the @chunk Attribute in a Submap</a>	117

### **Adding Existing Topics to a Map**

Because topic IDs are not human-readable, it is easier to add existing topics to your map by dragging and dropping them into your map in the DITA Map view and then customizing the map in the editor.

#### **Procedure**

1. Lock the map.
2. Open the map in the DITA Map view.
3. Use the Search view to locate the topic.
4. Drag and drop the topic from the Search Results view to the map in the DITA Map view.

## Creating Topic Stubs

During documentation planning, you can easily add a topic stub to your map as a placeholder.

### Procedure

1. Use the Search view to locate the map.
2. Right-click the map in the Search Results view and select **Lock**.

---

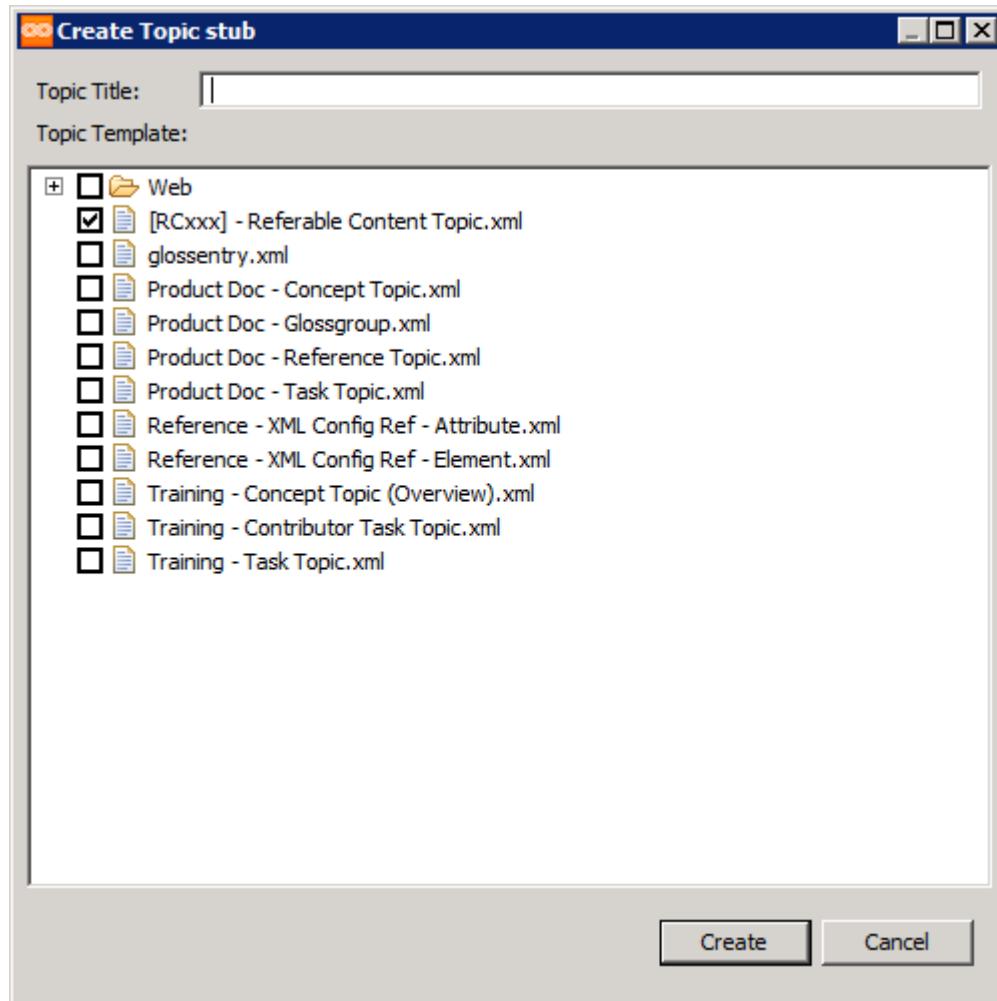
 **NOTE:** If the map is already locked (appears in bold blue text in the Search Results view, skip this step.)

---

3. Double-click the map in the Search Results view to open it in the DITA Map view.
4. Click the **Create Topic Stub** icon in the DITA Map view toolbar.



The Create Topic Stub dialog box appears.



5. Type the title for your topic.

Wind River standards require the following for topic titles:

Concepts

Use a noun phrase, for example, *The Topic Cloud* or *About Files in the Ixiasoft CCMS*

Tasks

Use a gerund, for example, *Adding Topics to a Map*.

References

Use a noun phrase, including "Reference" if the content is not clear without it. For example, *Parameters for make* or *Security Command Reference*.

6. Select the type of topic.

Topic Type	Description
[RCxxx] - Referable Content Topic.xml	Used for conrefs. For more information, see <a href="#">Guidelines for Creating Referable-Content Topics</a> on page 141.

Topic Type	Description
<b>glossentry.xml</b>	Not currently used by Technical Publications.
<b>Product Doc - Concept Topic.xml</b>	Use the concept topic type to create support or overview information that users need to complete tasks successfully. They respond to the user's question, "What is this about?"
<b>Product Doc - Glossgroup.xml</b>	Use a glossary group topic type to create a glossary for your document.
<b>Product Doc - Reference Topic.xml</b>	Use a reference topic type to provide reference material that is essential for the successful performance of technical tasks.
<b>Product Doc - Task Topic.xml</b>	Use a task topic type to create a step-by-step procedure for completing a task. A task helps the users to answer the question "How do I?"
<b>Reference - XML Config Ref - Attribute.xml</b>	Use to create a topic containing XML configuration attribute definitions. The attribute definitions are included as a part of an XML configuration reference document.
<b>Reference - XML Config Ref - Element.xml</b>	Use to create a topic containing XML configuration element definitions. The element definitions are included as a part of an XML configuration reference document.
<b>Training - Concept Topic (Overview).xml</b>	Use the training concept topic type only for a training lab, to provide standard overview information that the student must know and understand to complete the training lab.
<b>Training - Contributor Task Topic.xml</b>	Not well-defined or used at present.
<b>Training - Task Topic.xml</b>	Use the training task topic type only for training labs, to describe steps the student must perform to complete a principal task in the lab.

7. Click **Create**.

The topic stub is added to your map as a <topicref>.

#### Postrequisites

When you are ready to start writing your topic, you must generate the topic from the topic stub. For more information, see [Generating Topics from Stubs](#) on page 122.

## Creating a Hierarchy in a Submap Using Topic References

Nesting `<topicref>` elements in one another creates a hierarchical structure for your readers. You can also use this hierarchy to represent your table of contents, site map, and online navigation.

### Procedure

Include `<topicref>` elements for the topics you want to appear as subtopics inside the `<topicref>` element for your parent topic.

For example:

```
<topicref href="dia1400172931688.dita" collection-type="family">
  <topicref href="dia1400188729316.dita" ixia_locid="6"/>
  <topicref href="rec1384889855768.xml" ixia_locid="7"/>
  <topicref href="rec1384889856423.xml" ixia_locid="8"/>
  <topicref href="dia1400191998074.xml" ixia_locid="9"/>
  <topicref href="rec1384889858522.xml" ixia_locid="10"/>
  <topicref href="rec1384885228985.xml" ixia_locid="10"/>
</topicref>
```

For more information about adding `<topicref>` elements to your map, see [Adding Existing Topics to a Map](#) on page 109.

The resulting hierarchy in the table of contents appears as follows:

- Deep Packet Inspection**
  - [Configuring INP with Hyperscan](#)
  - [Hyperscan Example Programs](#)
    - [Downloading the Hyperscan Example Programs](#)
    - [Building and Running Hyperscan Example 1](#)
    - [Building and Running Hyperscan Example 2](#)
    - [Building and Running Hyperscan Example 3](#)
    - [Building and Running Hyperscan Example 4](#)
    - [Example: Integrating Hyperscan](#)

- QEMU**
  - [Running INP on a QEMU Simulator](#)
  - [Using QEMU](#)

Another benefit of nesting `<topicref>` elements is that links to the subtopics are automatically created in the parent topic. For example:

## Hyperscan Example Programs

Hyperscan is bundled with four example programs.

After you build your project, the example programs are located at:

`projectDir/build/hyperscan/share/libhs/examples`

Example 1 is a C program that searches a file for a pattern on the command line. Examples 2, 3, and 4 are C++ programs that deal with a pattern file and captured data.

The compilation instructions for each of the example programs are included in the source file for the program. The program also provides usage instructions. For example:

```
$ ./example1
Usage: example1 <pattern> <input file>

$ ./example2
Usage: example2 <pattern file> <pcap file>

$ ./example3
Usage: example3 [-n repeats] <pattern file> <pcap file>

$ ./example4
Usage: example4 <pattern file> <pcap file>
```

---

### [Downloading the Hyperscan Example Programs](#)

The source code for the Hyperscan examples is available on the Wind River Online Support Web site.

#### [\*\*Building and Running Hyperscan Example 1\*\*](#)

Example 1 searches a file for a pattern you type on the command line.

#### [\*\*Building and Running Hyperscan Example 2\*\*](#)

Example 2 searches a file for a pattern in a pattern file, rather than text entered on the command line.

#### [\*\*Building and Running Hyperscan Example 3\*\*](#)

Example 3 analyzes the input file and provides some statistics.

#### [\*\*Building and Running Hyperscan Example 4\*\*](#)

Example 4 combines some of the features of Examples 2 and 3.

#### [\*\*Example: Integrating Hyperscan\*\*](#)

You can integrate the Hyperscan scanner with INP.

#### **Related information**

[Configuring INP with Hyperscan](#)

## Creating a Navigation Structure in a Submap Using Topic Headings

You can use the `<topichead>` element to create a navigation structure for related topics in your map. Use a `<navtitle>` element within a `<topicmeta>` element to enter the title text.

#### **Procedure**

1. Add a `<topichead>` element to your DITA map.
2. Inside the `<topichead>` element, add a `<topicmeta>` element:

```
<topichead>
  <topicmeta></topicmeta>
<topichead>
```

3. Inside the <topicmeta> element, add a <navtitle> element and give it an appropriate title:

```
<topichead>
  <topicmeta>
    <navtitle>L3 Forwarding</navtitle>
  </topicmeta>
</topichead>
```

4. Add <topicref> elements to your map:

```
<topichead>
  <topicmeta>
    <navtitle>L3 Forwarding</navtitle>
  </topicmeta>
  <topicref href="dia1400172931688.dita" ixia_locid="6"/>
  <topicref href="dia1400188729316.dita" ixia_locid="6"/>
  <topicref href="rec1384889855768.xml" ixia_locid="7"/>
  <topicref href="rec1384889856423.xml" ixia_locid="8"/>
</topichead>
```

The resulting structure is in the table of contents is as follows:

- L3 Forwarding**
  - [L3 Forwarding Example: Overview](#)
  - [Building and Booting the INP Application](#)
  - [Configuring the INP Application](#)
  - [Running the L3 Forwarding Use Case Example](#)
- L4 Termination**
  - [L4 Termination Example: Overview](#)
  - [Building and Booting the INP Application](#)
  - [Configuring the INP Application](#)
  - [Running the L4 Termination Example](#)

## Creating Topic Groups in Submaps

You can use topic groups to create collections of topic references.

### Procedure

1. Add a <topicgroup> element to your DITA map.
2. Set a value for the @collection-type attribute.

The @collection-type attribute describes how links relate to each other in the topics included in the group.

---

Value	Description
<b>unordered</b>	Indicates that the order of the child topics is not significant.
<b>sequence</b>	Indicates that the order of the child topics is significant; links to previous and next topics will be created at the bottom of your topics.

---

Value	Description
<b>choice</b>	Indicates that one of the children should be selected.
<b>family</b>	Represents a tight grouping in which each of the referenced topics not only relates to the current topic but also relate to each other. Related links to all the other topics in the group will be created at the bottom of your topics.

---

For example:

```
<topicgroup collection-type="family"></topicgroup>
```



**NOTE:** The @collection-type attribute can also be added to **<topichead>** and **<topicref>** elements.

---

3. Add **<topicref>** elements to your map:

```
<topicgroup collection-type="family">
  <topicref href="dia1400172931688.dita" ixia_locid="6"/>
  <topicref href="dia1400188729316.dita" ixia_locid="6"/>
  <topicref href="rec1384889855768.xml" ixia_locid="7"/>
  <topicref href="rec1384889856423.xml" ixia_locid="8"/>
  <topicref href="dia1400191998074.xml" ixia_locid="9"/>
  <topicref href="rec1384889858522.xml" ixia_locid="10"/>
</topicgroup>
```

Related links are automatically created in your topic according to the value you set for the **@collection-type** attribute. For example, the following image shows links that are created when the **@collection-type** attribute is set to **family**:

## inp-addvrf

You can use this utility to add a VRF to the system.

### Usage

```
$ inp-addvrf -h
Usage: inp-addvrf <vrf> <pid>
      <vrf>   - VRF number to map to
      <pid>   - PID to map to VRF
```

### Example

The following command maps an LXC container as VRF 2:

```
$ inp-addvrf 2 2283
```

---

### Related information

[inp-delvr](#)  
[inp-ifnetns](#)  
[inp-listvrf](#)  
[inp-setns](#)  
[inp-setup](#)  
[inp-unshare](#)

## Using the @chunk Attribute in a Submap

You can use the **@chunk** attribute to force a group of nested topics to be combined into a single HTML page in the output.

The **@chunk** attribute should only be used for short, nested topics that the customer would benefit from seeing on the same page. For example, a workflow topic and a few short, one- to three-step tasks that demonstrate the different methods for performing the same task.

The resulting HTML page should be short and limited in scope – ideally one screen long, two at the very most.

---

 **NOTE:** Do not use the **@chunk** attribute for concept or reference topics. These topics provide section elements that may be used instead.

---

### Procedure

1. Set the value of the **@chunk** attribute to **to-content** in the **<topicref>** element of the parent topic.

For example:

```
<topicref chunk="to-content" href="dia1400172931688.dita">
  <topicref href="dia1400188729316.dita" ixia_locid="6"/>
  <topicref href="rec1384889855768.xml" ixia_locid="7"/>
  <topicref href="rec1384889856423.xml" ixia_locid="8"/>
</topicref>
```

2. Set the value of the @toc attribute to **no** for each of the child elements.

For example:

```
<topicref chunk="to-content" href="dia1400172931688.dita">
  <topicref toc="no" href="dia1400188729316.dita" ixia_locid="6"/>
  <topicref toc="no" href="rec1384889855768.xml" ixia_locid="7"/>
  <topicref toc="no" href="rec1384889856423.xml" ixia_locid="8"/>
</topicref>
```

If you do not set the @toc attribute, the output includes links to the child topics, but the text for all the links is the title of the parent topic. For example:

### Foobar Creation

Foobar Creation  
Foobar Creation  
Foobar Creation

The infrastructure team is looking into a solution to the problem, but until it is solved you need to hide the links to the child topics.

# 13

## *Topics*

[Creating Topics in the Ixiasoft CCMS](#) 119

[Generating Topics from Stubs](#) 122

[Previewing Content](#) 123

### **Creating Topics in the Ixiasoft CCMS**

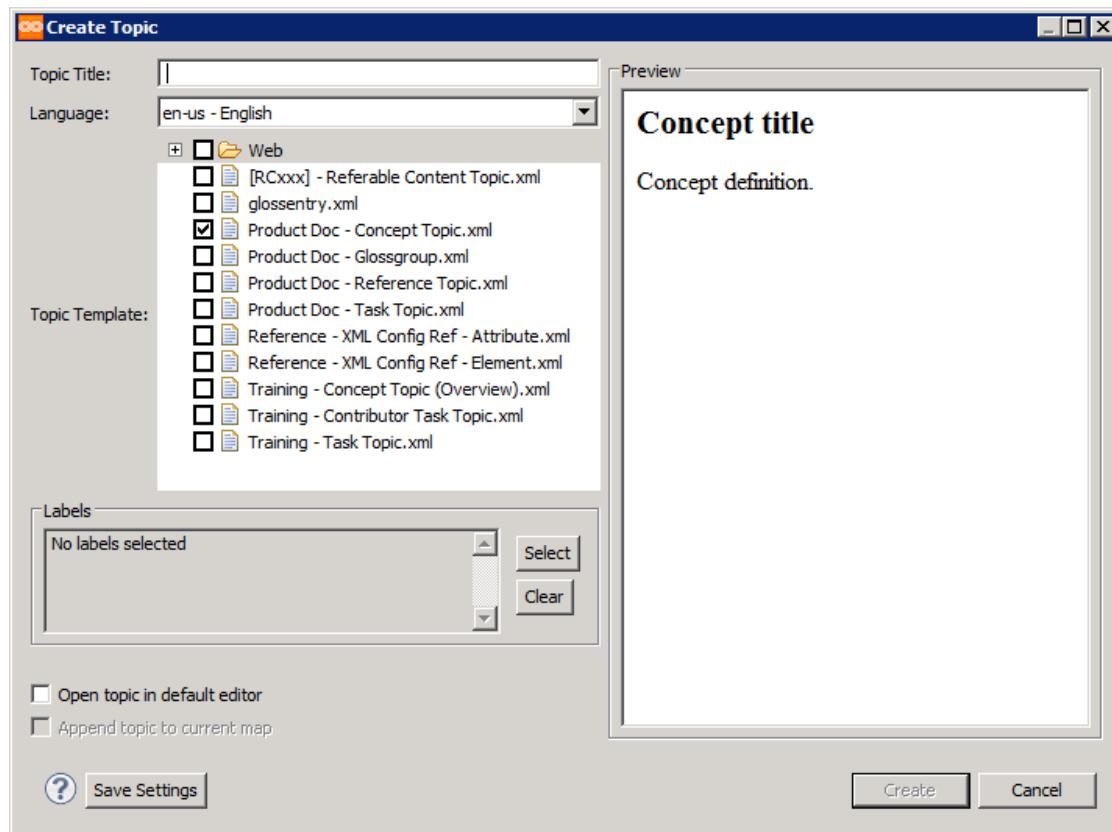
You can create topics by using the **IXIASOFT CCMS** menu item or by clicking the Create New Topic icon.

#### **Procedure**

1. Select **IXIASOFT CCMS > Create Topic** or click on the Create New Topic icon:



The Create Topic dialog box appears. The Preview pane shows the content included in the template.



2. Type the title for your topic.

Wind River standards require the following for topic titles:

#### Concepts

Use a noun phrase, for example, *The Topic Cloud* or *About Files in the Ixiasoft CCMS*

#### Tasks

Use a gerund, for example, *Adding Topics to a Map*.

#### References

Use a noun phrase, including "Reference" if the content is not clear without it. For example, *Parameters for make* or *Security Command Reference*.

#### Referable-content topics

Use the following format:

**[RC:*elementType*] *description***

Where *elementType* is the type of DITA elements contained in the referable-content topic, and *description* describes the purpose of the content. For example, if your referable-content topic contains steps for how to configure VxWorks, your title would be **[RC:steps] Configuring VxWorks**.

3. Select the type of topic.

---

Topic Type	Description
<b>[RCxxx] - Referable Content Topic.xml</b>	Used to store and organize reusable content.
<b>glossentry.xml</b>	Not currently used by Technical Publications.
<b>Product Doc - Concept Topic.xml</b>	Use the concept topic type to create support or overview information that users need to complete tasks successfully. They respond to the user's question, "What is this about?"
<b>Product Doc - Glossgroup.xml</b>	Use a glossary group topic type to create a glossary for your document.
<b>Product Doc - Man Page Reference Topic.xml</b>	Use a man page reference topic to provide reference material that is essential for the successful performance of technical tasks in a "man page" format (for example, a complete API reference).
<b>Product Doc - Reference Topic.xml</b>	Use a general reference topic to provide reference material that is essential for the successful performance of technical tasks in a simple format (typically presented as a table of items and descriptions or as a definition list).
<b>Product Doc - Task Topic.xml</b>	Use a task topic type to create a step-by-step procedure for completing a task. A task helps the users to answer the question "How do I?"
<b>Reference - XML Config Ref - Attribute.xml</b>	Use to create a topic containing XML configuration attribute definitions. The attribute definitions are included as a part of an XML configuration reference document.
<b>Reference - XML Config Ref - Element.xml</b>	Use to create a topic containing XML configuration element definitions. The element definitions are included as a part of an XML configuration reference document.
<b>Training - Concept Topic (Overview).xml</b>	Use the training concept topic type only for a training lab, to provide standard overview information that the student must know and understand to complete the training lab.
<b>Training - Contributor Task Topic.xml</b>	Not well-defined or used at present.
<b>Training - Task Topic.xml</b>	Use the training task topic type only for training labs, to describe steps the student must perform to complete a principal task in the lab.

---

4. Select any other desired options.

Option	Description
<b>Open topic in default editor</b>	This option locks the topic and opens it in the default editor after creating it.
<b>Append topic to current map</b>	This option is only available when you have a locked map selected in the DITA Map view. The option locks the topic and appends it to the bottom of the selected map.

5. Click **Create**.

## Generating Topics from Stubs

If you added a topic stub to your map as a placeholder during documentation planning, you must generate a topic from that stub before you can start writing.

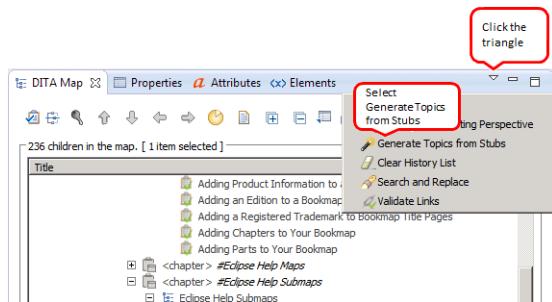
For information about creating topic stubs, see [Creating Topic Stubs](#) on page 110.

### Procedure

1. Use the **Search** view to locate the map.
2. Right-click the map in the **Search Results** view and select **Lock**.

**NOTE:** If the map is already locked (appears in bold blue text in the **Search Results** view, skip this step.)
3. Double-click the map in the **Search Results** view to open it in the DITA Map view.

**NOTE:** You cannot generate stubs by selecting the map in a parent map (such as an Eclipse Help map, an Eclipse Help Submap, or a bookmap). You must have the map open in the DITA Map view.
4. Click on the triangular menu icon at the upper right of the view and select **Generate Topics from Stubs** from the drop-down list.



5. In the Generate Topics from Stubs dialog box, select the desired topic stubs and click **OK**.

## Previewing Content

The Preview view allows you to see a WYSIWYG version of the topic you are editing or the contents of an image you have listed in various views.

### Procedure

1. If necessary, open the Preview view.

Select **Window > Show View > Preview**.

The Preview view opens by default in the bottom pane where the Search Results view appears.

2. Place focus on the item you want to preview.

- For a topic, make the topic active in the editor.
- For an image, highlight the image name in the view where it is listed.

3. Refresh the Preview view using the icon in the upper-right of the view.

Any time you change the item that has focus, you must refresh the Preview view to see the current content.

If you change the content, you must save before the new content appears in the Preview view.



# *Concept Topics*

[DITA Concept Structure](#) 125

[Workflow Topics](#) 126

## **DITA Concept Structure**

The concept topic allows you the flexibility to use almost any DITA element to create support or overview information that users need to complete tasks successfully.

The only elements not allowed in a concept topic are those elements specifically designed for task and reference topics; for example, <steps> and <properties>.

Concept topics contain the following mandatory elements:

DITA Element	Description
<title>	<p>Each concept must begin with a title to identify the subject.</p> <p>Use a noun phrase for your title; for example:</p> <ul style="list-style-type: none"><li>• The Topic Cloud</li><li>• About Files in the Ixiasoft CCMS</li></ul>
<shortdesc>	<p>A one or two sentence summary of what the concept is about and why the user should care. It functions as the introductory sentence to the concept, as well as the summary for links to the topic.</p> <p>For more information, see:</p> <ul style="list-style-type: none"><li>• <a href="#">Guidelines for Writing Short Descriptions</a> on page 153</li></ul>

DITA Element	Description
	<ul style="list-style-type: none"><li>• <a href="#">Examples of Short Descriptions</a> on page 155</li></ul>

## Workflow Topics

A workflow topic is a concept topic that informs the user of what tasks are involved in a process and in what order they must be completed. Actual steps are not included.

Use the `<ol>` element to create an ordered list of the tasks in the workflow. Include references to the task topics in the workflow.

You can add additional information to each list item in the ordered list; however, do not provide specific instructions. The actual steps should be included in a separate task topic.

### Example

 **NOTE:** This example just shows the body of a typical workflow topic. In a real workflow topic, you must also add a short description that acts as the first paragraph of your topic. Do not use a sentence that introduces the ordered list as your short description.

In a real workflow topic, you would also provide links to the appropriate topics. The links were not included here, as they would be out of scope for this document.

The workflow for creating a platform project and building your own Wind River Linux distribution is as follows:

1. Set up the platform project development environment.

This includes:

- creating a platform project directory
- defining your project configuration
- sourcing the development environment

2. Customize the platform project.

3. Deploy the platform project.

# 15

## *Task Topics*

<a href="#">Guidelines for Creating Tasks</a>	<a href="#">127</a>
<a href="#">DITA Task Structure</a>	<a href="#">129</a>
<a href="#">Step Structure</a>	<a href="#">131</a>
<a href="#">Prerequisites</a>	<a href="#">132</a>
<a href="#">Troubleshooting Information</a>	<a href="#">133</a>
<a href="#">Related Links in Task Topics</a>	<a href="#">134</a>
<a href="#">Formatting Steps that Involve Commands</a>	<a href="#">134</a>
<a href="#">Step Results</a>	<a href="#">135</a>

### **Guidelines for Creating Tasks**

Following the Wind River standard guidelines will help ensure your task topics are more effective and customer focused.

#### **Task Scope**

The task is a simple structure only containing actions the user should follow to reach a result. If more explanation is needed to complete the task, create a concept or reference topic.

Your task should contain neither background information about the product or features of the product nor should it contain specific information about the product or the feature, such as specifications or parameters. For example, an extensive list of all the parameters for a command should be included in a reference topic, rather than inline in the task.

You should describe only one task per topic. If you need more than one set of steps in your topic, you most likely have two separate topics.

## Number of Steps

As a general guideline, tasks should be 12 steps or less, including substeps, while still enabling the user to achieve a meaningful result. However, product constraints may require longer procedures. For example, a procedure may require more than 12 steps because the user interface times out if the user stays on a single screen in a series of screens but completion of all screens is required to complete the task.

## Multiple Ways to Complete a Task

In general, if there are multiple ways to accomplish a task (for example, using a GUI or the command-line or the same procedure for different processor architectures), create separate task topics. Providing instructions for multiple ways of completing the task in the same topic is distracting and interrupts the task flow.

## Prerequisites

Prerequisites should be placed in a `<prereq>` element before the steps, not embedded within the steps.

For more information, see [Prerequisites](#) on page 132.

## Troubleshooting Information

Troubleshooting information related to a task should not be part of the task. It should be part of a separate troubleshooting topic.

For more information, see [Troubleshooting Information](#) on page 133.

## Related Links

Links to information that is helpful or required for the task should be placed outside of any steps to avoid distraction. Only provide links that are directly relevant to the task.

For more information, see [Related Links in Task Topics](#) on page 134.

## Cross References to Other Documents

To keep the reader focused on the task, you should provide the appropriate level of granularity when directing users to other documents.

This is especially true when the scope of the document the reader is being guided to is broader than the information they are looking for. Be as fine-grained as possible in describing what material they should consider in the other document. For example:

- For information on configuring product licenses, see the Wind River product installation and licensing guides at the following website:  
[windriver.com/licensing/documents](http://windriver.com/licensing/documents)
- For information on activating your products, go to the Licensing Portal at the following website:  
[windriver.com/licensing](http://windriver.com/licensing)
- For information on configuring a VIP, see the *VxWorks Configuration and Build Guide: Configuring a VxWorks Image Project*.

### Formatting of Steps That Involve Commands

If your documentation contains tasks that require users to type commands or enter source code, you should use a minimalist approach and avoid redundancy. It is typically obvious to users that they must type a command.

For more information, see [Formatting Steps that Involve Commands](#) on page 134.

### Step Results

Results should only be used for key milestones during the task. Trivial or otherwise non-important results should be avoided.

For more information, see [Step Results](#) on page 135.

### Workflow vs. Task

A workflow topic is a concept topic that informs the user of what tasks are involved in a process and in what order they must be completed. Actual steps are not included. Use the `<ol>` element to create an ordered list of the tasks in the workflow. Include references to the task topics in the workflow.

A task topic includes actual steps for tasks the customer must perform.

## DITA Task Structure

The task topic type provides a simple structure that allows you to document a step-by-step procedure for completing a task.

Task topics contain the following mandatory and optional elements:

DITA Element	Mandatory or Optional	Description
<code>&lt;title&gt;</code>	Mandatory	<p>Each task must begin with a title to identify the task subject.</p> <p>Use the gerund form of the verb for your title; for example:</p> <ul style="list-style-type: none"><li>• Installing the Product</li><li>• Configuring Foo</li></ul> <p>Make sure your title reflects the actual user task or goal. For example, rather than creating a topic such as "Using the Project Configuration Editor", create a topic that focuses on the actual user task, such as, "Configuring the Project to Include Debugger Support".</p>
<code>&lt;shortdesc&gt;</code>	Mandatory	<p>A one or two sentence summary of what the task is about and why or when the user should perform the task. It functions as the introductory sentence to the task, as well as the summary for links to the topic.</p>

DITA Element	Mandatory or Optional	Description
		<p>For more information, see:</p> <ul style="list-style-type: none"> <li>• <a href="#">Guidelines for Writing Short Descriptions</a> on page 153</li> <li>• <a href="#">Examples of Short Descriptions</a> on page 155</li> </ul>
<context>	Optional, but recommended	<p>Important information the user needs to know to perform the task. It should expand on the short description, and the first sentence should make sense when read after the short description.</p> <p>Ensure that the information is directly related to the task and avoid introducing conceptual information.</p> <p>If the task is short or simple, and there is enough text in the short description, you can omit the &lt;context&gt; element.</p>
<prereq>	Optional	<p>Tasks the user must have performed, the required state of the system, or other knowledge the user needs before performing this task. It is appropriate to provide links to the relevant tasks or web links for knowledge that is not specific to the product.</p>
<steps> or <steps-unordered>	Mandatory	<p>One or more instructions to perform the task.</p> <p>As a general guideline, tasks should be 12 steps or less, including substeps, while still enabling the user to achieve a meaningful result.</p> <p>For details about step structure, see <a href="#">Step Structure</a> on page 131.</p>
<result>	Optional	<p>The overall outcome of the task. Typically, it is only necessary if the overall outcome of the procedure encompasses more than the result of the final step in the procedure.</p>
<postreq>	Optional	<p>Optional steps not directly related to the task or a recommended but not mandatory task that is not part of the workflow to which the task belongs. It can also specify the next required or recommended workflow.</p> <p>Do not specify the next step in a workflow.</p>
<related-links>	Optional	<p>Additional information the user may want to know, but that is not essential to perform a step and is not a prerequisite.</p>

## Step Structure

Steps are a single instruction in an overall procedure.

Steps should be concise and contain all information the user needs to successfully complete the step.

Steps contain the following mandatory and optional elements.

DITA Element	Mandatory or Optional	Description
<cmd>	Mandatory	<p>The step instruction (or command) should be a single sentence describing the action the user needs to take. Write the instruction in the imperative voice; for example:</p> <ul style="list-style-type: none"> <li>• Connect your target hardware to your host system.</li> <li>• Query the patching status of all hosts in the cluster.</li> <li>• Right-click the VIP and select <b>Rebuild Project</b>.</li> </ul> <p>If your documentation contains tasks that require users to type commands or enter source code, you should use a minimalist approach and avoid redundancy. For instructions on how to format your step instruction in this case, see <a href="#">Formatting Steps that Involve Commands</a> on page 134.</p>
<info>	Optional	Additional information the user might need to perform the task or side effects of the steps that are not readily apparent in the result.
<stepxmp>	Optional	Specific examples, such as code snippets or command-line input.
<substeps>	Optional	<p>Steps may be divided into substeps. Each substep has the same basic structure as a step, except it cannot contain another level of substeps, nor can it contain the &lt;choices&gt; or &lt;choicetable&gt; elements.</p>
<stepresult>	Optional	<p>The outcome of performing the step, such as console output, the new state of the system, or output the user sees on the user interface.</p> <p>Results should only be used for key milestones during the task. Trivial or otherwise non-important results should be avoided.</p> <p>For more information, see <a href="#">Step Results</a> on page 135.</p>

DITA Element	Mandatory or Optional	Description
<choices>	Optional, should be used sparingly	A bulleted list of simple one-part items the user can select to complete the task; for example: 1. Select your operating system: <ul style="list-style-type: none"><li>• Windows</li><li>• Linux</li><li>• Unix</li></ul> If you must also provide a description of the items, use the <choicetable> element instead.
<choicetable>	Optional, should be used sparingly	A table that includes a two-part list of items (typically an option and its description) the user can select to complete the task. However, if there are multiple ways to accomplish a task (for example, using a GUI or the command-line or the same procedure for different processor architectures), create separate task topics for each method. Providing instructions for multiple ways of completing the task in the same topic is distracting and interrupts the task flow.

## Prerequisites

Prerequisites should be placed in a <prereq> element before the steps, not embedded within the steps.

Do not include procedures for completing the prerequisite in the task topic. Instead, create a separate task topic for the instructions for completing the prerequisite and provide a link.

If there is more than one prerequisite, use an unordered list to present the information.

### Examples:



**NOTE:** The examples below are real-world examples; however, since links to the original topics are out-of-scope for the DITA User's Guide, the links now go to other topics in this document rather than to the original topics.

### Prerequisites

The agent must be running on the device and the device must be connected to the server.

### Prerequisites

To assign a permission group to a user, the permission group must already exist. For information about creating permission groups, see *Wind River Helix Device Cloud Platform User's Guide: Creating Permissions Sets*.

## Prerequisites

Before you create the boot media and boot your device, you need the following:

- the images generated from your platform project (see [Building Your Wind River Linux Platform Project for ARM Target Devices](#))
- your tenant user name and password (received when you registered for Helix Device Cloud or provided by your tenant administrator)
- ports 8883, 465, and 443 open on the firewall of the network to which the device connects

## Troubleshooting Information

Troubleshooting information related to a task should not be part of the task. It should be part of a separate troubleshooting topic.

Troubleshooting within the context of a task can be distracting and confusing. The task should assume that all steps are successful.

A separate troubleshooting topic is an appropriate place for users to find solutions for failed steps. You can refer the user to the troubleshooting topic as shown in the post-requisites section of the example.

It is good practice to provide guidance about how to verify steps have been successful. In these cases, suggesting verification methods are appropriate when they are commonly applied and are not distracting from the task at hand. Step 1 of the example illustrates this method.

---

**NOTE:** The example below is a real-world example; however, since links to the original topics are out-of-scope for the DITA User's Guide, the links now go to other topics in this document rather than to the original topics.

---

### Example

#### Procedure

1. In the confirmation window that appears, click **OK**.

The device is enabled when the **agent\_enable** action is no longer available in the **device\_manager 1.0** menu. You may need to refresh the page a few times until the device status changes.

2. On the Systems Information page of the device, select **Actions > software\_manager 1.0 > Deploy package**.

#### Postrequisites

If your device does not appear on the server, see [Diagnosing Initial Connectivity Failures](#) for troubleshooting information.

To enable you to do software updates, you must now preserve your build environment and create a project manifest as the baseline for creating the update files. For more information about software updates and how to create the manifest, see the following:

- [About Software Updates](#)
- [Creating a Project Manifest](#)

## Related Links in Task Topics

Links to information that is helpful or required for the task should be placed outside of any steps to avoid distraction.

Only provide links that are directly relevant to the task. You can place them in the `<context>`, `<prereq>`, or `<related-links>` sections of the topic. Examples of both are shown below.



**NOTE:** The example below is a real-world example; however, since links to the IDP XT topics are out-of-scope for the DITA User's Guide, the links now go to other topics in this document rather than to the IDP XT topics.

### Example

There are additional configuration options available in IDP XT and Wind River Linux. The IDP XT documentation describes the available security features and how to enable and disable them. For more information about the available IDP XT and Wind River Linux documents, see [Where to Find Information](#).

### Procedure

1. Prepare a directory to house your project.
2. Configure your project by editing the configuration file.
3. Build the project by performing the appropriate command.

### Related Links

[Logging in to the Device Remotely](#) on page 2

SVN is a file management system, while the Ixiasoft CCMS is a topic and map management system.

[Creating a Project Manifest](#) on page 1

The Ixiasoft component content management system (CCMS) is used to store and publish DITA topics, maps, images, and resources.

## Formatting Steps that Involve Commands

If your documentation contains tasks that require users to type commands or enter source code, you should use a minimalist approach and avoid redundancy. It is typically obvious to users that they must type a command.

To avoid redundancy, do not write your commands using any of the following formats:

- Execute the following command to query the patch status of all nodes.
- Query the patch status of all nodes using the following command.
- To show the patch status of all nodes, execute the following command.
- Type the following at the command prompt.
- Do the following.

Use an imperative sentence in the `<cmd>` element clearly stating what executing the command accomplishes, and include the actual command in a `<codeblock>` element within a `<stepxmp>` element, as shown in the example.

### Example

### Procedure

1. Apply all available patches.  

```
$ sudo sw-patch apply --all
```
2. Query the patch status of all nodes.

```
$ sudo wrs-patch query-hosts
  Hostname      IP Address      Patch Current      Reboot Required
  ======  ======  ======  ======
controller-0  192.168.204.3      No            Yes
controller-1  192.168.204.4      No            Yes
```

## Step Results

The step results element provides the outcome of performing a step, such as console output, the new state of the system, or output the user sees on the user interface.

Results should only be used for key milestones during the task. Trivial or otherwise non-important results should be avoided. A common example of this is when a task involves multiple steps in a GUI where actions are performed in several windows. The next window opening is not usually an interesting result, and it should not be captured as a step result.

However, context can be important. If you think users might be confused about where they are in the GUI, add that information as part of the step, as shown in Step 3 of the example.

### Example

To produce Y, you must do X.

### Procedure

1. Select **One** > **Two** > **Three**.
2. In the **Three** window, click **Button**.
3. To X, select **Four** > **Five** > **Six**.

Y appears.



# 16

## *Reference Topics*

[Guidelines for Creating Reference Topics](#) 137

[DITA Reference Structure](#) 139

### **Guidelines for Creating Reference Topics**

Following the Wind River standard guidelines for reference topics helps ensure that your reference information is presented in a consistent and clear manner.

Reference topics present some of the most helpful and frequently used information in your product topic set. Experienced users may focus almost exclusively on reference information during their development. For this reason, be sure to present reference information consistently and always use a reference topic for reference information. (The Ixiasoft CCMS includes two reference topic templates; one for presenting general reference information and one for "man-page" type information.)

A reference topic can present information for a single item, or multiple items. (That is, even a one item name and description is a valid reference topic.) If you have a large number of reference topics related to a common topic, such as a set of routines for an API, consider making a standalone reference document for your product. Otherwise, include your reference topics after any related conceptual and task topics.



**NOTE:** In DITA, topic sets typically begin with concept topics, followed by task topics, and end with any reference topics.

---

#### **Examples of Reference Information**

Some examples of typical reference information include:

- error codes
- routines and libraries
- UI elements

configuration options

This is not an exhaustive list. Any information that is essential for the successful performance of technical tasks can be considered reference information. However, the information must be essential and should not include lengthy explanations of concepts.

When reviewing your content, always check concept and task topics for hidden reference information. While it may be ok to include a brief reference for a single option in a task or concept topic, exhaustive information or lengthy lists of items such as routines or configuration options should be included in a separate reference topic.

### Reference Topic Formats

General reference information is typically presented as a list or table with a term describing the reference item and a short description for the item. For example:

Configuration Option	Description
<code>setFoo</code>	Sets a value for the Foo element.
<code>setBar</code>	Sets a value for the Bar element. The default value is 1.
<code>setFooBar</code>	Sets a value for FooBar. Currently unused.

When creating your reference topic, use the **Product Doc - Reference Topic.xml** template for this type of presentation.

You can also use a reference topic to display information in a "man page" format. This works best for items such as routines or configuration options where the description is more involved and a standard set of information is presented for each item. For example:

### Element: Manager

This contains the characteristics of a custom manager.

#### Document Type

ModuleOS

#### XPath

/ModuleOS/Managers/Manager

#### Description

This contains the characteristics of a custom manager.

Normally, all managers are blocked on `receive()` waiting for requests from partitions. While the scheduler can prevent a partition from running outside its window, it cannot prevent an unrestricted manager from servicing a partition's request outside the partition's window. To mitigate this, VxWorks 653 implements worst-case execution times for managers.

When creating a reference topic for man page information, use the **Product Doc - Man Page Reference Topic.xml** template.

## DITA Reference Structure

A DITA reference topic provides a simple structure for you to present essential reference information. It typically includes a simple table but can be adapted to support man-page-style information or definition lists.

Reference topics allow only a small subset of DITA elements at the top level (that is, directly under `<refbody>`) and, in its most basic form, a reference topic is typically nothing more than a table of items with associated descriptions. However, `<refbody>` also allows the `<section>` and `<example>` elements. These elements open up most other DITA elements including paragraphs, lists, and definition lists. This allows you the flexibility to present your reference information as needed.

---

**NOTE:** Wind River recommends that you use the templates provided for general references and man pages and follow the guidance provided in the template comments. These templates are designed to handle most types of reference information provided in our documentation.

---

### General Reference Topics

A general reference topic typically consists of a table element that presents a basic list of items and descriptions. The table can have two or more columns and as many rows as needed. If your items need to be grouped, multiple tables can be used.

Use the **Product Doc - Reference Topic.xml** template for this type of reference.

---

**NOTE:** If you need to include any introductory text for your table, you can insert a `<section>` element and use it to contain one or more paragraphs (`<p>`) and any tables. However, if the purpose of the table and its elements is clear from the topic or table title and any column headings, introductory text is not required and the section element is not needed.

---

Alternatively, you can insert a section element and use it to contain a definition list (`<dl>`). This alternative can be useful when you have lengthy routine or option names that are difficult to fit in a reasonable table column width.

### Man Page Reference Topics

When creating a man-page-style reference topic, the topic title typically reflects the item you are describing (such as a routine name). The `<refbody>` element then contains one or more `<section>` elements with titles that describe the type of information that you are presenting for the title element. For example, a man page for a routine might include a `<section>` element with a `<title>` of *Synopsis* and another with a `<title>` of *Description*.

When creating a reference topic for man page information, use the **Product Doc - Man Page Reference Topic.xml** template.

## Other Options

Because you can include a **<section>** element in a **<refbody>** element, you can exercise a great deal of flexibility when formatting your reference topic. Wind River recommends that you use the general reference table or definition list approach, or the man page approach as much as possible. These two options cover most types of reference information presented in our documentation. However, you can use an alternative when needed. For example, if you have a single item to describe, you might want to use a standard paragraph (although a single entry table is acceptable, and preferred in many cases). You can also have a reference that consists of one or more examples with related code snippets. In this case, use one or more **<example>** elements to present your information.

# Referable-Content Topics

[Guidelines for Creating Referable-Content Topics](#) 141

[Example Referable-Content Topics](#) 142

## Guidelines for Creating Referable-Content Topics

A referable-content topic is a specialized topic that allows you to store and organize reusable content. Following the Wind River guidelines for creating referable-content topics helps you create content that is easy to maintain and reuse.

### Guidelines

- Unlike the other topics types, a referable-content topic does not have a defined structure. You can add any body or inline element directly in the top-level `<rcbody>` element.
- You should limit the amount of content in a referable-content topic to keep the file dependencies manageable. Best practices state that you should create a referable-content topic for each piece of content you want to reuse. However, you can create a referable-content topic that contains a group of the same type of element; for example, a group of steps or list items.
- Reuse content at the topic or paragraph level. Do not reuse content at the phrase or word level.
- Only conref content stored within referable-content topics. If you create conrefs between regular topics, you risk creating complex and unintended file dependencies.
- If it takes longer to create a conref than it does to type the text, type the text.
- Avoid cross-references in reused content. When your reused content contains cross-references, it limits where you can reuse the content, and you create file dependencies.
- Use the following format for the title of your referable-content topic:

*[RC:`elementType`] description*

Where *elementType* is the type of DITA elements contained in the referable-content topic, and *description* describes the purpose of the content. For example, if your referable-content topic contains steps for how to configure VxWorks, your title would be similar to the following:

### [RC:steps] Configuring VxWorks

## Example Referable-Content Topics

Reviewing examples of referable-content topics may make it easier to create your reusable content.

Unlike the other topics types, a referable-content topic does not have a defined structure. You can add any body or inline element directly in the top-level `<rcbody>` element. However, you should reuse content at the topic or paragraph level. Do not reuse content at the phrase or word level.

The following examples, show some of the more common ways to add content to a referable-content topic.

### Body Elements

You can create a referable-content topic that contains several `<p>` or other body elements, and pull in only the content you need. For example, if you have two topics that both contain similar overview sections, but you want a more basic overview in one topic and an expanded overview in the other topic. In this case, you would create a referable-content topic that contains only the shared content.

```
<referable-content>
  <title>[RC:p] VxSim Overview</title>
  <rcbody>
    <p>The VxWorks simulator is a simulated hardware target for use as a prototyping and test-bed environment for VxWorks.</p>

    <p>The VxWorks simulator allows the development and test of VxWorks applications on a host system, reducing the need for target hardware during the initial phases of development. The VxWorks simulator supports the creation of a simulated target network for developing and testing complex networking applications.</p>
  </rcbody>
</referable-content>
```

If you want to reuse a note in several topics, you would create a referable-content topic similar to the following:

```
<referable-content>
  <title>[RC:note] WR Linux 7 Caveats</title>
  <rcbody>
    <note>
      <p>Support for Wind River Linux 7 guests is not yet integrated with this VxWorks 653 feature.</p>
    </note>
  </rcbody>
</referable-content>
```

## Step Elements

If you want to reuse all the steps in a task, you must create a referable-content topic containing the **<steps>** element.

However, you are more likely to reuse individual steps in the task. For example, if you have two topics that share many of the same steps and you want to add additional steps in one topic but not the other. You cannot reuse the **<steps>** element, because you will not be able to add additional steps. You must create a referable-content topic that just contains the individual steps.

```
<referable-content>
    <title>[RC:step] Assigning a New Host Node</title>
    <rcbody>
        <step>
            <cmd>Add the host to the system inventory.</cmd>
            <info>
                <note>
                    <p>The host must be added to the system inventory before it is powered
on.</p>
                </note>
            </info>
        </step>
        <step>
            <cmd>Power on the host.</cmd>
        </step>
        <step>
            <cmd>Assign a personality.</cmd>
        </step>
        <step>
            <cmd>Obtain the host ID.</cmd>
        </step>
    </rcbody>
</referable-content>
```

You can also reuse select parts of a step. For example, if you have several topics that include a step where the **<cmd>** and the **<stepresult>** elements are the same, but the **<stepxmp>** element is different. You cannot reuse the whole step, because you must add different **<stepxmp>** elements between the **<cmd>** and the **<stepresult>** elements. You must create a referable-content topic that contains just the **<cmd>** and **<stepresult>** elements.

```
<referable-content>
    <title>[RC:cmd and info] Building the Project</title>
    <rcbody>
        <cmd>Build the project.</cmd>
        <stepresult>
            <p>After the build completes, the kernel image resides in the default folder.
For example:</p>
            <p><filepath><varname>installDir</varname>/workbench-4/workspace/VxSimVIP/
default/vxWorks</filepath></p>
        </stepresult>
    </rcbody>
</referable-content>
```

## Lists

You can reuse entire lists, or you can reuse individual list items. For example, if you have two topics that include similar unordered lists and you want add additional list items to one topic. If

you reuse the `<ul>` element, you are unable to add additional list items. You must create a referable-content topic that contains just the `<li>` elements of the list.

```
<referable-content>
  <title>[RC:li] Supported BSPs</title>
  <rcbody>
    <li><p>Freescale i.MX6</p></li>
    <li><p>Freescale P1/P2</p></li>
    <li><p>Intel Generic</p></li>
    <li><p>VxSim for Linux</p></li>
  </rcbody>
</referable-content>
```

## Entire Topics

If you need to include a topic several times in the same bookmap, you must create a referable-content topic that contains a topic. For example, if you want to include a task topic, your referable-content topic would look similar to the following:

```
<referable-content>
  <title>[RC:step] Assigning a New Host Node</title>
  <rcbody>
    <task-wr>
      <title/>
      <shortdesc/>
      <prolog>
        <author/>
      </prolog>
      <taskbody>
        <context/>
        <steps>
          <step>
            <cmd/>
          </step>
        </steps>
      </taskbody>
    </task-wr>
  </rcbody>
</referable-content>
```

For more information, see [Including a Topic Multiple Times in the Same Book](#) on page 236.

# 18

## *Resource Files*

<a href="#">Creating Resource Files</a>	145
<a href="#">Editing Resource Files</a>	146
<a href="#">Deleting Resource Files</a>	147
<a href="#">Importing Resource Files that Were Exported from the Ixiasoft CCMS</a>	147

### **Creating Resource Files**

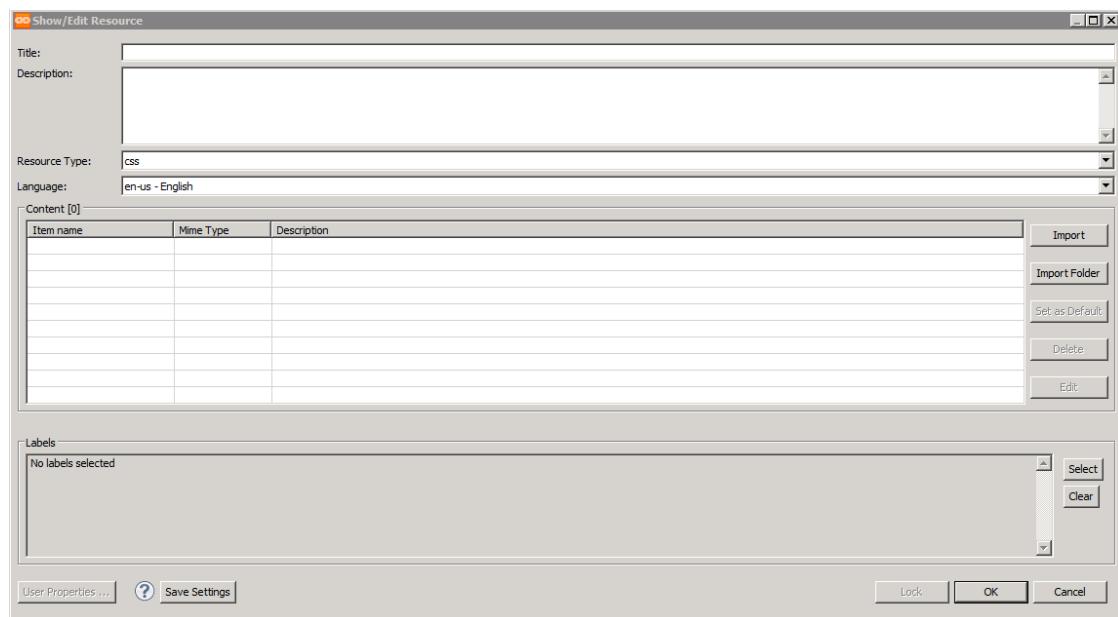
You can create a resource file to store non-DITA files in the Ixiasoft CCMS.

A typical use case for creating a resource file is to store the `dctrl_output.txt` file for the [Diab Compiler Configuration Reference](#).

#### **Procedure**

1. Select **IXIASOFT CCMS > Create Resource**.

The Show/Edit Resource dialog box appears.



2. Type a title for the resource file in the **Title** text box.
3. Type a description of the resource file in the **Description** text box.
4. Select the resource type from the **Resource Type** drop-down list.
5. Select the import file or folder by clicking either **Import** or **Import Folder**, as appropriate.
6. Click **Open**.
7. In the Show/Edit Resource dialog box, click **OK**.

The file or folder is stored in the Ixiasoft CCMS as a resource.

## Editing Resource Files

If you want to edit a resource file, you must edit it in the Edit/Show Resource dialog box.

### Procedure

1. Right-click the desired resource file and select **Edit**.

The resource file is locked and the Edit/Show Resource dialog box appears. This step generates some traffic and messages.

2. Edit the resource file as necessary and click **OK**.
3. Right-click the resource file and select **Release** to release the changes to the Ixiasoft CCMS.

## **Deleting Resource Files**

If a resource file is no longer needed, you can delete it from the Ixiasoft CCMS.

### **Prerequisites**

The file must not be locked.

### **Procedure**

1. Right-click the desired resource file and select **View Dependencies**.
2. Remove all dependencies on the file.
3. Right-click the resource file and select **Delete**.

If the **Delete** option is grayed out, the file may still be referenced by a map.

4. When prompted if you want to delete the file, click **Yes**.

## **Importing Resource Files that Were Exported from the Ixiasoft CCMS**

To import resource files that were exported from the Ixiasoft CCMS, you must import the map that was exported.

### **Procedure**

1. Import the map that was exported.

For more information, see [Importing Files](#) on page 15.

The Ixiasoft CCMS will claim that there were fewer files imported than were processed. The resource file will be among the missing files. However, the resource file is actually imported.

The resource has no title or description.

2. Change the name of the resource file in any element or attribute that refers to it.

When the resource file is imported, it has a new Ixiasoft CCMS ID and, therefore, a new file name. You must update the file name in any references.



# Glossaries

[Creating a Glossary Topic](#) 149

[Adding a Glossary to a Bookmap](#) 150

[Referencing a Glossary Entry](#) 150

## Creating a Glossary Topic

You can create a glossary for your document by using the **Glossgroup - Wind River** topic type and defining each term in a **<glossentry>** element.

### Procedure

1. Create a **Glossgroup - Wind River** topic.
2. Add a **<glossentry>** element for each glossary term.

When you add the **<glossentry>** element, empty **<glossterm>** and **<glossdef>** elements are added inside the **<glossentry>** element. A unique ID attribute is also added.

```
<glossentry id="glossentry_csm_x15_5bb">
  <glossterm/>
  <glossdef/>
</glossentry>
```

3. Add the appropriate text to the **<glossterm>** and **<glossdef>** elements.

```
<glossentry id="glossentry_csm_x15_5bb">
  <glossterm>Data Definition Language</glossterm>
  <glossdef><p>A data definition language or data description language (DDL) is a
syntax similar to a
computer programming language for defining data structures, especially
database
schemas.</p></glossdef>
</glossentry>
```

## Adding a Glossary to a Bookmap

To add a glossary to your bookmap, you must add it as an appendix.

### Procedure

1. Add an `<appendices>` element after the chapters in your bookmap.
2. Add an `<appendix>` element to the `<appendices>` element.  
If your document contains other appendices, place the `<appendix>` element for the glossary after all other appendices.
3. Add a `<topicmeta>` element to the `<appendix>` element.
4. Add a `<navtitle>` element to the `<topicmeta>` element and use **Glossary** as the navigation title.
5. Add a `<topicref>` element after the closing tag for the `<topicmeta>` element.
6. Add an `@href` attribute pointing to your glossary topic.

```
<appendices>
  <appendix>
    <topicmeta>
      <navtitle>Glossary</navtitle>
    </topicmeta>
    <topicref href="dia1403047595564.dita"/>
  </appendix>
</appendices>
```

## Referencing a Glossary Entry

To create cross-references to the glossary terms in your glossary, you must use a slightly different format than the format for links to internal targets in other types of files.

### Procedure

1. Open the topic that will contain the cross-reference.
2. Place your cursor in the topic where you want to add a cross-reference to the glossary entry.
3. Right-click on the glossary topic in your bookmap and select **Oxygen Editor > Insert as XRef**.

An `<xref>` element is added to your topic with the file name of the glossary as the value of the `@href` attribute.

```
<xref href="dia1403114928243.dita"/>
```

4. Open the glossary topic in Text mode in the editor.
5. Copy the ID of the `<glossentry>` element of the term you want to reference.
6. In the topic that contains the `<xref>` element, add the symbol # followed by the ID of the `<glossentry>` element to the end of the `@href` attribute.

```
<xref href="dia1403114928243.dita#glossentry_pb2_ky1_vbb"/>
```

7. Add the appropriate text to the `<xref>` element.

If you do not add text, the name of the glossary topic is used.

```
<xref href="dia1403114928243.dita#glossentry_pb2_ky1_vbb">My Term</xref>
```



# 20

## *Short Descriptions*

[Guidelines for Writing Short Descriptions](#) 153

[Examples of Short Descriptions](#) 155

### **Guidelines for Writing Short Descriptions**

To be effective, a short description must work both as the first paragraph of a topic, and as the summary for links to the topic.

As a first paragraph, it must flow naturally. As the summary for a link, it must help the user decide whether to follow the link.

To ensure that your short descriptions work well in both contexts, observe the following guidelines:

- Keep short descriptions short. A limit of 35 words is recommended, using one or two sentences.
- Use complete sentences.

For example:

- "Author mode allows you to edit XML in a formatted editor."

Not:

- "Allows you to edit XML in a formatted editor."

- Use the short description to begin the topic, not to talk about the topic.

For example, for a topic about short descriptions:

- "To be effective, a short description must work both as the first paragraph of a topic, and as the summary for links to the topic."

Not:

- "Learn how to write an effective short description."

- "This topic explains how to create a good short description."
- Clarify the value of the topic. In the context of a linked topic, the user must understand why the linked information is important.

For example:

- "The oXygen editor has a rich set of keyboard functionality. The shortcut keys help provide an easier and usually quicker method of navigating and using oXygen."

Not:

- "The oXygen editor has a rich set of keyboard functionality."
- Summarize the content of the topic, so that the user understands what is covered. If you cannot summarize the content of the topic in one or two short sentences, consider refactoring your content.
- Do not simply repeat the topic title.

For example, for a topic describing the guidelines for writing short descriptions, do not write "There are guidelines to follow when writing a short description."

- Do not use constructions requiring completion.

For example, do not say "To write an effective short description, use the following guidelines." Such constructions do not work well in a link context.

- Do not place cross-references in the short description.
- Do not use an imperative sentence for your short description.

For example, do not say "Learn how to write an effective short description." In most cases, it just repeats the topic title, and it does not work well as the first paragraph of your topic.

In addition to these guidelines, you can ask yourself some questions for each information type:

Concept	Task	Reference
<ul style="list-style-type: none"><li>• What is the object, concept, or idea?</li><li>• Why should the user care about this object, concept, or idea?</li></ul>	<ul style="list-style-type: none"><li>• What does the task help users accomplish?</li><li>• What are the benefits of doing this task or why is the task important?</li><li>• When would users perform the task?</li><li>• What is involved in the task?</li><li>• Why do users need to perform this task?</li><li>• How does the task fit with other related tasks?</li></ul>	<ul style="list-style-type: none"><li>• What does this object do?</li><li>• How does the object work?</li><li>• What is the object used for or why is it helpful?</li></ul>

## Examples of Short Descriptions

You can learn to write more effective short descriptions by studying examples of incorrect and correct short descriptions.

### Task Topics

Topic	Short Description	Comments
Creating a Layer Project Using Workbench	<p>INCORRECT:</p> <p>You can create a layer project using Workbench.</p> <p>CORRECT:</p> <p>To create a layer with Workbench, you must create and configure a project with a build type of <b>layer</b>.</p>	<p>The incorrect example basically repeats the topic title. It does not provide any information about what is involved or why users would want to perform the task.</p> <p>The correct example provides an overview of what is involved in the task. It also provides a quick reminder for those who already know the procedure.</p>
Restricting Access to a Single License Type	<p>INCORRECT:</p> <p>Learn how to restrict access to a single license type.</p> <p>CORRECT:</p> <p>If your organization has purchased both named-user and floating license types, you can use the <b>INCLUDE</b> action keyword to restrict access for a group of users to one license type or another.</p>	<p>The incorrect example basically repeats the topic title. It is also an imperative statement, which does not work well as the first paragraph of the topic.</p> <p>The correct example tells readers why they need to perform the task and provides an overview of what is involved in the task.</p>
Manually Downloading Source Files	<p>INCORRECT:</p> <p>The following supported source packages are known not to be available from the product distribution and must be downloaded manually:</p> <p>CORRECT:</p> <p>Due to licensing restrictions, some required source packages are not included with this Wind River Linux product. They must be downloaded manually and incorporated into the platform project.</p>	<p>The incorrect example introduces a list. This does not work well in a link context.</p> <p>The correct example tells readers why they need to perform the task and provides an overview of what is involved in the task.</p>

Topic	Short Description	Comments
Starting a QEMU Target from the SDK Directory	<p>INCORRECT:</p> <p>This task describes how to start a QEMU target from the SDK directory.</p> <p>CORRECT:</p> <p>If you do not have access to the platform project directory, you can use the SDK directory on a Linux host to deploy an application project for testing.</p>	<p>The incorrect example is self-referential; that is, it describes what the topic is going to do, rather than describing the task.</p> <p>The correct example tells readers why they need to perform the task.</p>

### Concept Topics

Topic	Short Description	Comments
About Patching Packages	<p>INCORRECT:</p> <p>The Wind River Linux build system uses the Quilt patching model. Quilt lets you easily add a series of patches, where each patch may contain multiple files. Also see "How To Survive With Many Patches" or "Introduction to Quilt," <a href="http://www.suse.de/~agruen/quilt.pdf">http://www.suse.de/~agruen/quilt.pdf</a> by Andreas Grünbacher.</p> <p>CORRECT:</p> <p>The Wind River Linux build system uses the Quilt patching model. Quilt lets you easily add a series of patches, where each patch may contain multiple files.</p>	<p>The incorrect example includes a cross reference. By design, DITA does not allow the <code>&lt;xref&gt;</code> element in short descriptions. You do not want to send readers to another source before they read the current topic.</p>
Downloadable Kernel Module Projects	<p>INCORRECT:</p> <p>You can use DKM projects to manage and build modules that will exist in the kernel space. You can separately build the modules, run, and debug them on a target running VxWorks, loading, unloading, and reloading on the fly. Once your development work is complete, the modules can be statically linked into the kernel, or they can use a file system if one is present.</p>	<p>The incorrect example is too long and not to the point.</p> <p>The correct example is concise and gives the reader just enough information to determine whether or not they should click on the link or if they have reached the correct topic.</p>

Topic	Short Description	Comments
	<p>CORRECT:</p> <p>Downloadable kernel module (DKM) projects provide the development environment to manage and build kernel application modules that execute in kernel space.</p>	
RTP Applications	<p>INCORRECT:</p> <p>Overview of RTP applications.</p> <p>CORRECT:</p> <p>Real-time process (RTP) applications are user-mode applications similar to those used with other operating systems, such as UNIX and Linux.</p>	<p>The incorrect example is not a complete sentence, and simply restates the topic title.</p> <p>The correct example provides a brief description of the feature.</p>

#### Reference Topics

Topic	Short Description	Discussion
Compilation Modes and Options for C Language Dialects	<p>INCORRECT:</p> <p>This topic lists the compilation modes and options for C language dialects.</p> <p>CORRECT:</p> <p>C code can be compiled in conformance with several different standards: C89 , C99, ANSI, K&amp;R, and PCC. These choices may facilitate porting of existing C programs.</p>	<p>The incorrect example is self-referential; that is, it describes what the topic is going to do, rather than providing information about the modes or options.</p> <p>The correct example briefly describes the modes and why they are useful.</p>
Sample Templates for Titanium Server	<p>INCORRECT:</p> <p>Titanium Server provides sample templates.</p> <p>CORRECT:</p> <p>You can evaluate selected features of Heat using sample templates. The samples also demonstrate some Titanium Server extensions.</p>	<p>The incorrect example does not explain what the feature is or what it is used for.</p> <p>The correct example provides more information about the feature and why it should be used.</p>



## *Inline Elements*

[DITA Elements for Inline Content](#) 159

[Quick Reference Sheet of DITA Elements for Inline Content](#) 171

### DITA Elements for Inline Content

DITA elements for inline content are used for inline content that is not simple text (for example, a macro/library/function/method, a dialog title, or a key combination). Understanding when to use each element is key to producing accurate and consistent documentation.

This reference works like a reverse look-up for the *Wind River DITA Information Model* to tell you, for a specific type of content, what tag to use. The purpose of this reference is to guide writers, content development engineers, and editors in the use of correct and standard DITA tagging practices.



**NOTE:** Note the following bad practices:

- *Do not use the bold (<b>)* or italic (<i>) elements unless all other elements are inappropriate.
- *Do not use the <keyword>* element to format inline content.

#### GUI or Dialog Title

Use the <wintitle> element for GUI elements that users see on the screen, but that are merely containers for interactive GUI elements, these include views, dialog boxes, perspectives, windows, and wizards.

---

DITA Element:      <wintitle>

Code Example:

```
<p>Select the <wintitle>Debug</wintitle> perspective and ...</p>
<p>In the <wintitle>Project Explorer</wintitle> view, ...</p>
<p>In the <wintitle>Connect</wintitle> dialog box, ...</p>
```

Output Example:

Select the Debug perspective and ...  
In the Project Explorer view, ...  
In the Connect dialog box, ...

---

**NOTE:** The output of the `<wintitle>` element does not display as bold.

---

### Identifying a GUI Field or Button

Use the `<uicontrol>` element for names of GUI elements that a user must click, select, type in, or otherwise directly interact with. Interactive GUI elements include buttons, text boxes, drop-down lists, check boxes, menus, menu commands, tabs, and so on.

DITA Element:      `<uicontrol>`

Code Example:

```
<p>...in the <uicontrol>IP Address</uicontrol> field, type...</p>
<p>...click <uicontrol>Next</uicontrol>.</p>
```

Output Example:

...in the **IP Address** field, type...  
...click **Next**.

### Keyboard Keys, Combinations, and Sequences

Use the `<uicontrol>` element to define keyboard keys that must be pressed either by themselves, in combination with other keys, or in a sequence.

DITA Element:      `<uicontrol>`

Code Example:

```
<p>Press <uicontrol>ENTER</uicontrol>.</p>
<p>Press <uicontrol>ALT+F2</uicontrol> to...</p>
<p>Press <uicontrol>ALT, F, P</uicontrol> to...</p>
```

Output Example:

Press **ENTER**.  
Press **ALT+F2** to...  
Press **ALT, F, P** to...

### Menu Chain

Use the `<menucascade>` element to indicate a series of menu choices in an application. The `<menucascade>` element contains a series of `<uicontrol>` elements; one for each menu item in the chain.

---

DITA Element: <menucascade> followed by <uicontrol>

Code Example:

```
<p>From the menu, select <menucascade><uicontrol>File</uicontrol><uicontrol>Save As</uicontrol></menucascade>.</p>
```

Output Example: From the menu, select **File** > **Save As**.

---

**NOTE:** Do not type the > character that separates the menu selections in your DITA code - the DITA transformation automatically adds the > part of the chain.

---

### Command (in a Step or Body Paragraph)

*Command* refers to a command that a user executes on a CLI. Use the <cmdname> element to specify the name of a command when it is part of a discussion in a step or body paragraph.

If the command is presented as a separate line outside the step or body paragraph, use the <codeblock> and <userinput> elements - see [User Commands](#).

When describing the VxWorks kernel shell, use the <cmdname> element for command interpreter commands; for example, **rtp exec**. However, do not use the <cmdname> element for C interpreter commands; they are C functions.

---

DITA Element: <cmdname>

Code Example:

```
<p>Execute the <cmdname>cyclictest</cmdname> command to...</p>
```

```
<p>Execute the <cmdname>rtp exec</cmdname> command from the prompt.</p>
```

```
<p>Execute the <cmdname>bootm</cmdname> U-Boot command from the prompt.</p>
```

Output Example: Execute the **cyclictest** command to...

Execute the **rtp exec** command from the prompt.

Execute the **bootm** U-Boot command from the prompt.

---

### User Name and Password

Use the <nameliteral> element when instructing the user to type a literal string for either a user name or a specific password that represents the literal string that a user must type.

For a generic password that a user will customize on their installation, use the <varname> element - see [Placeholders in the Document](#).

---

DITA Element: <nameliteral>

Code Example:

```
<p>In the <uicontrol>User</uicontrol> text box, type the user name <nameliteral>wruser</nineliteral>.</p>
```

```
<p>When prompted, type the password <nameliteral>wrpwd</nineliteral>.</p>
```

Output Example:

In the **User** text box, type the user name **wruser**.

When prompted, type the password **wrpwd**.

---

### String to Enter in a Text Box

Use the **<nameliteral>** element when instructing the user to type a literal string into a dialog text box.

Use the **<uicontrol>** element for the name of the text box in which you type the literal string.

---

DITA Element:      **<uicontrol>** followed by **<nameliteral>**

Code Example:

```
<p>In the <uicontrol>IP Address</uicontrol> text box, type <nameliteral>10.20.30.123</nineliteral>.</p>
```

Output Example:

In the **IP Address** text box, type **10.20.30.123**.

---

### Path or File Name

Use the **<filepath>** element to define the name and location of a file or directory.

If your file path or file name contains a placeholder, use the **<varname>** element inside the **<filepath>** element. However, in cases where you use a wildcard character, such as \*, you do not need to use the **<varname>** element.

Place long file paths in a separate **<p>** element, so they render on a separate line. Examples of long file paths include:

- paths that are long enough to wrap over a line break
  - paths that make a sentence difficult to read
  - paths that mitigate line wrapping of the **<filepath>** element
- 

DITA Element:      **<filepath>**

Code Example:

```
<p>Save the file as <filepath>path/to/filename.ext</filepath>...</p>

<p>...in the directory <filepath><varname>installDir</varname>/path/to/directory</filepath>...</p>

<p>Open the <filepath>*.config</filepath> file.</p>

<p>Modify the <filepath>first<varname>thing</varname>.config</filepath> file to configure your project.</p>

<p>Save the file to the following location:</p>
<p><filepath>really/really/really/really/long/path/to/a/file/location</filepath><p>
```

Output Example:

Save the file as **path/to/filename.ext...**  
...in the directory *installDir/path/to/directory*...  
Open the **\*.config** file.  
Modify the **firstthing.config** file to configure your project.  
Save the file to the following location:  
**really/really/really/really/long/path/to/a/file/location**

---

**Library**

*Library* refers to the name of a software library. Use the **<libraryname>** element to reference the name of a library.

DITA Element:

**<libraryname>**

Code Example:

```
<p>...a static library (<libraryname>libadv.a</libraryname>) that contains...</p>

<p>The <filepath>libc.a</filepath> archive file provides the static <libraryname>libc</libraryname> library.
```

Output Example:

...a static library (**libadv.a**) that contains...

The **libc.a** archive file provides the static **libc** library.

---

**Macro**

*Macro* refers to a macro in a makefile or in source code. Use the **<macro>** element to reference the name of a macro.

Do not use the **<macro>** element for component names in a VxWorks configuration (.cdf) file, which are used in OS configuration with **wrtool**, **vxprj**, and Workbench. Instead, use the **<cdfcomponent>** element. For more information, see [VxWorks Component Description File Configuration Component](#) on page 170.

---

DITA Element:

**<macro>**

Code Example:

```
<p>...the macro <macro>ROMFS_DIR</macro>...</p>
```

```
<p>Use the <macro>FOOFILE</macro> macro as shown in the  
following code snippet.</p>  
<codeblock>#define FOOFILE foo.c</codeblock>
```

Output Example:

...the macro **ROMFS\_DIR**...

Use the **FOOFILE** macro as shown in the following code snippet.

```
#define FOOFILE foo.c
```

---

**Function (or Routine), Java Class, Method**

Use the **<apiname>** element for C function names (both Wind River APIs and user-defined functions), as well as for VxWorks C interpreter commands, which are C functions.

In the future, for VxWorks-based API reference documentation, items using the **<apiname>** element should render as an HTML link to the API reference documentation.

**NOTE:** If you do not type the () characters as part of the function name, the DITA transformation automatically adds them to the end of the function. Information Development recommends that authors add the () in the typed text, to make it very clear to authors that the () is part of the generated result. The generated output does not duplicate the () characters.

---

DITA Element:      **<apiname>**

Code Example:

```
<p>...in the function <apiname>main( )</apiname>...</p>
```

```
<p>Use the <apiname>printf( )</apiname> function to...
```

```
Update the <apiname>do_unpack( )</apiname> function...
```

Output Example:      ...in the function **main()**...

Use the **printf()** function to...

Update the **do\_unpack()** function...

---

**Function Arguments**

Use the **<option>** element when describing an argument or option of a CLI command in a text paragraph.

---

DITA Element:      **<option>**

Code Example:

```
<p>...with the argument <option>Arg_Name</option>...</p>
```

Output Example:      ...with the argument **Arg\_Name**...

## Parameter Name

Use the `<parmname>` element to reference the name of an application programming interface parameter within the text flow of your topic.

Do not use the `<parmname>` element for VxWorks configuration parameters included in a `.cdf` file, which are used in OS configuration with `wrtool`, `vxprj`, and Workbench. Instead, use the `<cdfcomponentparam>` element. For more information, see [VxWorks Component Description File Configuration Component Parameter](#) on page 170.

---

DITA Element:      `<parmname>`

Code Example:

```
<p>Use the <parmname>SHELL_STACK_SIZE</parmname> parameter to...</p>
```

```
<p>Set <parmname>wr_machine</parmname> to <option>${MACHINE}</option>...</p>
```

```
<p>Set <parmname>wr_with_template</parmname> to<br/><option>Feature/package-management</option> or <option>feature/build-toolchain</option>...</p>
```

Output Example:      Use the **SHELL\_STACK\_SIZE** parameter to...

Set **wr\_machine** to  **\${MACHINE}**...

Set **wr\_with\_template** to **feature/package-management** or **feature/build-toolchain**...

---

## An Option in a CLI Command

Use the `<option>` element to describe an option that can be used to modify a command.

---

DITA Element:      `<option>`

Code Example:

```
<p>Use the <option>-l</option> with the <cmdname>ls</cmdname> command to...</p>
```

Output Example:      Use the **-l** option with the **ls** command to...

---

## Configuration Option (Except for Configuration Options Included in a VxWorks .cdf File)

Use the `<option>` element to describe configuration options; for example, options in a Linux configuration file.

Do not use the `<option>` element for VxWorks configuration components or parameters that are listed in `.cdf` files, which are used in OS configuration with `wrtool`, `vxprj`, and Workbench. Instead, use the `<cdfcomponent>` and `<cdfcomponentparam>` elements. For more information, see [VxWorks Component Description File Configuration Component](#) on page 170 and [VxWorks Component Description File Configuration Component Parameter](#) on page 170.

---

DITA Element:      `<option>`

Code Example:

```
<p>Use the configuration option <option>ConfigOption</option>  
to...</p>
```

```
<p>...refer to the <option>POLLING_FREQUENCY</option>  
option...</p>
```

Output Example:

Use the configuration option **ConfigOption** to ...  
...refer to the **POLLING\_FREQUENCY** option...

---

### Placeholders in the Document

Use the **<varname>** element for placeholders for user-defined items, such as installation directories (for example, *installDir*), passwords, or project directories (for example, *projName*).

---

DITA Element:

**<varname>**

Code Example:

```
<p>Change to the <filepath><varname>installDir</varname>/  
vxworks-7</filepath> directory.</p>
```

```
<p>At the login prompt, enter <varname>yourPassword</  
varname>.</p>
```

Output Example:

Change to the *installDir/vxworks-7* directory.  
At the login prompt, enter *yourPassword*.

---

### Software Variable

Use the **<varname>** element to define software application variables.

---

DITA Element:

**<varname>**

Code Example:

```
<p>Assign the <option>FOO</option> option to  
<varname>SomeVariable</varname>...</p>
```

Output Example:

Assign the **FOO** option to *SomeVariable*...

---

### Makefiles

Generally, use the term **makefile** as a generic word (without the **<filepath>** element or initial capital). Use the **<filepath>** element only to specify the literal file name (with appropriate capitalization).

---

DITA Element:

plain text (no cap, no tag) for a file type or **<filepath>** for a file name

Code Example:

```
<p> Open the makefile, then...</p>  
<p> Open the file <filepath>Makefile</filepath>, then... </p>
```

Output Example:

Open the makefile, then...  
Open the file **Makefile**, then...

---

### Any Unspecified Literal String

Use the `<nameliteral>` element to reference items with literal names that do not otherwise have their own tag. Examples are names of command flags, constants, language keywords, operators, and register names.

---

DITA Element:      `<nameliteral>`

Code Example:

```
<p>...set the flag <nameliteral>O_RDWR</nameliteral> to  
<nameliteral>SomeValue</nameliteral> </p>
```

Output Example:     ...set the flag **O\_RDWR** to **SomeValue**...

---

### XML Element

Use the `<xmlelem>` element to identify an XML element.

---

DITA Element:      `<xmlelem>`

Code Example:

```
<p>Use the <xmlelem>startdate</xmlelem> element to...</p>
```

Output Example:     Use the `<startdate>` element to...

---

**NOTE:** The DITA transformation automatically adds the  
`<>` characters.

---

### XML Attribute

Use the `<xmlatt>` element to identify an XML attribute.

---

DITA Element:      `<xmlatt>`

Code Example:

```
<p>The <xmlatt>id</xmlatt> attribute assigns...</p>
```

Output Example:     The `@id` attribute assigns...

---

**NOTE:** The DITA transformation automatically adds the  
`@` character.

---

### XML Path

Use the `<>xpath>` element to mark up a full path through XML to an element or an attribute.

---

DITA Element:      `<>xpath>`

Code Example:

```
<p>The value must not be less than the corresponding <>xpath>/  
Partition/Settings/@MemorySize</xpath> attribute.</p>
```

Output Example: The value must not be less than the corresponding */Partition/Settings/@MemorySize* attribute.

---

#### Term to Define

Use the **<term>** element to identify words requiring an extended definition or explanation.

Consider adding any terms using this markup to your glossary.

---

DITA Element: **<term>**

Code Example: 

```
<p>The Titanium Server includes <term>compute</term> nodes...</p>
```

Output Example: The Titanium Server includes *compute* nodes...

---

#### Book Title

Use the **<cite>** element to identify the title of a book.

---

DITA Element: **<cite>**

Code Example: 

```
<p>...see the <cite>Wind River Workbench User's Guide</cite>.</p>
```

Output Example: ...see the *Wind River Workbench User's Guide*.

---

#### Trademark

Use the **<tm>** element to identify a term or phrase that is trademarked. This includes registered trademarks, service marks, slogans, and logos.

---

DITA Element: **<tm>** with **@tmtype** attribute

Code Example: 

```
<p>Wind River works with the <tm tmtype="reg">Foo</tm> third-party software.</p>
```

Output Example: Wind River works with the Foo® third-party software.

---

#### Mimicking a Terminal Session with User Commands and System Responses

Use the **<codeblock>** element along with the **<userinput>** and **<systemoutput>** elements to specify information the user must enter and the output or responses from a software program.

---

DITA Element: **<codeblock>** containing **<userinput>** and **<systemoutput>**

Code Example: 

```
<codeblock><systemoutput>$</systemoutput> <userinput>syncfoo -a -f myfile</userinput>
<systemoutput>gibberish system response</systemoutput></codeblock>
```

Output Example:

```
$ syncfoo -a -f myfile  
gibberish system response
```

---

### User Commands

Use the `<codeblock>` and `<userinput>` elements to specify information the user must enter at the command prompt.



**NOTE:** This DITA structure renders the content within the `<codeblock>` element as a separate line. To name a user command within a text block, use the [`<cmdname>`](#) element.

---

DITA Element:      `<codeblock>` containing `<userinput>`

Code Example:

```
<step>  
  <cmd>Execute the following command to ... </cmd>  
  <info>  
    <codeblock><systemoutput>$</systemoutput>  
    <userinput>syncfoo -a -f myfile</userinput></codeblock>  
  </info>  
</step>
```

Output Example:      Execute the following command to ...

```
$ syncfoo -a -f myfile
```

---

### System Output

Use the `<codeblock>` and `<systemoutput>` elements to indicate output or responses from a software program.



**NOTE:** This DITA structure renders the content within the `<codeblock>` element as a separate line - structure the preceding text to read as an independent explanation of what the subsequent system output line will demonstrate.

---

DITA Element:      `<codeblock>` containing `<userinput>`

Code Example:

```
<p>The following response indicates...</p>  
<codeblock><systemoutput>gibberish system response</systemoutput></codeblock>
```

Output Example:      The following response indicates...

```
gibberish system response
```

---

### Code Snippets

Use the `<codeblock>` element without the `<userinput>` or `<systemoutput>` elements to identify a code snippet.

A `<codeblock>` element that contains no other elements is formatted differently in the output than `<codeblock>` elements that contain `<userinput>` or `<systemoutput>` elements. The different formatting allows code snippets or examples to stand out from other types of code paragraphs.

---

DITA Element:      `<codeblock>` without `<userinput>` or `<systemoutput>`

Code Example:

```
<codeblock>if (hr < 10){  
    current_time = "0";  
    current_time += String(hr);  
}  
else if (hr == 0){  
    current_time = "12";  
}</codeblock>
```

Output Example:

```
if (hr < 10){  
    current_time = "0";  
    current_time += String(hr);  
}  
else if (hr == 0){  
    current_time = "12";  
}
```

---

### VxWorks Component Description File Configuration Component

Use the `<cdfcomponent>` to reference a component name in a component description file (CDF).

In the future, for VxWorks-based component reference documentation, this should render as an HTML link into the component reference documentation.

---

DITA Element:      `<cdfcomponent>`

Code Example:

```
<p>You must use the <cdfcomponent>INCLUDE_HRFS_FORMAT</cdfcomponent> component for formatting your hard drive.</p>
```

Output Example:

You must use the **INCLUDE\_HRFS\_FORMAT** component for formatting your hard drive.

---

### VxWorks Component Description File Configuration Component Parameter

Use the `<cdfcomponentparam>` to reference a parameter of a component in a component description file (CDF).

In the future, for VxWorks-based component reference documentation, this should render as an HTML link into the component reference documentation.

---

DITA Element:      `<cdfcomponentparam>`

Code Example:

```
<p>Use the <cdfcomponentparam>NUM_DRIVERS</cdfcomponentparam> parameter to specify the maximum number of device drivers in the system.</p>
```

Output Example:

Use the **NUM\_DRIVERS** parameter to specify the maximum number of device drivers in the system.

## Quick Reference Sheet of DITA Elements for Inline Content

DITA tags for inline content are used for inline content that is not simple text. A quick reference sheet can help refresh your memory of when to use each of the elements.

This is a simplified version of the more comprehensive [DITA Elements for Inline Content](#) on page 159, but without the code and output examples.

*Do not use* the bold (**<b>**) or italic (**<i>**) elements unless all other element are inappropriate.

*Do not use* the **<keyword>** element to format inline content.

The table is deliberately placed on the next PDF page - you can print that page to use as a pin-up on your office wall, for a quick reference guide.

Table 1 Quick Reference Sheet of DITA Tags for Inline Content

Type of Inline Content	DITA Elements
GUI/dialog title	<b>&lt;wintitle&gt;</b>
identifying a GUI field or button	<b>&lt;uicontrol&gt;</b>
keyboard keys, combinations, and sequences	<b>&lt;uicontrol&gt;</b>
menu chain	<b>&lt;menucascade&gt;&lt;uicontrol&gt;&lt;/uicontrol&gt;&lt;/menucascade&gt;</b>
command (in a step or body paragraph)	<b>&lt;cmdname&gt;</b>
user name and password	<b>&lt;nameliteral&gt;</b>

Type of Inline Content	DITA Elements
string to type in a dialog text box	<uicontrol> followed by <nameliteral>
path or file name	<filepath>
library	<libraryname>
macro	<macro>
function (or routine), Java class, method	<apiname>
function arguments	<option>
parameter name	<parmname>
an option in a CLI command	<option>
configuration option (except options included in a VxWorks .cdf file)	<option>
placeholder in the document	<varname>
software variable	<varname>
makefiles	plain text (no cap, no tag) for a file type <filepath> for a file name
any unspecified literal string	<nameliteral>
XML element	<xmlelem>
XML attribute	<xmlatt>
XML path	<>xpath>
term to define	<term>
book title	<cite>
trademark	<tm> with @tmtype attribute
mimicking a terminal with user commands and system responses	<codeblock> containing <userinput> and <systemoutput>
user commands	<codeblock> containing <userinput>
system output	<codeblock> containing <systemoutput>
code snippets	<codeblock> without <userinput> or <systemoutput>
VxWorks component description file configuration component	<cdfcomponent>
VxWorks component description file configuration component parameter	<cdfcomponentparam>

## *Wide Codeblock Elements*

In PDF output some codeblock elements may be too wide to be displayed within the page width. You can set the value of the `@outputclass` attribute to "wide" to tag these wide elements.

For example, if you have long lines of code that you do not want to wrap in the output, your DITA source should be similar to the following:

```
<codeblock outputclass="wide">for a in 1 2 3; do echo "how wide am i?"; echo " - just about
$a"; done</codeblock>
```

In the output, elements with an `@outputclass` set to "wide" will have a horizontal scroll. For example:

```
$ sudo su -
$ image-backup export 63aa7396-0587-4ee0-9a:
Exporting image: 100% complete...done.
Exporting image: 100% complete...done.
Creating backup archive...done
Backup archive /opt/backups/image_63aa7396-0
```



# 23

## *Graphics*

<a href="#">Wind River Graphics Standards</a>	175
<a href="#">Creating Images in the Ixiasoft CCMS</a>	181
<a href="#">Including Images in a Topic from the Ixiasoft CCMS Repository</a>	183
<a href="#">Editing Images</a>	184
<a href="#">Automated Image Resizing in PDF Output</a>	185
<a href="#">Creating Screen Captures</a>	185
<a href="#">Creating Images Using Visio 2007</a>	191

## **Wind River Graphics Standards**

The Wind River graphics standards help you create images that are consistent not only within the same document, but also across the documentation set. They also allow for better reuse of images.

### **Terms**

The discussion of graphics standards below uses the following terms:

*callout*

In a figure, text coupled with a line (called a leader) that points to some part of a drawing or imported bitmap.

*leader*

The straight line leading from a callout to that which it identifies. A leader should never be an arrow.

*label*

Text applied directly on a specific element that it identifies.

## General Guidelines

While this topic describes standards that are primarily for diagrams, much of them apply to figures that contain screenshots as well, because screenshot figures often include callouts and other diagram elements.

Lay out diagrams and screenshot callouts in a logical, clear, comprehensible, and consistent fashion. If you can add aesthetic orderliness, do so. Copy and paste commands are valuable for achieving this. Zooming in to 200%, 400%, or greater can help you clean up edges and spaces in a drawing. Use the graphic alignment and distribution functions to arrange objects and text blocks neatly.

## Line Widths

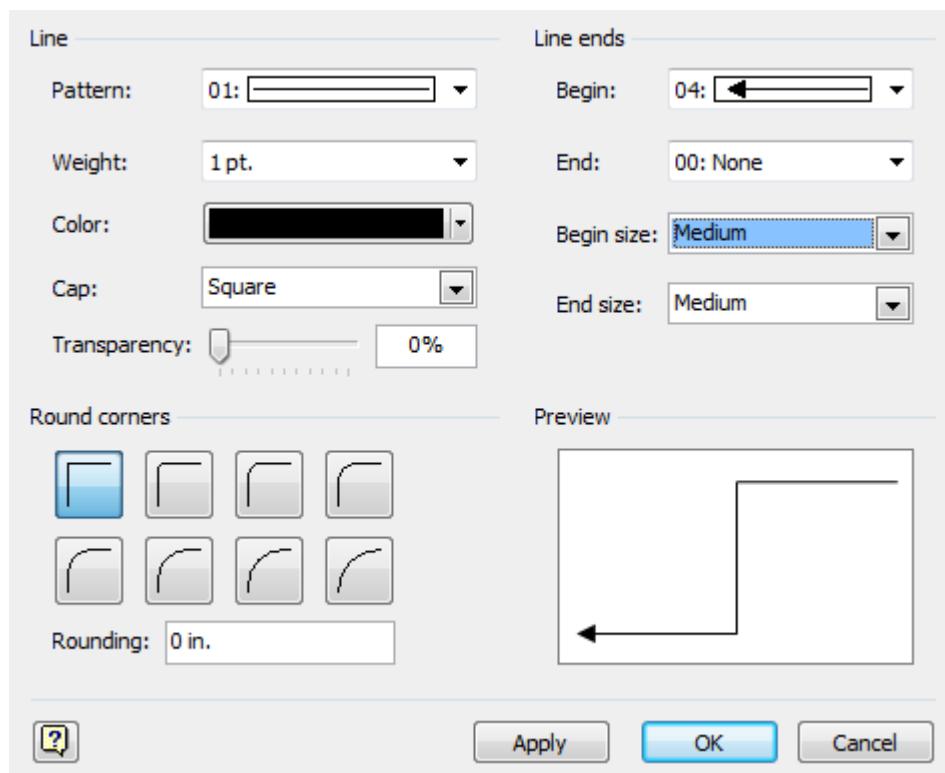
Apply the following line widths to the various elements of drawings:

- 0.2 pt – line width for callout leaders and braces
- 0.4 pt – line width for shapes
- 1.0 pt – line width for arrows, particularly in flow diagrams

## Arrows

Arrows and connector lines in flow diagrams should be formatted in Visio as follows:

Option	Value
Line	
Pattern	01
Weight	1 pt.
Cap	Square
Line ends	
Begin	00: None (for no arrow) 04 (for beginning arrow)
End	00: None (for no arrow) 04 (for end arrow)
Begin size	Medium (default)
End size	Choose size as appropriate based on the scale of the drawing.



## Color

Use the following guidelines when applying color to your drawings:

- Use only the following standard colors:

## Standard Color Palette

ID1 RGB = 191,191,191	ID2 RGB = 255,224,133	ID3 RGB = 117,163,209	ID4 RGB = 204,153,102
ID5 RGB = 112,184,184	ID6 RGB = 255,204,51	ID7 RGB = 204,0,0	ID8 RGB = 172,115,57
ID9 RGB = 51,102,102	ID10 RGB = 51,102,153		

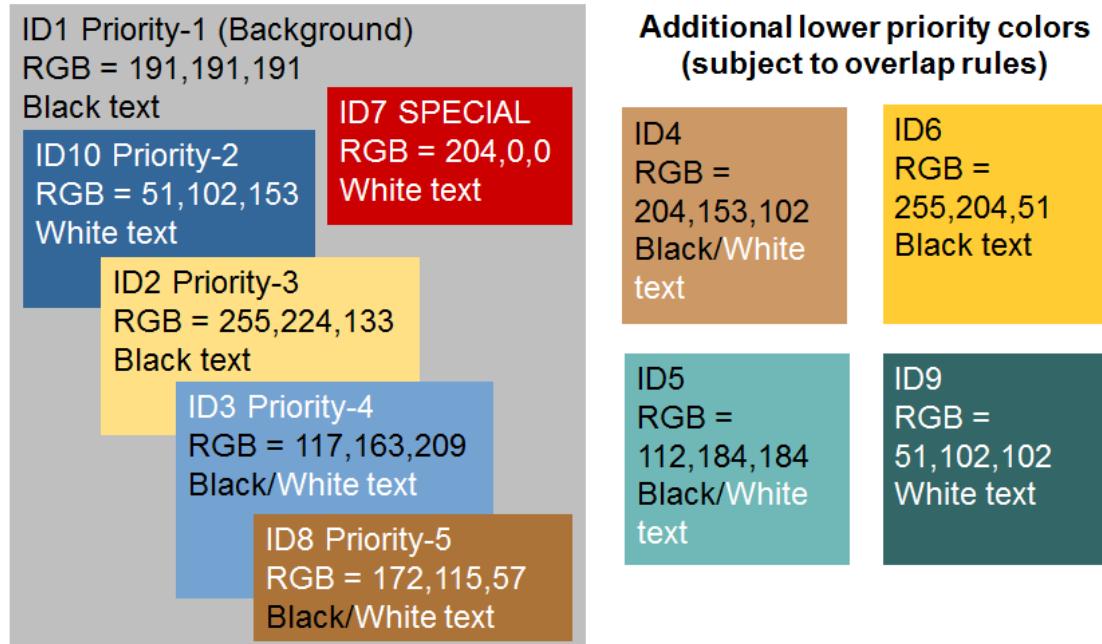
- Text in a colored box must adhere to the following rules:

## Text Color on Standard Color Blocks

Must use black	May use either (black preferred)	Must use white
ID1 RGB = 191,191,191 Black text	ID3 RGB = 117,163,209 Black/White text	ID7 RGB = 204,0,0 White text
ID2 RGB = 255,224,133 Black text	ID4 RGB = 204,153,102 Black/White text	ID9 RGB = 51,102,102 White text
ID6 RGB = 255,204,51 Black text	ID5 RGB = 112,184,184 Black/White text	ID8 RGB = 172,115,57 Black/White text
		ID10 RGB = 51,102,153 White text

- Follow the color hierarchy for your first five colors:

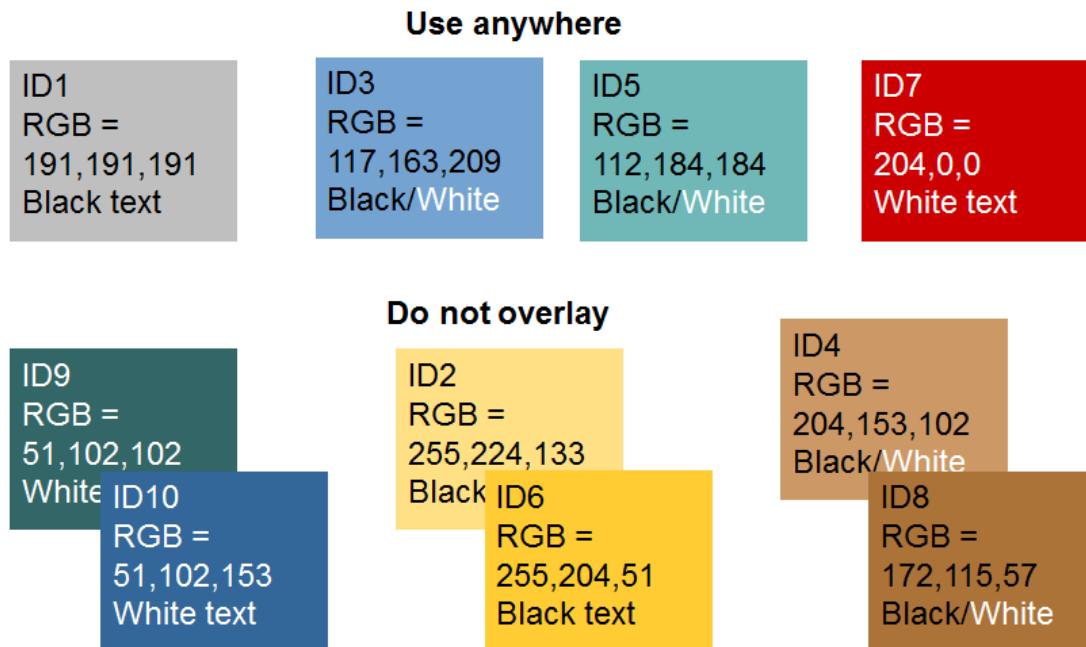
## Color Hierarchy and Usage



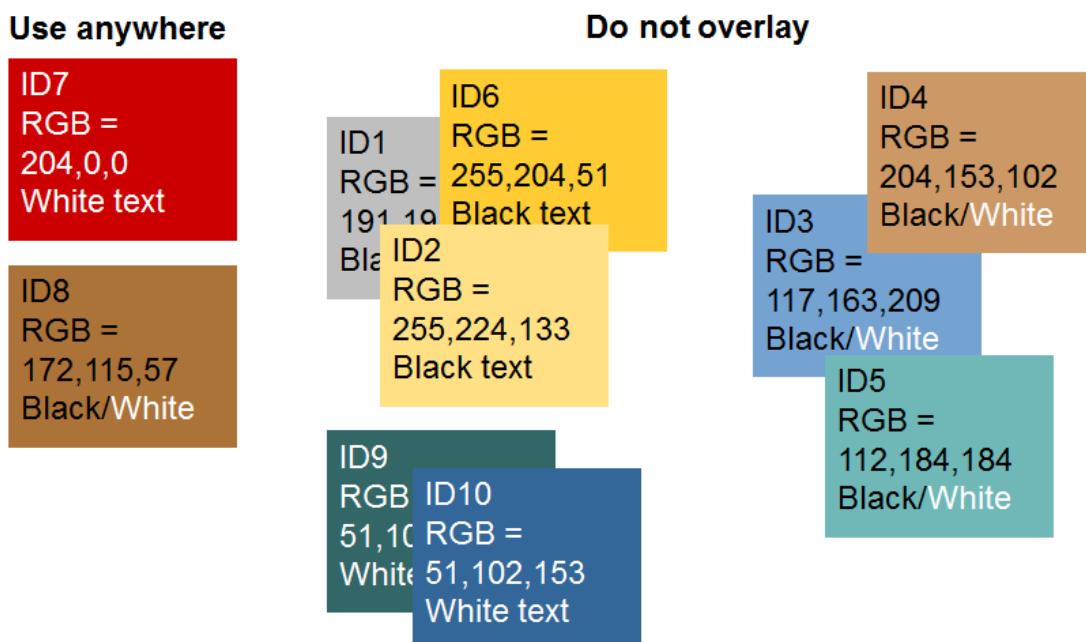
- Use Wind River red (ID7, RGB=204,0,0) to call attention to an important block, or for corporate branding purposes.
- Use low priority colors only when you have six or more colors.

- Always use the following color overlap rules:

## Standard Color Block Overlapping



## Standard Color Block Overlapping (B&W)



- Code snippet boxes must always use color ID2 (RGB=255,224,133)

### **Figure Labels**

Use Arial font. Tailor the size as necessary, generally from 8 to 12 points, but never less than 7 points. For emphasis, you can apply bold manually.

Do not create white labels on black or dark gray objects.

### **Callouts**

If a callout is a complete sentence, start it with a capital and end it with a period. However, most callouts are simply noun phrases that identify elements; do not treat these as sentences, and capitalize words only as necessary.

Draw callout leaders at angles so that they stand out against the horizontal/vertical orientation of screenshots. If you include numbered callouts, put the number in square brackets.

### **Drawing Boxes**

Do not apply rounded corners to boxes. In other words, do not use the rounded rectangle tool.

## **Creating Images in the Ixiasoft CCMS**

After you create your image (screen shot or line drawing) outside the Ixiasoft CCMS, you must import the image into the Ixiasoft CCMS. This is done by using the Create Image feature.

For each image, you can import up to three files. You can use this feature to keep together an original source file (for example, Visio or Inkscape) and two bitmapped versions (high-resolution and low-resolution), which the Ixiasoft CCMS could use for printed or online outputs. Currently, the Ixiasoft CCMS uses whichever file is set as default, regardless of output. You can set any one of the files as the default.

### **Procedure**

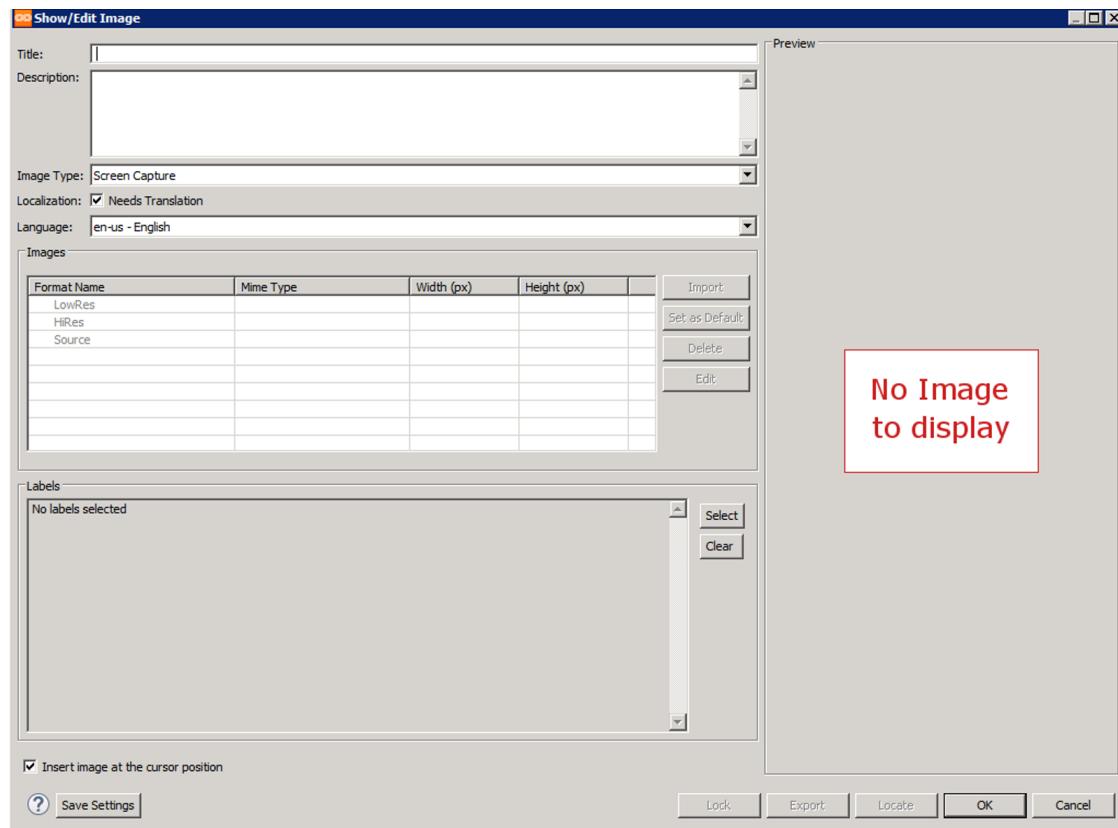
1. Create a screen shot using Snagit or a line drawing using Visio.
2. Save the image in a working directory.

You must save the image on the computer where the Ixiasoft CCMS is running.

<b>Client</b>	<b>Location</b>
<b>Local Client</b>	Save the file on your local host computer where you would normally store the image.
<b>Remote Desktop Client</b>	Save the image on the Ixiasoft CCMS remote desktop server (\\\ala-ixia-app\Shared) in a folder with your Windows user name (for example, for rechlin, \\\ala-ixia-app\Shared\rechlin). If the personal folder does not already exist, create it.

3. From the Ixiasoft CCMS menu, select **IXIASOFT CCMS > Create Image**.

The Show/Edit Image dialog box appears.



4. In the **Title** text box, type a title for the image.

The title should be descriptive of the image. For example, if the image is a screen shot of the Show/Edit Image dialog box, your title should be "The Show/Edit Image dialog box".

5. In the **Description** text box, type a description for the image.

Include the type of image. For example:

- Screen shot of the Show/Edit Image dialog box.
- Drawing of Foo.

6. From the **Image Type** drop-down list, select the image type.

7. If the image does not require localization, clear **Localization**.

An image only requires localization if it contains text that is not a literal (for example, a command). Currently Wind River does not localize images.

8. Select the appropriate format name.

Format	Description
<b>LowRes</b>	Use for bitmaps (.png files for screen shots and line drawings)
<b>HiRes</b>	Do not select - this functionality is not currently enabled.

Format	Description
Source	Use for source files (.vsd files, .svg files, and so forth)

**9.** Click **Import**.



**NOTE:** If you have a topic locked, **Insert image at the cursor position** is selected by default. If you do not want the image inserted into your locked topic, you must clear this selection before importing.

If the selection is not cleared, the CCMS tries to add the image at the cursor location. If it is not a valid location (for example, if the cursor is not inserted in a **<fig>** element), you will get an error message.

**10.** Browse to and select the image you created.

**11.** Click **Open**.

**12.** Click **OK**.

The image is created.

## Including Images in a Topic from the Ixiasoft CCMS Repository

Most images are managed in the Ixiasoft CCMS repository and included in topics from there.

### Procedure

**1.** Locate the image.

In the Search window, search with **Images** selected and a string from the image name or description in the Search for field.

**2.** Insert a **<fig>** element in your topic.



**NOTE:** Always insert your image within a **<fig>** element. Never insert an image directly within a text element (for example, **<cmd>** or **<p>**). Wind River style does not allow inline images within a text block, to avoid disrupted line spacing in PDF output.

**3.** Place the cursor inside the **<fig>** element.

**4.** In the Search Results window, right-click the image you want to add and select **Oxygen Editor > Insert Image**.

The image appears in the topic.

## Editing Images

To edit your images in any way, such as resizing or adding callouts, you must export them out of the Ixiasoft CCMS, make the changes to them, and reimport the them into the Ixiasoft CCMS.

### Procedure

1. Locate the image.

In the Search window, search with **Images** selected and a string from the image name or description in the Search for field.

2. In the Search Results window, right-click the image you want to edit and select **Edit**.
3. Click **Export** and select a location to save the image.

You must save the image on the computer where the Ixiasoft CCMS is running.

Client	Location
<b>Local Client</b>	Save the file on your local host computer where you would normally store the image.
<b>Remote Desktop Client</b>	Save the image on the Ixiasoft CCMS remote desktop server (\\\ala-ixia-app\Shared) in a folder with your Windows user name (for example, for <b>rechlin</b> , \\\ala-ixia-app\Shared\rechlin).  If the personal folder does not already exist, create it.

If the image contains multiple files (for example, LowRes and Source), each file is saved to the output location.

4. Make any required changes to the files.
5. In the Show/Edit Image dialog box, reimport the changed files to the appropriate format:

Format	Description
<b>LowRes</b>	Use for bitmaps (.png files for screen shots and line drawings)
<b>HiRes</b>	Not currently used, since Wind River does not print hard-copy documents.
<b>Source</b>	Use for source files (.vsd files, .svg files, and so forth)

6. Release the image.

## Automated Image Resizing in PDF Output

When you generate PDF output from the Ixiasoft CCMS, images within `<fig>` elements are automatically rendered to fit within the page dimensions.

As image size is rendered automatically, do not set manual `@height` or `@width` attribute values. You should only apply the `@height` and `@width` attribute value to force an image to display to a custom dimension. If you apply custom `@height` and `@width` attributes to your image, you will see a warning message when releasing the topic.

As of September 2015, use of the `@scale` attribute has been deprecated in the PDF outputs.

## Creating Screen Captures

### About Snagit 12

TechSmith Snagit is the Information Development group's preferred screen capture tool for Microsoft Windows PCs.

Snagit is designed for professional content developers that require advanced screen capture capabilities. Additionally, Snagit includes a comprehensive image editing tool, a library feature to organize images, and the ability to create short videos. For most content developers, Snagit is an effective replacement for PaintShop Pro and any other screen capture tool.

Snagit launches on start up and runs unobtrusively, ready to capture an image at any time. For a complete list of features, see:

<https://www.techsmith.com/snagit-features.html>

In addition to online help, complete documentation and short video tutorials are available at:

<https://www.techsmith.com/tutorial-snagit-12-documentation.html>

### Default Configuration

The Snagit 12 default screen capture settings are the same as the Information Development group's recommended screen capture settings, and are as follows:

- 96 dots per inch (DPI) capture resolution
- saves files in the `.png` format
- image resizing using resampling and lockable proportions

### Screen Capture Features

Snagit 12 has advanced screen capture capabilities. A few of the more advanced capabilities are as follows:

- captures vertical, horizontal, or complete scrolling windows
- a time-delay feature captures drop-down lists and tool tips, with or without the cursor
- capture area automatically adjusts to window borders
- a magnifier shows pixel level detail
- pre-defined and custom **Profiles** store capture settings, effects, and file destinations

### **Editing Features**

Some typical needs for content developers include the following:

- crop, trim, cut
- rotate, resize the image or canvas
- drawing tools including predefined callouts, stamps, and special effects
- the **.snag** file format allows continued editing after adding special effects

## **Creating Screen Captures Using Snagit 12 (Windows)**

Screen captures using Snagit are quick and intuitive.

### **Prerequisites**

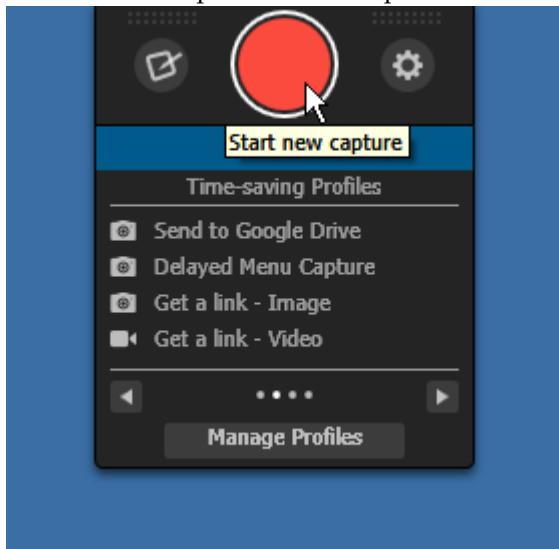
This procedure assumes you have installed Snagit 12 on a Windows based PC using the installation defaults.

### **Procedure**

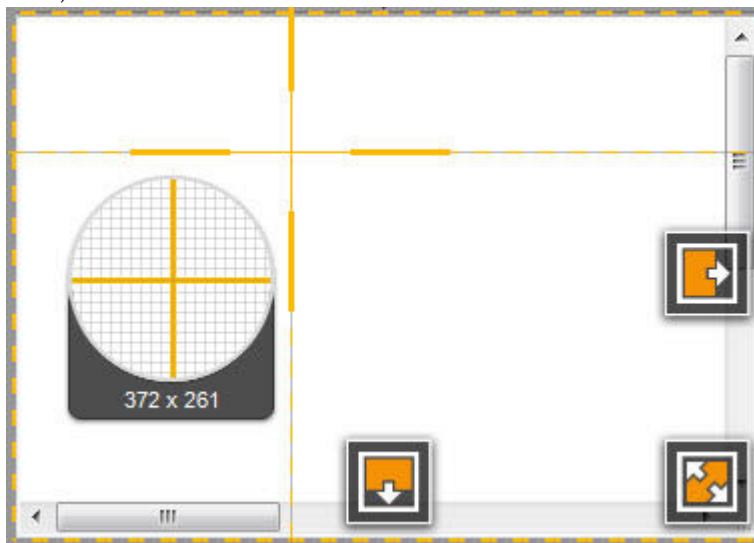
1. Display what you want to capture.
2. Move the cursor to the blue bar at the top of your screen to expand the Capture window.



3. Click the red Capture button or press Print Screen to initiate a screen capture.



4. Select the desired capture area or click an arrow to select a scroll area (vertical, horizontal or both).

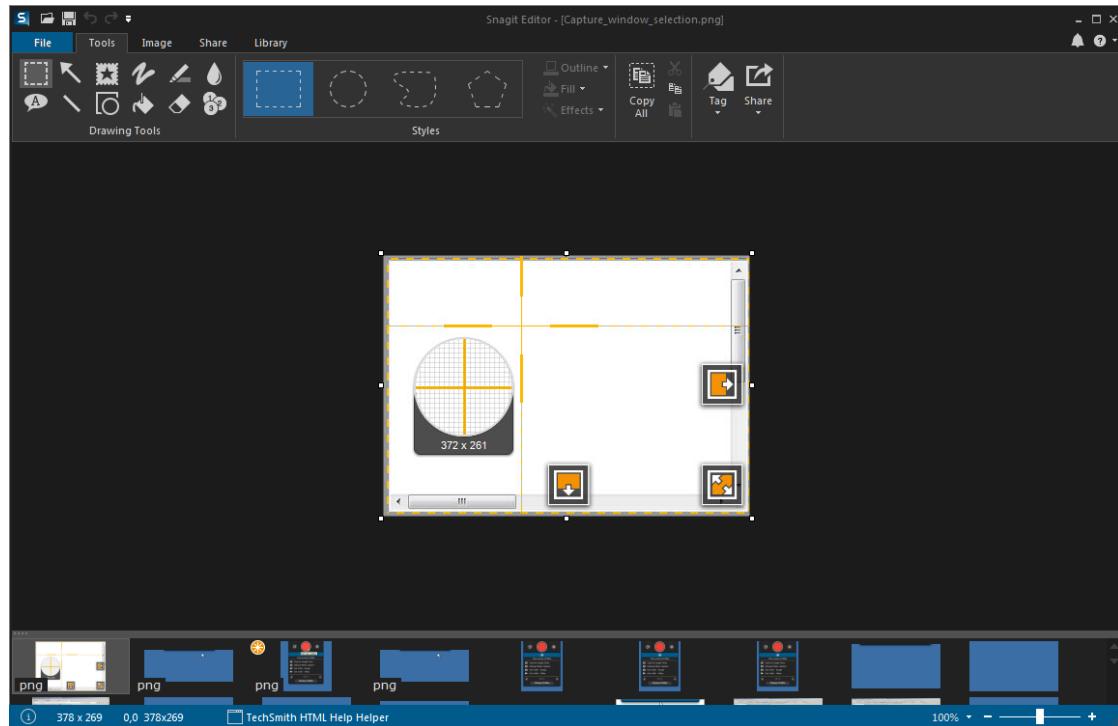


5. Click the Capture Image (camera) button to open the image in the Snagit Editor.



6. In the Editor, modify the image as necessary.

For drawing and callout styles and standards, see [Wind River Graphics Standards](#) on page 175.



7. Save the image.

By default, the image is saved in the **.png** file format. This format allows you to use the image in your topics, once you have imported the image to the Ixiasoft CCMS.

If you modify the image extensively with callouts, text or other effects, it is beneficial to also save the image in the Snagit file format (**.snag**). The **.snag** format saves all the effects so that you can modify the file and the effects separately in the future. If you only save the file as a **.png**, the effects become integrated into the file (flattened) and cannot be edited individually.

## Resizing Screen Captures Using Snagit 12

The Snagit Editor allows quick and simple resizing of images.

As a general rule, resizing bit mapped images (**.bmp**, **.jpg**, **.png**, and so forth) should be kept to a minimum, particularly when enlarging an image. Resizing, or making any changes to an image should always be done using an image editor outside the Ixiasoft CCMS.

**NOTE:** DITA elements and attributes that affect image size and appearance should not be used. Once an image is imported into the Ixiasoft CCMS its output is controlled exclusively by the transformation process.

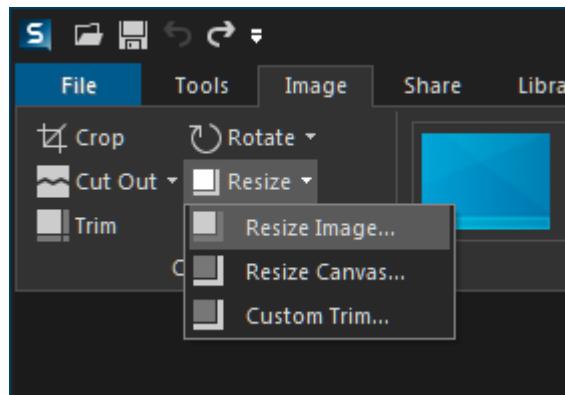
The Ixiasoft CCMS resizes any image larger than 5.75 inches wide to exactly 5.75 inches. Therefore, there is no need to perform any resizing of large screen captures unless you want the size to be smaller than 5.75 inches wide. In fact, using the larger screen capture allows the image to maintain its quality when the image is sized larger in a PDF file.

The typical Windows PC has a screen resolution of 96 dots per inch (DPI). Capturing a screen image at 96 DPI means that unless the application resizes the image, there will be no size difference between the captured and displayed images on a PC screen.

To resize an image, do the following:

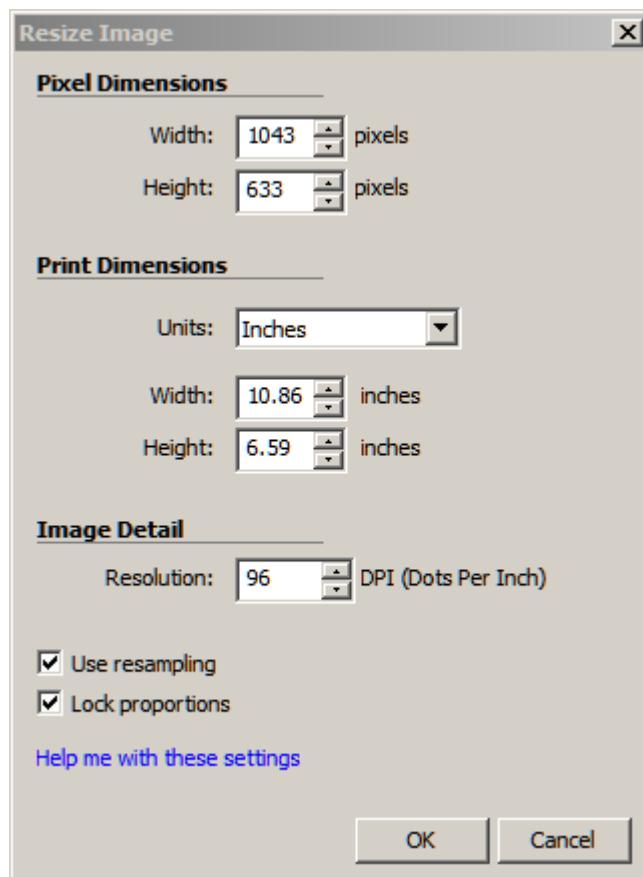
#### Procedure

1. Select **Image > Resize > Resize Image**.



2. Adjust the **Pixel Dimensions** or **Print Dimensions** to the desired size.

Whether you adjust the **Pixel Dimensions** or **Print Dimensions** is a matter of choice; they both accomplish the same resizing.



3. Leave the **Resolution** set to the default 96 DPI and leave the **Use resampling** and **Lock proportions** selected.
4. Click **OK**.

## Creating Screen Captures on Linux

To create screen captures on Linux, you must use a combination of Shutter and Gimp.

### Prerequisites

You must be familiar with and have Shutter and Gimp installed.

### Procedure

1. Open Shutter.
2. If necessary, change the image format to **.png** by selecting **Edit > Preferences > Main**.
3. Capture the image with Shutter using the appropriate option on the menu bar (**Selection, Window, Desktop**).
4. Optional: Edit the image to add highlight boxes or other callouts.
  - a) In the menu bar (not the **Edit** menu) click **Edit**.
  - b) Make your changes and click **Save**.
5. Optional: Check and modify the size for use in PDFs.
  - a) Select **Screenshot > Open with > Gimp Image Editor**.
  - b) In the Gimp window, select **Image > Print Size**.
  - c) In the Width and Height section of the Set Image Print Resolution window, select **inches**.
  - d) If the width is greater than 6, change the value to **6**.  
The height should adjust proportionally.
6. Optional: Do any further editing in Gimp.
7. Select **File > Export as**.
8. Optional: Rename the file to avoid overwriting the original screen capture by typing the file name in the **Name** text box.
9. In the menu in the lower-right corner, select **PNG image** and click **Export**.
10. In the Export Image as PNG window, click **Export**.

## **Creating Images Using Visio 2007**

### **Connecting to Visio 2007 Using the Remote Desktop Application**

Visio 2007 is located on a virtual server. To access it, you must connect to the server using the Remote Desktop application.

#### **Procedure**

1. Start the Remote Desktop application:
  - a) Click your Start menu and type **Remote desktop** in the **Search programs and files** field.
  - b) Select **Remote Desktop Connection** in the results window.
2. Configure your connection:
  - a) In the Remote Desktop Connection dialog box, type **ala-tpvm-apps** for the computer name.
  - b) Click **Show Options**.
  - c) Select the **Local Resources** tab.
  - d) Under Local devices and resources, click **More**.
  - e) Select **Drives**, and click **OK**.
3. Connect to the server:
  - a) Click **Connect**.
  - b) When asked if you trust the connection, click **Connect**.
  - c) Type your Windows password, and click **OK** to log in.
4. Start Visio 2007.

From the Start menu of the remote desktop, select **Microsoft Office Visio 2007**.

### **Setting the Text Quality Options**

To ensure the text in your images displays properly in both HTML and PDF output, you must set the text quality options. This task only needs to be done once.

#### **Procedure**

1. Select **Tools > Options**.
2. Select the **View** tab.
3. In the Text quality section, select **ClearType text display (anti-aliased, LCD displays)**.
4. Click **OK**.

## Creating a Drawing Using a Stencil

A stencil is a collection of shapes you can use to create your images by dragging and dropping them from the stencil onto your drawing page.

Using the Wind River stencil saves you time and preserves consistency across your drawings, because all the shapes in the stencil conform to Wind River graphics standards.

Alternately, you can base your drawings on the Wind River template. For more information, see [Creating a Drawing Using a Template](#) on page 192.

### Procedure

1. Open your drawing.

You can either open an existing drawing or create a new drawing by selecting **File > New Drawing (US units)**.

2. Open the Wind River stencil.

Select **File > Shapes > Open Stencil** and navigate to:

**T:\templates\visio\wr-standard-stencil.vss**

The Wind River stencil opens in the right pane.

3. Add shapes to your drawing by dragging them from the stencil and dropping them on your drawing page.

## Creating a Drawing Using a Template

A template contains a group or groups of predefined shapes on which you can base your drawing. You can copy, paste, and modify the shapes to meet your needs.

Using a Wind River template saves you time and preserves consistency across your drawings, because all the shapes in the template conform to Wind River graphics standards.

Alternately, you can create your drawings using the Wind River stencil. For more information, see [Creating a Drawing Using a Stencil](#) on page 192.

### Procedure

Create a new drawing by opening a template in one of the following ways:

Option	Method
<b>From Visio:</b>	<ol style="list-style-type: none"><li>1. Select <b>File &gt; New Drawing from Template</b>.</li><li>2. Navigate to: <b>T:\templates\visio\wr-generic-template.vst</b></li></ol>
<b>From an explorer window:</b>	<ol style="list-style-type: none"><li>1. Navigate to: <b>T:\templates\visio</b></li><li>2. Double-click <b>wr-generic-template.vst</b>.</li></ol>

## Saving the Image Source

Saving the image source file allows you to make future changes or edits.

### Procedure

1. Select **File > Save**.

**NOTE:** You can save your image locally in **C:\users\userName.CORP**, which is convenient for quick saves during your work. However, when you are ready to import your image to the Ixiasoft CCMS, you must save it on the computer where the Ixiasoft CCMS is running. For more information, see [Creating Images in the Ixiasoft CCMS](#) on page 181.

2. Type a descriptive name for the file in the **File name** field of the Save dialog box.
3. Ensure **Drawing** is selected in **Save as type**.
4. Click **Save**.

The image is saved as a **.vsd** file.

## Saving an Image as a **.png** File

Saving an image as a **.png** file allows it to be included in your topics. The **.png** format works well in both HTML and PDF output.

### Prerequisites

For optimal results in the output, you must first set the text quality options appropriately. For more information, see [Setting the Text Quality Options](#) on page 191.

### Procedure

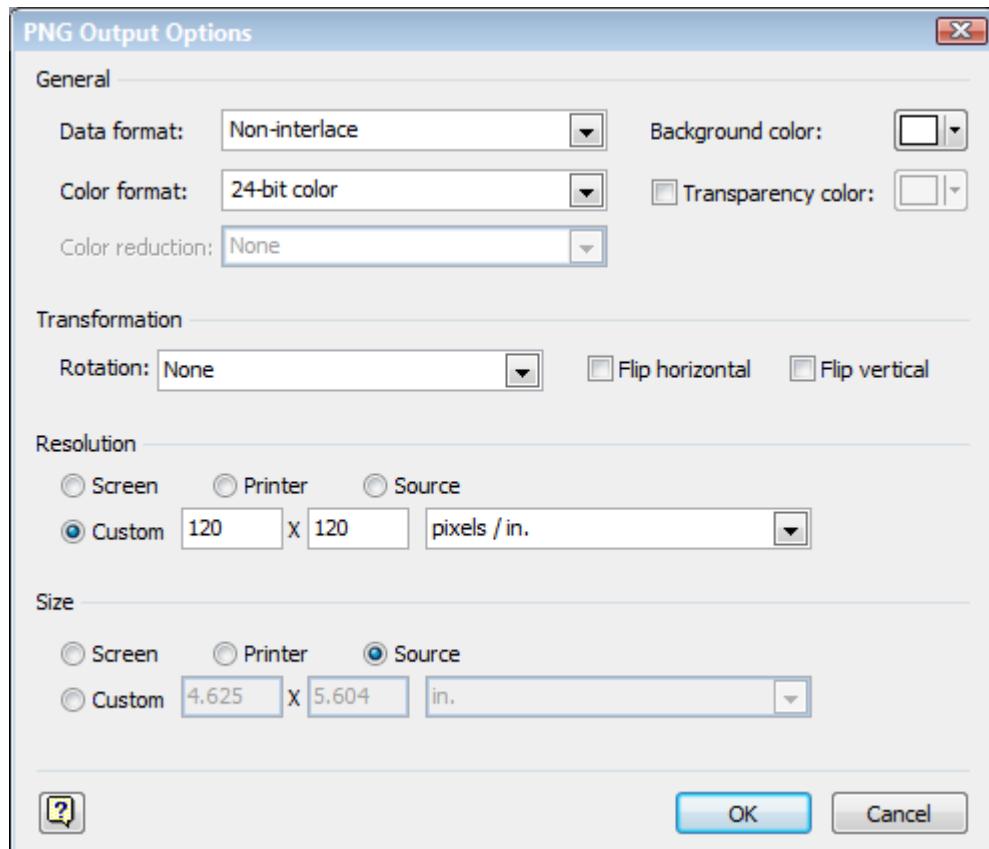
1. Select **File > Save As**.

**NOTE:** You can save your image locally in **C:\users\userName.CORP**, which is convenient for quick saves during your work. However, when you are ready to import your image to the Ixiasoft CCMS, you must save it on the computer where the Ixiasoft CCMS is running. For more information, see [Creating Images in the Ixiasoft CCMS](#) on page 181.

2. Type a descriptive name for the file in the **File name** field of the Save As dialog box.
  3. Select **Portable Network Graphics (\*.png)** from the **Save as type** drop-down list.
  4. Click **Save**.
- The PNG Output Options dialog box appears.
5. Select the following options in the PNG Output Options dialog box:

Section	Option	Value
General	Data Format	Non-interlace
	Color Format	24-bit color
Transformation	Rotation	None
Resolution	Custom	120 x 120 pixels/in.
Size	Source	N/A

For example:



## Visio Quick Reference

This quick reference table describes how to perform actions commonly used in Visio.

Action	Method
Add a new shape	Drag and drop a shape from the Wind River stencil onto your page. For more information, see <a href="#">Creating a Drawing Using a Stencil</a> on page 192. Or, use the drawing tools:

Action	Method						
	<ol style="list-style-type: none"> <li>1. Click on the <b>Drawing Tools</b> icon.   </li> <li>2. Select the desired shape from the Drawing toolbar.</li> </ol> <hr/> <p><b>NOTE:</b> If you use the drawing tools to add a shape, you must make sure the line width meets the Wind River standard. For more information about the standard, see <a href="#">Wind River Graphics Standards</a> on page 175.</p>						
Select a single shape	<ol style="list-style-type: none"> <li>1. Click on the <b>Pointer Tool</b> icon.   </li> <li>2. Click on the desired shape.  Green handles appear on the shape when it is selected.</li> </ol>						
Select multiple shapes	<ol style="list-style-type: none"> <li>1. Click the drop-down list next to the <b>Pointer Tool</b> icon.   </li> <li>2. Select the desired option:</li> </ol> <hr/> <table> <tbody> <tr> <td style="vertical-align: top;"> <b>Area Select</b> </td><td>Allows you to draw a rectangle around the shapes you want to select.</td></tr> <tr> <td style="vertical-align: top;"> <b>Lasso Select</b> </td><td>Allows you to draw a "lasso" around the shapes you want to select.</td></tr> <tr> <td style="vertical-align: top;"> <b>Multiple Select</b> </td><td>Allows you to click multiple shapes to select them.</td></tr> </tbody> </table> <hr/> <p>Or, you can select a single shape, then press <b>CTRL</b> and click on another shape to select it.</p>	<b>Area Select</b>	Allows you to draw a rectangle around the shapes you want to select.	<b>Lasso Select</b>	Allows you to draw a "lasso" around the shapes you want to select.	<b>Multiple Select</b>	Allows you to click multiple shapes to select them.
<b>Area Select</b>	Allows you to draw a rectangle around the shapes you want to select.						
<b>Lasso Select</b>	Allows you to draw a "lasso" around the shapes you want to select.						
<b>Multiple Select</b>	Allows you to click multiple shapes to select them.						
Align shapes	<ol style="list-style-type: none"> <li>1. Select the desired shapes.</li> <li>2. Select <b>Shape &gt; Align Shapes</b>.</li> <li>3. Select the desired option from the Align Shapes dialog box.</li> </ol>						
Distribute shapes	<ol style="list-style-type: none"> <li>1. Select the desired shapes.</li> <li>2. Select <b>Shape &gt; Distribute Shapes</b>.</li> <li>3. Select the desired option from the Distribute Shapes dialog box.</li> </ol>						
Group shapes	<ol style="list-style-type: none"> <li>1. Select the desired shapes.</li> </ol>						

Action	Method
	<ol style="list-style-type: none"><li>2. Select <b>Shape &gt; Grouping &gt; Group</b>. Or, press <b>SHIFT+CTRL+G</b>.</li></ol>
Ungroup shapes	<ol style="list-style-type: none"><li>1. Select the desired group of shapes.</li><li>2. Select <b>Shape &gt; Grouping &gt; Ungroup</b>. Or, press <b>SHIFT+CTRL+U</b>.</li></ol>
Order shapes	<ol style="list-style-type: none"><li>1. Select the desired shapes.</li><li>2. Select <b>Shape &gt; Order</b>.</li><li>3. Select one of the following options:<ul style="list-style-type: none"><li>• <b>Bring to Front</b></li><li>• <b>Send to Back</b></li><li>• <b>Bring Forward</b></li><li>• <b>Send Backward</b></li></ul></li></ol>
Rotate or flip shapes:	
Rotate Left	<ol style="list-style-type: none"><li>1. Select the desired shape.</li><li>2. Select <b>Shape &gt; Rotate or Flip &gt; Rotate Left</b>. Or, press <b>CTRL+L</b>.</li></ol>
Rotate Right	<ol style="list-style-type: none"><li>1. Select the desired shape.</li><li>2. Select <b>Shape &gt; Rotate or Flip &gt; Rotate Right</b>. Or, press <b>CTRL+R</b>.</li></ol>
Flip Horizontal	<ol style="list-style-type: none"><li>1. Select the desired shape.</li><li>2. Select <b>Shape &gt; Rotate or Flip &gt; Flip Horizontal</b>. Or, press <b>CTRL+H</b>.</li></ol>
Flip Vertical	<ol style="list-style-type: none"><li>1. Select the desired shape.</li><li>2. Select <b>Shape &gt; Rotate or Flip &gt; Flip Vertical</b>. Or, press <b>CTRL+V</b>.</li></ol>
Set line weight	<ol style="list-style-type: none"><li>1. Select the desired line or shape.</li><li>2. Select <b>More Line Weights</b> from the drop-down list next to the <b>Line Weight</b> icon.  The Line dialog box appears.</li><li>3. Select <b>Custom</b> from the <b>Weight</b> drop-down list.</li></ol>

Action	Method
	<p>4. Type the appropriate line weight. For more information about Wind River standard line weights, see <a href="#">Wind River Graphics Standards</a> on page 175.</p>
	<p>5. Select <b>Square</b> from the <b>Cap</b> drop-down list.</p>
Set line ends (arrows)	<p>1. Select the desired line or shape.</p> <p>2. Select <b>More Line Ends</b> from the drop-down list next to the <b>Line Ends</b> icon.</p>  <p>The Line dialog box appears.</p> <p>3. Select <b>04:</b> from the <b>Begin</b> and/or <b>End</b> drop-down lists.</p> <p>4. Select <b>Square</b> from the <b>Cap</b> drop-down list.</p>
Insert a text box	<p>1. Select <b>Insert &gt; Text Box</b>.</p> <p>2. Create a text box of the desired size.</p> <p>3. Add your text.</p>
<p><b>NOTE:</b> Text should be in Arial font with a font size between 8 and 12 points.</p>	
Rotate text	<p>1. Select the desired text box.</p> <p>2. Select <b>Shape &gt; Rotate or Flip &gt; Rotate Text</b>.</p>
Spell check	<p>Select <b>Tools &gt; Spelling &gt; Spelling</b>. Or, press <b>F7</b>. Or, click on the <b>Spell Check</b> icon.</p> 



# 24

## Links

<a href="#">Updating @scope Attributes in Cross-References Using an XSLT Script</a>	199
<a href="#">Creating Links Between Topics in the Same Book</a>	201
<a href="#">Creating Links to Targets Within Topics</a>	201
<a href="#">Generating ID Attributes</a>	202
<a href="#">Creating Links to External Sources</a>	203
<a href="#">Creating Links to a Topic in a Different Book</a>	203
<a href="#">Creating Links to Documents on the Product Documentation Site</a>	205
<a href="#">Creating Links to Hierarchy Locations on the Product Documentation Site</a>	206
<a href="#">Creating Links to Videos on the Product Documentation Site</a>	207

### Updating @scope Attributes in Cross-References Using an XSLT Script

To enable cross-reference links to work correctly on the Product Documentation site, the different types of cross-references in DITA source must be properly identified by adding a `@scope` attribute to `<xref>` elements.

The `@scope` attribute takes values that depend on the type of link you are creating:

- Set `@scope` to `local` when the target of the link is part of the current set of content.
- Set `@scope` to `peer` when the target of the link is part of the current set of content but is not accessible at build time.
- Set `@scope` to `external` when the target of the link is not part of the current information set.

Cross-references in existing DITA source may not include the @scope attribute. To update your existing DITA source and ensure your cross-references will work properly, you can add the @scope attribute and associated values to your links manually or you can use the following instructions to run an XSLT script on all of the topics in your documentation set. The script identifies the type of link and adds the @scope attribute and appropriate value.

#### Procedure

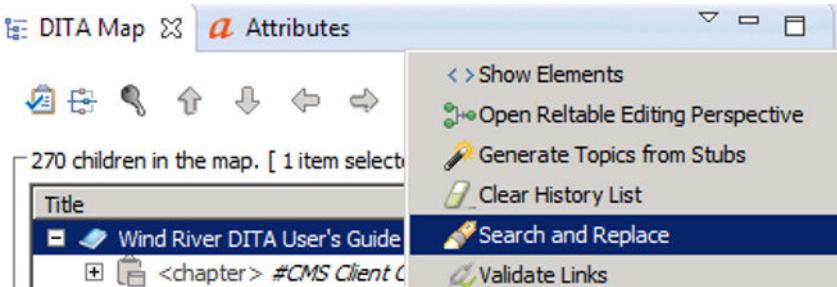
1. Navigate to \\ala-nas2\docs\tools\Ixiasoft-DITA-CMS\XSLT-scripts.
2. Save the **add-scope-attribute.xsl** script on your local drive.
3. In your Ixiasoft CCMS client, search for the documentation (bookmap or Eclipse Help map, if you are modifying topics in several bookmaps) you want to modify.
4. In the DITA Map view, open the bookmap or the parent Eclipse Help map.
5. In the map hierarchy, lock all of the topics you want to update.

---

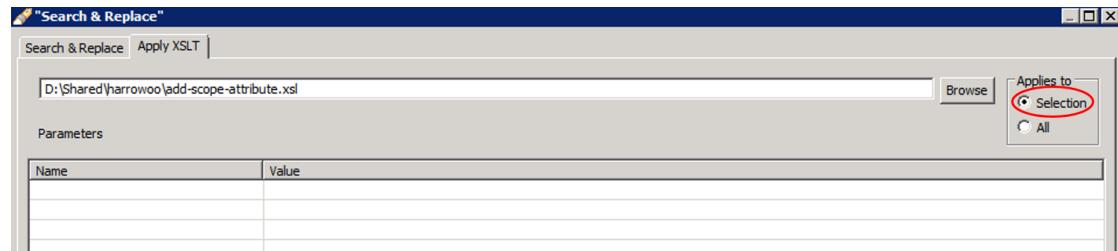
**NOTE:** The topics must be in the authoring:work state.

---

6. Select all the topics you locked in the previous step.
7. Click the View menu button (the downward facing triangle icon) at the upper-right corner of the DITA Map editor and select **Search and Replace**.



8. In the Search and Replace dialog box, select the **Apply XSLT** tab and browse to the XSLT file on your local drive.
9. Under Applies to, select **Selection**.



---

**CAUTION:** If you have **All** selected instead of **Selection** and your bookmap is locked, the script will run on your bookmap and may make it invalid.

---

**10. Click Apply.**

Unfortunately, the Ixiasoft CCMS does not show progress, but it applies the XSLT to the topics you have selected and refreshes the workspace.

**11. Review the <xref> elements in your topics to ensure they have been properly updated.****12. Release your topics to save the changes.**

## Creating Links Between Topics in the Same Book

Cross-references between topics are formed by links. You can create links either directly in XML in a text editor or using the oXygen editor Author view.

You can use this method for inline links and related links.

**Procedure**

1. Open the topic and place the cursor where you want to place the link.

You can work in either Text or Author mode.

2. Locate the topic you want to link to.

Use Search if the topic is not in the current map.

3. Right-click the topic you want to link to, and select **Oxygen Editor > Insert as XRef**.

An <xref> element is added to your topic, with the @href attribute set to the ID of the topic you are linking to. For example:

```
<xref href="jan1236299873904.xml"/>
```

4. Add a @scope attribute and set the value to **local**.

```
<xref href="jan1236299873904.xml" scope="local"/>
```

## Creating Links to Targets Within Topics

Cross-references to elements within a topic are formed by links to the element ID. You can create links either directly in XML in a text editor or using the oXygen editor Author view.

In most cases, best practice is to link only to the topic, not to an element within a topic. However, there are exceptions to this rule. A key exception is when you need to refer to a specific step. For example, you could say, "If x happens, go to step y."

You can also create cross-references to the glossary terms in your glossary; however, you must use a slightly different format. For more information, see [Referencing a Glossary Entry](#) on page 150.

## Prerequisites

The element you are linking to must have an **@id** attribute defined. For more information, see [Generating ID Attributes](#) on page 202.

## Procedure

1. Open the topic that will contain the cross-reference.

This can be the same topic if you are referring to an element in the same topic.

2. Locate the topic you want to link to if it is a different topic.

Use Search if the topic is not in the current map.

3. Copy the file name of the topic you are linking to.

a) Right-click the topic you want to link to.

b) Select **Copy > Copy Reference**.

The file name is copied to the clipboard.

4. Add an **<xref>** element in the original document.

5. Define the **@href** attribute for the **<xref>**.

Paste the file name into the **@href** attribute if you are linking to a separate topic.

6. Copy the element ID for the target element and add it to the link.

See the examples below for the formats for linking to elements in the same topic and linking to elements in a different topic. In both of these examples, **jan1236299873904** is the topic ID of the topic and **step\_N1002B\_N10028\_N1001C\_N10001** is the element ID.

### Example: Linking to Elements in the Same Topic

```
<xref href="#jan1236299873904/step_N1002B_N10028_N1001C_N10001">
```

### Example: Linking to an Element in a Different Topic

```
<xref href="jan1236299873904.xml#jan1236299873904/step_N1002B_N10028_N1001C_N10001">
```

## Generating ID Attributes

An **@id** attribute is required when you create a link to a target within a topic. In most cases, **@id** attributes are generated automatically by the Ixiasoft CCMS. However, if the element you are trying to link to does not have an **@id** attribute, you must generate one.

## Procedure

1. Lock and open the target topic.
2. Insert your cursor in the element for which you want to create an **@id** attribute.

3. Right-click and select **Generate IDs**.

The **@id** attribute is generated.

4. Save and release your topic.

## Creating Links to External Sources

You can create inline links or related links to external sources, such as Web sites and email addresses.

### Procedure

1. Add an **<xref>** element to your topic.
2. Define the **@href** attribute for the **<xref>**.

Options	Description
<b>For a Web site</b>	Add the URL.
<b>For an email address</b>	Add <b>mailto:</b> and the email address.

3. Set the **@scope** attribute to **external**.
4. For Web sites, set the **@format** attribute to **html**
5. Add link text to the **<xref>** element.

### Examples

For a link to a Web site, your **<xref>** element should look similar to the following:

```
<xref scope="external" format="html" href="http://www.windriver.com/support">Wind  
River Online Support Web site</xref>
```

For a link to an email address, your **<xref>** element should look similar to the following:

```
<xref scope="external" href="mailto:support@windriver.com">support@windriver.com</xref>
```

## Creating Links to a Topic in a Different Book

You can create links from one topic to a topic in a different book. The topics do not need to be in the same plugin or Eclipse Help map.

Linking to topics in other documents can be confusing for customers if it is not done properly. Because the links are not live in PDF output, you must specify the book name and topic name. In HTML output, including the book name also helps orient customers to the fact that clicking on

the link will take them out of the current document and into another. You must use `<cite>` and `<ph>` elements to format the links to ensure they show up properly in both PDF and HTML.

#### Procedure

1. Open the topic and place the cursor where you want to place the link.

You can work in either Text or Author mode.

2. Add a `<cite>` element.

3. Add the book title, followed by a colon and a space:

```
<cite>VxWorks Programmer's Guide: </cite>
```

4. Add a `<ph>` element after the book name, colon, and space:

```
<cite>VxWorks Programmer's Guide: <ph></ph></cite>
```

5. Insert the cursor between the opening and closing tags of the `<ph>` element you just created.

6. Locate the topic you want to link to.

Use Search if the topic is not in the current map.

7. Right-click the topic you want to link to, and select **Oxygen Editor > Insert as XRef**.

An `<xref>` element is added to your topic, with the `@href` attribute set to the ID of the topic you are linking to. For example:

```
<cite>VxWorks Programmer's Guide: <ph><xref href="jan1236299873904.xml"/></ph></cite>
```

8. Set the `@scope` attribute to `peer`.

For example:

```
<cite>VxWorks Programmer's Guide: <ph><xref href="jan1236299873904.xml" scope="peer"/></ph></cite>
```

9. Add the topic title as the link text.

10. If the link is going to a topic in another product, add a `<data>` element to your `<xref>` element.

If you are linking to a topic in the same product, you can skip this step.

However, if you want to link from a topic, for example, in the Helix Virtualization Platform Getting Started, SR0620 guide to a topic in the VxWorks 7 Programmer's Guide, you must ensure the link goes to the SR0620 version of the documentation. Since topic IDs may not change from release to release, you must specify which version of the product you want to link to. However, you can only have one `<othermeta>` element setting the facets to your product version; for example:

```
<othermeta content="version=os_helix_virtualization_platform_sr0620" name="facets"/>
```

Therefore, to ensure you are going to link to the proper version, you must override the version applied to the book in your `<xref>` by adding a `<data>` element.

- a) Inside the `<xref>` element, add a `<data>` element.

- b) Add a @name attribute to the <data> element and set the value to **facets**.
- c) Add a @value attribute to the <data> element and set the value to the Vocabulary Term Value for the product version.

This value is the term for the product you are linking to under the Vocabulary Term Value column in the Taxonomy Terms pane. For example, if you are linking to the SR0620 version of VxWorks 7, the value would be **os\_vxworks\_7\_sr0620**.

The value you place in the <data> element overrides the value in the <othermeta> element for this link only.

```
<p>For more information, see the: <cite>VxWorks Programmer's Guide:</cite>
<ph><xref href="jan1236299873904.xml" scope="peer"><data name="facets"
value="version=os_vxworks_7_sr0620"/>Using Dosfs</xref></ph></p>
```

## Creating Links to Documents on the Product Documentation Site

You can create links to the "top" page of a document on the Product Documentation site by linking to the first topic in that document.

**NOTE:** In the past, these links were created either by using the content ID (old Knowledge Library) or by using the bundle name of the document. If your documents contain either types of these links, they must be updated to the current method.

### Procedure

1. Open the topic and place the cursor where you want to place the link.  
You can work in either Text or Author mode.
2. Locate the first topic in the book you want to link to.
3. Right-click that topic, and select **Oxygen Editor > Insert as XRef**.

An <xref> element is added to your topic, with the @href attribute set to the ID of the topic you are linking to. For example:

```
<xref href="zif1556117500946.xml"/>
```

4. Set the @scope attribute to **peer**.
5. Add the title of the book as the link text.
6. If the link is going to a topic in another product, add a <data> element to your <xref> element.

If you are linking to a book in the same product, you can skip this step.

However, if you want to link from a book, for example, in the SR0620 release of Helix Virtualization Platform to a VxWorks 7 book, you must ensure the link goes to the SR0620 version of the documentation. Since topic IDs may not change from release to release, you

must specify which version of the product you want to link to. However, you can only have one **<othermeta>** element setting the facets to your product version; for example:

```
<othermeta content="version=os_helix_virtualization_platform_sr0620" name="facets"/>
```

Therefore, to ensure you are going to link to the proper version, you must override the version applied to the book in your **<xref>** by adding a **<data>** element.

- a) Inside the **<xref>** element, add a **<data>** element.
- b) Add a **@name** attribute to the **<data>** element and set the value to **facets**.
- c) Add a **@value** attribute to the **<data>** element and set the value to the Vocabulary Term Value for the product version.

This value is the term for the product you are linking to under the Vocabulary Term Value column in the Taxonomy Terms pane. For example, if you are linking to the SR0620 version of VxWorks 7, the value would be **os\_vxworks\_7\_sr0620**.

The value you place in the **<data>** element overrides the value in the **<othermeta>** element for this link only.

```
<xref href="zif1556117500946.xml" scope="peer"><data name="facets" value="version=os_vxworks_7_sr0620"/>VxWorks 7 Programmer's Guide</xref>
```

## Creating Links to Hierarchy Locations on the Product Documentation Site

You can create links to a category or hierarchy location on the Product Documentation site by including the category name in an **<xref>** element.

### Procedure

1. Add an **<xref>** element to your topic.
2. Define the **@href** attribute for the **<xref>** element.

The format is as follows:

```
<xref href="/category/catgoryName"/>
```

Where *categoryName* is the Vocabulary Term Value of the category in the Taxonomy Terms view. For example, the following screen capture shows the category name for the Connectivity category under the VxWorks 7 SR0600 product:

Readable	Vocabulary Term Value
VxWorks 7 SR0600	os_vxworks_7_sr0600
App Designer Toolkit	os_vxworks_7_sr0600_app_designer_toolkit
<b>Connectivity</b>	<b>os_vxworks_7_sr0600_connectivity</b>
Core	os_vxworks_7_sr0600_core

The @href for the above category is as follows:

```
<xref href="/category/os_vxworks_7_sr0600_connectivity"/>
```

3. Set the @scope attribute to **external**.
4. Set the @format attribute to **html**.
5. Add the page title as link text.

#### Example

The following example links to the Connectivity page under the VxWorks 7 SR0600 product:

```
<xref href="/category/os_vxworks_7_sr0600_connectivity" scope="external"
format="html">Connectivity</xref>
```

## Creating Links to Videos on the Product Documentation Site

You can create links to a video on the Product Documentation site by using the bundle name of the video.

**NOTE:** In the past, these links were created by using the content ID from the old Knowledge Library. If your documents contain this type of links, they must be updated to the current method.

Context for the current task.

#### Procedure

1. Find the bundle name of the video you want to link to:
  - a) On the Product Documentation site, click on your user name and select **Admin**.

- b) Click on the **Publications** tile.
- c) Do a search for your video, using the title.
- d) Copy the file name.

The file name is the same thing as the bundle name.

**Creating a VxWorks Downloadable Kernel Module — Video, 03:37**

*File name:* **Creating\_a\_VxWorks\_Downloadable\_Kernel\_Module\_Video\_0337\_1**

2. Add an **<xref>** element to your topic.
3. Define the **@href** attribute for the **<xref>**.

The syntax is as follows:

```
href="/bundle/bundleName"
```

Where *bundleName* is the bundle name you copied from the Product Documentation site.

4. Set the **@scope** attribute to **external**.
5. Set the **@format** attribute to **html**.
6. Type the video title as link text for your **<xref>**.

```
<xref href="/bundle/Creating_VxWorks_Downloadable_Kernel_Module_Video_0337_1"  
      scope="external" format="html">Creating a VxWorks Downloadable Kernel Module - Video,  
      03:37</xref>
```

# 25

## *Relationship Tables*

<a href="#">Relationship Tables Overview</a>	209
<a href="#">Options for Structuring Relationship Tables</a>	210
<a href="#">Models for Relationship Tables</a>	211
<a href="#">Opening the RelTable Editing Perspective</a>	211
<a href="#">Adding a Relationship Table to a Map</a>	213
<a href="#">Editing Relationship Tables</a>	214
<a href="#">Best Practices for Relationship Tables</a>	216

### **Relationship Tables Overview**

Relationship tables are special tables that you can add to a map to create links between topics. You can use them instead of embedded links to simplify topic reuse and to apply subject-based linking.

Each relationship table, or *reltable*, contains one or more rows that define links between topics. The cells in each row identify the topics to be linked. When output is generated from the map, the links appear automatically in a **Related Links** section at the end of each topic.

For example, the following simple reltable defines links between three topics:

```
<reltable>
  <relrow>
    <relcell><topicref href="Lending_libraries.xml"/></relcell>
    <relcell><topicref href="Borrowing_a_Book.xml"/></relcell>
    <relcell><topicref href="Libraries_in_Your_Area.xml"/></relcell>
  <relrow>
</reltable>
```

When the output is generated:

- *Lending Libraries* includes generated links to *Borrowing a Book* and *Libraries in Your Area*.

- *Borrowing a Book* includes generated links to *Lending Libraries* and *Libraries in Your Area*.
- *Libraries in Your Area* includes generated links to *Lending Libraries* and *Borrowing a Book*.

Relationship tables provide a method of linking that does not rely on `<xref>` links or `<related-links>` sections embedded within topics. This allows topics to be reused easily in any context; when you add a topic to a map, you do not have to worry that it may contain links not resolved in the map.

## Options for Structuring Relationship Tables

Relationship tables offer a flexible structure to meet different requirements. You can specify one-way or two-way links, create links between groups of topics, and add columns as needed to develop complex networks of related links.

Each row in a relationship table can have any number of columns. Instead of the three-column example above, you could create a relationship table with two columns or four columns.

Each cell can include any number of topics. This allows you to relate a single topic in one column to multiple topics in the second column, as in the following example:

```
<reltable>
  <relrow>
    <relcell><topicref href="Lending_libraries.xml"/></relcell>
    <relcell>
      <topicref href="Finding_a_Book.xml"/>
      <topicref href="Borrowing_a_Book.xml"/>
      <topicref href="Returning_a_Book.xml"/>
    </relcell>
  <relrow>
</reltable>
```

In the output, the topic *Lending Libraries* includes three generated links. Each of the other topics contains one generated link back to *Lending Libraries*.

 **NOTE:** The topics in the second column are *not* linked to each other. In theory, you can add a `@collection-type` attribute on the `<relcell>` element and set it to **family** to create links between topics in a cell. However, in many production environments, including the one at Wind River, the Ixiasoft CCMS output generator does not generate the links.

You can control the direction of the links. By default, the links are bidirectional; topics in the first column are linked to those in the second, and topics in the second are linked to those in the first. You can change this for a topic by adding a `@linking` attribute. If you set the attribute to **targetonly**, links are generated to the topic, but not from it. If you set it to **sourceonly**, links are generated from the topic, but links to it are not generated. You can apply the `@linking` attribute to individual `<topicref>` elements, or you can apply it to the `<relcell>` element so that all the topics in the cell inherit it.

## Models for Relationship Tables

There are several common models for relationship tables. Each model has its own advantages.

You can apply multiple models by adding more than one reltable to a map. For example, you could add a three-column model for subject-based linking, and then add a two-column model with bidirectional linking so that you can create links between concepts or tasks.

### Two-Column Unidirectional Tables

This model is simple to understand and manage, but requires additional effort when setting up bidirectional links.

One column is used for **sourceonly** links, and the other is used for **targetonly** links.

### Three-Column Task/Concept/Reference (CTR) Tables

This model supports subject-based linking, where links are automatically generated for a set of topics about a particular subject.

One column is used for concepts, a second is used for tasks, and a third is used for reference topics. You can enforce the type of topic accepted in each cell by setting the **@type** attribute for the cell (for example, to **concept-wr**).

## Opening the RelTable Editing Perspective

You can use the RelTable Editing perspective included with the Ixiasoft CCMS to work with relationship tables in an easy-to-use graphical environment.

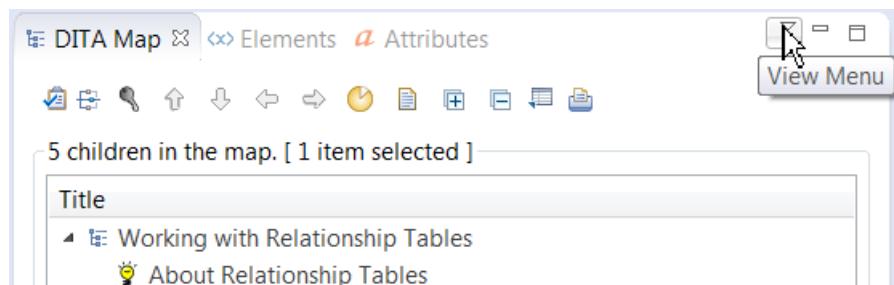
### Procedure

1. Open the associated map in DITA Map view.

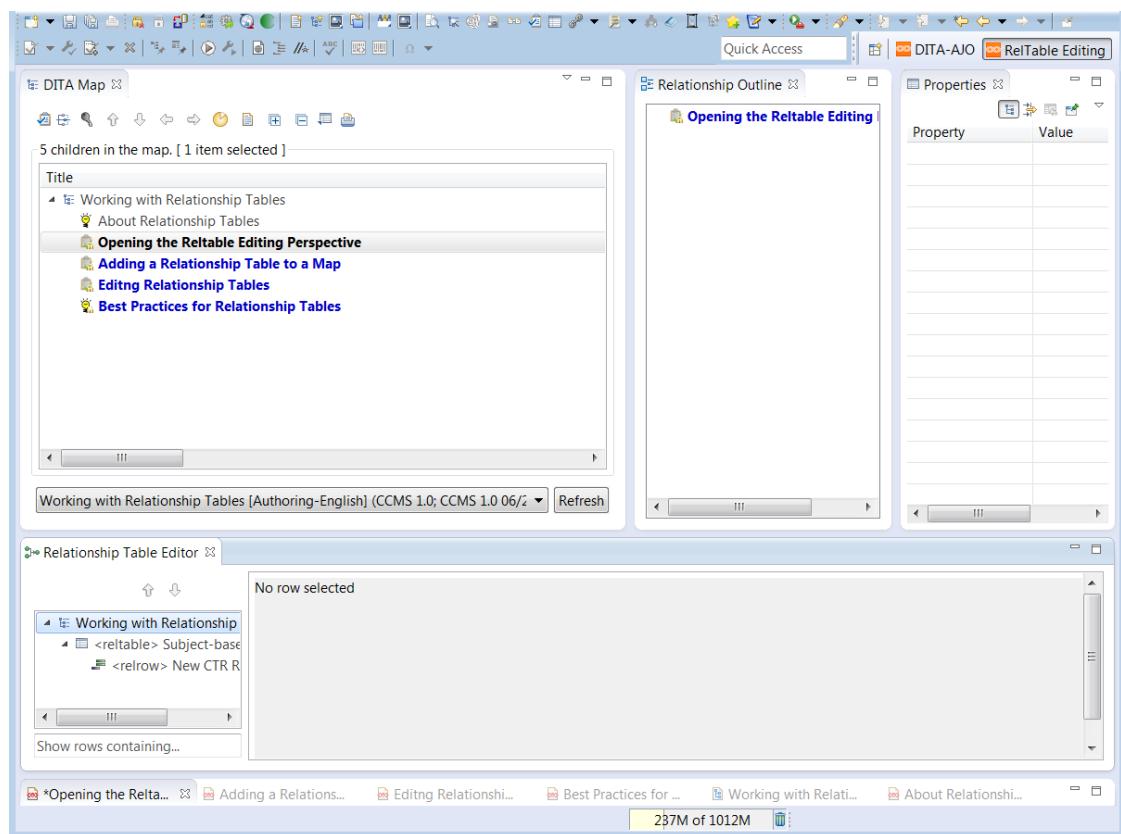
Relationship tables are contained in maps. To create or edit a reltable, you must open the associated map.

2. In the View Menu for the map, select **Open Reltable Editing Perspective**.

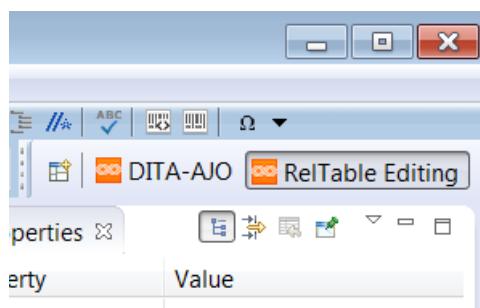
The View Menu is accessible in the upper right corner of the DITA Map View.



The RelTable Editing perspective appears. By default it includes the Relationship Table Editor view, the Relationship Outline view, the DITA Map view, and the Properties view.



In addition, a **Reltable Editing** button is added to the perspective controls. You can use these controls to change perspectives.



## Adding a Relationship Table to a Map

You can add several different types of relationship tables to a map to add links between your topics.

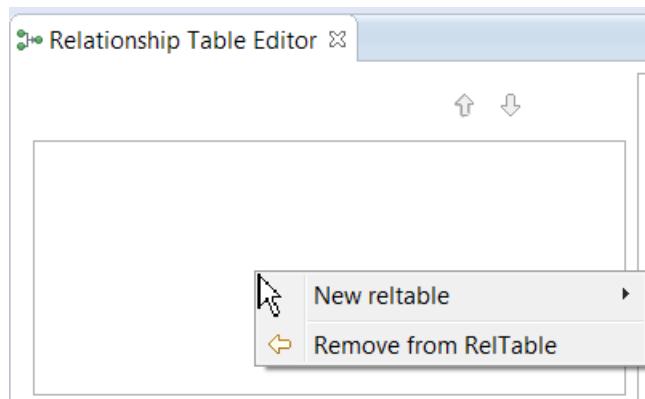
### Prerequisites

- The map must be locked.
- This task assumes that you have the RelTable Editing perspective open.

### Procedure

1. Right-click the white space in the list of relationship tables, and then select **New reltable**.

The list of relationship tables is in the **Relationship Table Editor** view of the Reltable Editing perspective.

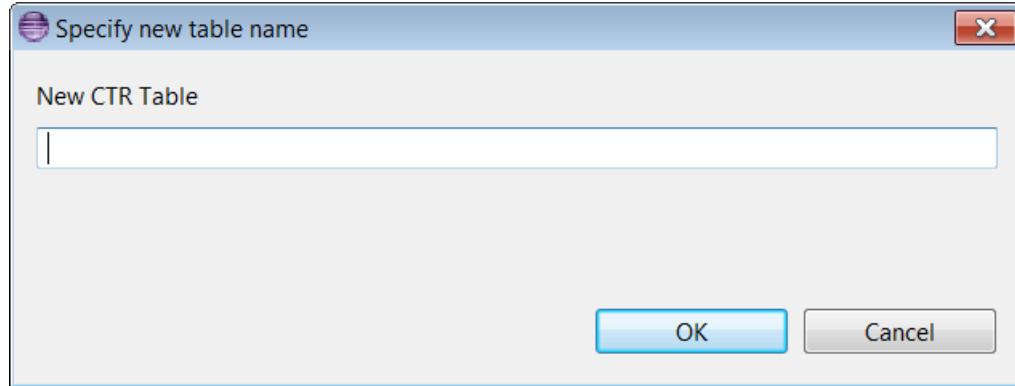


2. Select the type of reltable to add.

Table Type	Description
<b>CTR Table</b>	Adds a three-column table to your map. Each column supports a specific type of topic: Concept, Task, and Reference.
<b>Sequence Table</b>	Adds a one-column table to your map. You can then nest topics and use the <b>@collection-type</b> attribute to specify the relationship between the topics; for example, sequence or family.

Table Type	Description
<b>Source-Target Table</b>	Adds a two-column table to your map. Topics in the first (source) column link to topics in the second column (target), but not vice versa. That is, topics in the target column do not link back to the topics in the source column.
<b>Two-column Table</b>	Adds a two-column table to your map. The linking in this table is bidirectional, unless you specify otherwise. That is topics in the first column link to topics in the second column, and topics in the second column link back to topics in the first column.

The Specify new table name dialog box appears.



3. Provide a name for the reltable, and click **OK**.

The reltable is inserted in the map, and an entry for it is added to the list of relationship tables in the Relationship Table Editor view. The new reltable includes a single empty row.

## Editing Relationship Tables

You can use the Relationship Table Editor view to add rows and cells to a reltable, and to populate them with topics to be linked.

### Prerequisites

The map must be locked.

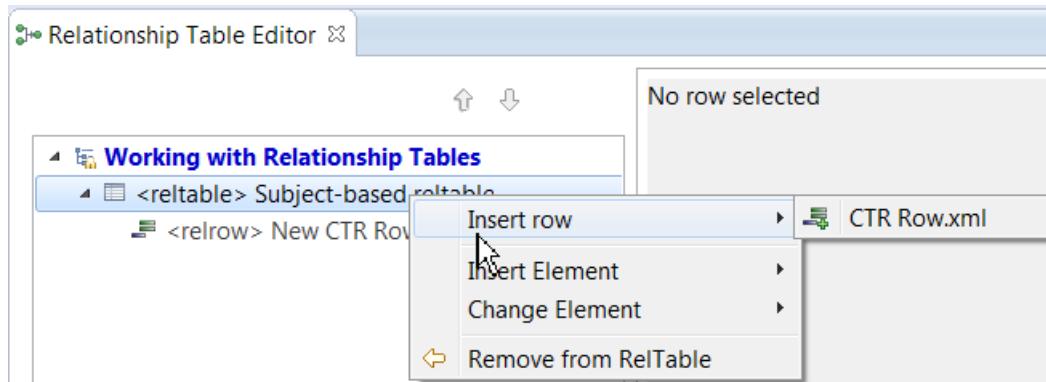
### Procedure

1. If required, add a row to the reltable.

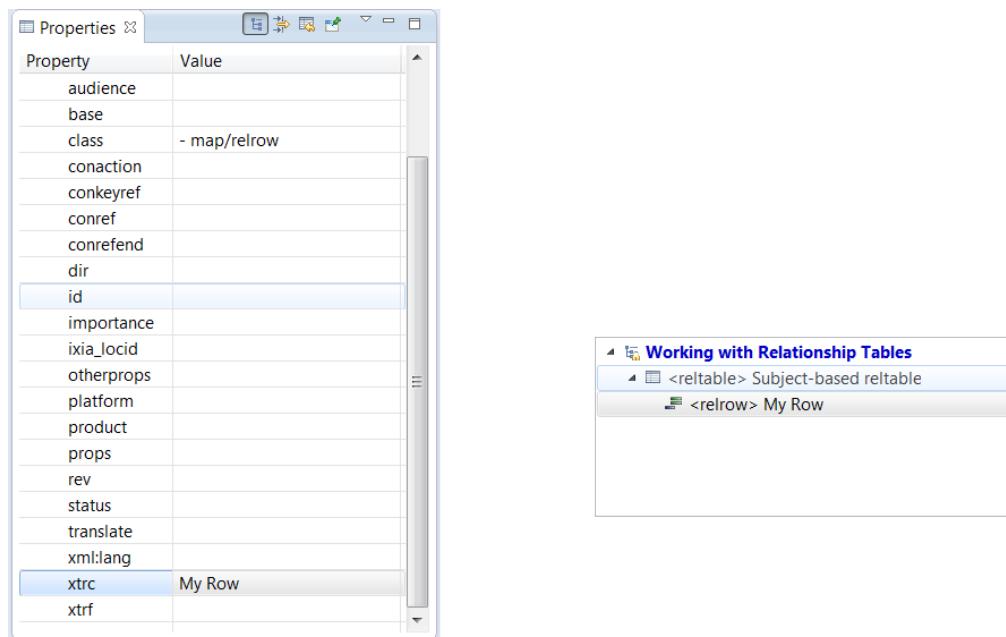


**NOTE:** A new reltable has one empty row by default. Use this row before adding others.

- Right-click the reltable in the Relationship Table Editor view, and then select **Insert Row** from the context menu.

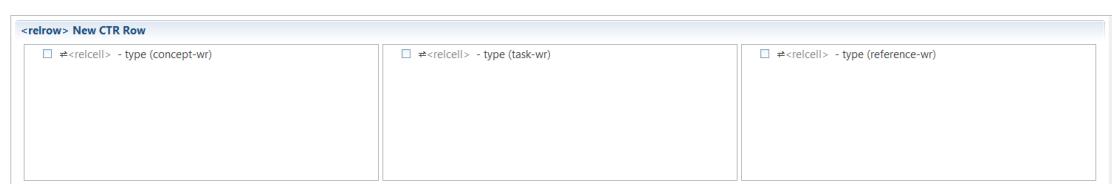


- Set the @xtrc attribute on each row to identify it in the Relationship Table Editor view.



- Select the row to edit.

The cells in the row are displayed.



- Drag the topics you want to link from the DITA Map view into the appropriate cells.

## Best Practices for Relationship Tables

The recommended best practices ensure that your reltables are easy to manage and generate useful output.

- Use a reltable for each map.

Smaller reltables are easier to manage. Create one for each submap. If the map contains other maps, create links between topics in the lowest-level map that contains both topics.

- Do not create parent-child links. These are already generated automatically in HTML output.
- If your topics use conditions (for example, @product, @platform, or @audience attributes), ensure that the condition attributes are included when you add the <topicref> to the reltable.



**CAUTION:** If the topic inherits a condition from a parent element, the condition for the parent element is *not* automatically added when the <topicref> is dragged to the reltable.

- Manage the reltable whenever you make changes to the map.

If you delete a topic from a map, any corresponding <topicref> is *not* automatically deleted from the reltable. You must delete it manually.

# 26

## Tables

<a href="#">General Guidelines</a>	217
<a href="#">Types of Tables</a>	218
<a href="#">Creating Simple Tables with the Insert Table Dialog Box</a>	221
<a href="#">Creating Complex Tables with the Insert Table Dialog Box</a>	222
<a href="#">Creating Choice Tables with the Insert Choice Table Dialog Box</a>	223
<a href="#">Resizing Columns in Simple Tables, Choice Tables, or Property Tables</a>	224
<a href="#">Resizing Columns in Complex Tables</a>	225
<a href="#">Spanning Columns in a Complex Table</a>	225
<a href="#">Spanning Rows in a Complex Table</a>	227
<a href="#">Splitting Cells in a Spanned Row or Column</a>	228
<a href="#">The Table Toolbar Reference</a>	229

### General Guidelines

You should follow some general guidelines when creating tables.

- Avoid creating one-column tables. Use the `<sl>` or `<choices>` elements instead.
- Consider whether your information is best suited for a table or a definition list (`<dl>`). Be consistent with table/definition list usage.
- Avoid creating extremely wide tables. A good guideline is to avoid creating tables with more than four columns. They are usually too wide to display properly in PDF.

## Types of Tables

### Simple Tables

You use a simple table (`<simpletable>`) to describe tabular information that has a simple layout.

The simple table is less sophisticated than the [complex table](#):

- You cannot control the format of the table.
- You cannot span rows or columns.
- A simple table does not have a title.

You can create a simple table manually or with the table wizard. You can add simple tables in all topic types.

#### Example of a Simple Table

```
<simpletable>
  <sthead>
    <stentry><p>Format</p></stentry>
    <stentry><p>FAT Table Entry Size</p></stentry>
    <stentry><p>Usage</p></stentry>
    <stentry><p>Size</p></stentry>
  </sthead>
  <strow>
    <stentry><p>FAT12</p></stentry>
    <stentry><p>12 bits per cluster number</p></stentry>
    <stentry><p>Appropriate for very small devices with up to 4,084 KB
clusters.</p></stentry>
    <stentry><p>Typically, each cluster is two sectors large.</p></stentry>
  </strow>
  <strow>
    <stentry><p>FAT16</p></stentry>
    <stentry><p>16 bits per cluster number</p></stentry>
    <stentry><p>Appropriate for small disks of up to 65,524 KB clusters.</p></
stentry>
    <stentry><p>Typically, used for volumes up to 2 GB; can support up to 8 GB.</p></
stentry>
  </strow>
</simpletable>
```

### Complex Tables

You use a complex table (`<table>`) to describe complex tabular information.

With complex tables, you can:

- span rows and columns
- add a table title
- control the display properties and layout

You can create complex tables manually or with the table wizard. You can add complex tables in all topic types.

## Example of a Complex Table

```
<table>
  <tgroup cols="2">
    <colspec colname="c1" colnum="1"/>
    <colspec colname="c2" colnum="2"/>
    <thead>
      <row>
        <entry><p>Animal</p></entry>
        <entry><p>Gestation</p></entry>
      </row>
    </thead>
    <tbody>
      <row>
        <entry><p>Elephant</p></entry>
        <entry><p>19-22 months</p></entry>
      </row>
      <row>
        <entry><p>Giraffe</p></entry>
        <entry><p>15 months</p></entry>
      </row>
    </tbody>
  </tgroup>
</table>
```

## Related Links

[Creating Complex Tables with the Insert Table Dialog Box](#) on page 222

The Insert Table dialog box allows you to quickly create and configure complex tables using a GUI.

[Resizing Columns in Complex Tables](#) on page 225

You can use the `@colwidth` attribute to resize columns in complex tables. The quickest and easiest way to do this is using the GUI in the oXygen XML Author editor.

[Spanning Columns in a Complex Table](#) on page 225

You can join cells of a complex table so that one entry spans across two or more columns of the table.

[Spanning Rows in a Complex Table](#) on page 227

You can join cells of a complex table so that one entry spans across two or more rows of the table.

[Splitting Cells in a Spanned Row or Column](#) on page 228

You can easily split cells in a spanned row or column by using the options in the **Join or split tables** drop-down list on the table toolbar

## Choice Tables

You use a choice table (`<choicetable>`) to include a two-part list of choices for a specific step in a task.

Choice tables are only allowed in `<step>` elements in task topics. They are not allowed in substeps.

You can create choice tables manually or with the table wizard.



**NOTE:** The option for creating a choice table with the table wizard is only available when your cursor is placed in `<step>` element where choice tables are allowed.

If you do not add a header row, one is automatically added for you in the transform with the headings **Option** and **Description**.

### Example of a Choice Table

```
<step>
  <cmd>Specify the BSP for your host OS:</cmd>
  <choicetable id="choicetable_e2d1b135-5e4a-444e-b771-2213a0fc58ce">
    <chhead>
      <choptionhd>Host Operating System</choptionhd>
      <chdeschd>Value</chdeschd>
    </chhead>
    <chrow>
      <choption>Windows</choption>
      <chdesc>simpC</chdesc>
    </chrow>
    <chrow>
      <choption>Linux</choption>
      <chdesc>linux</chdesc>
    </chrow>
    <chrow>
      <choption>Solaris</choption>
      <chdesc>solaris</chdesc>
    </chrow>
  </choicetable>
</step>
```

### Related Links

[Creating Choice Tables with the Insert Choice Table Dialog Box](#) on page 223

The Insert Table dialog box allows you to quickly create and configure choice tables using a GUI.

[Resizing Columns in Simple Tables, Choice Tables, or Property Tables](#) on page 224

You can use the `@relcolwidth` attribute to resize columns in simple tables, choice tables, or property tables.

## Properties Tables

You use properties tables (`<properties>`) in reference topics to inform users about terms, values, and descriptions.

Properties tables are only allowed in reference topics at the `<refbody>` level. They cannot be inserted in `<section>` or other elements.

Properties tables are limited to three columns. If you need fewer columns or more columns, you must use either a simple table or a complex table.

Properties tables must be created manually. They cannot be created using the table wizard.

If you do not include a header row in your table, one is automatically added for you in the transform with the headings **Type**, **Value**, and **Description**.

### Example of a Properties Table

```

<properties>
  <prophead>
    <proptypehd>Configuration Option</proptypehd>
    <propvaluehd>Default Value</propvaluehd>
    <propdeschd>Description</propdeschd>
  </prophead>
  <property>
    <proptype><uicontrol>Backplane Name</uicontrol></proptype>
    <propvalue><b>publicWMB</b></propvalue>
    <propdesc><p>Type an appropriate name for the backplane.</p></propdesc>
  </property>
  <property>
    <proptype><uicontrol>Initial Max Resources</uicontrol></proptype>
    <propvalue>1000</propvalue>
    <propdesc><p>Set the initial limit of WindMark resources that can register with
the
backplane instantiated in this project.</p></propdesc>
  </property>
</properties>

```

### Related Links

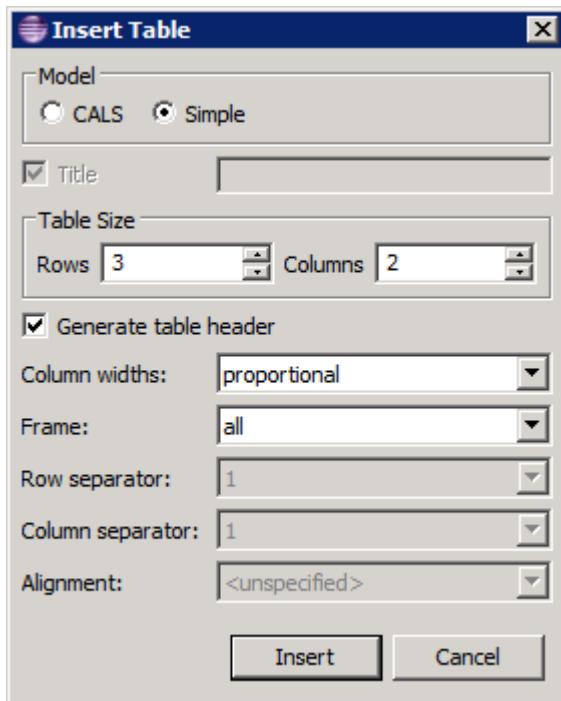
[Resizing Columns in Simple Tables, Choice Tables, or Property Tables](#) on page 224

You can use the **@relcolwidth** attribute to resize columns in simple tables, choice tables, or property tables.

## Creating Simple Tables with the Insert Table Dialog Box

The Insert Table dialog box allows you to quickly create and configure simple tables using a GUI.

**Figure 1: The Insert Table Dialog Box**



#### Procedure

1. Insert the cursor in your topic where you would like to add the table.

2. Click the Insert a table icon.

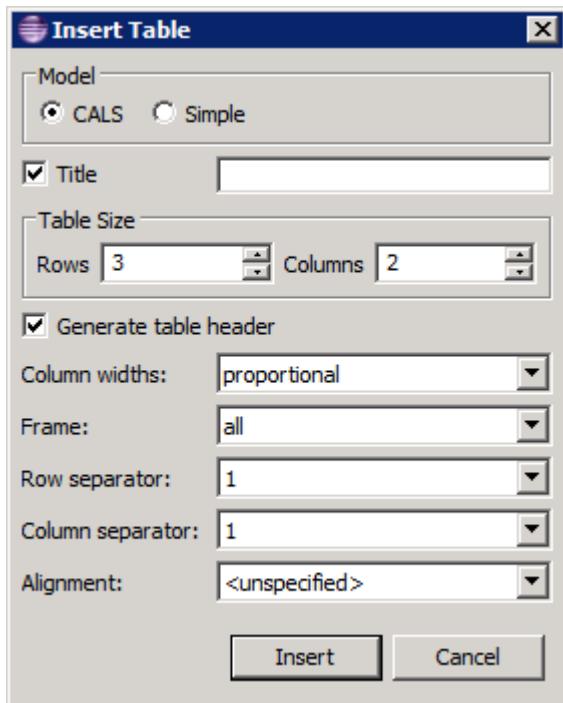


3. In the Model section of the Insert Table wizard, select **Simple**.
4. In the Table Size section, select or type the desired number of rows and columns.
5. If you want a table header row, select **Generate table header**.  
If you do not want a table header row, clear the selection.
6. Accept all the remaining default settings, and click **Insert**.

## Creating Complex Tables with the Insert Table Dialog Box

The Insert Table dialog box allows you to quickly create and configure complex tables using a GUI.

**Figure 2: The Insert Table Dialog Box**



#### Procedure

1. Insert the cursor in your topic where you would like to add the table.

2. Click the Insert a table icon.

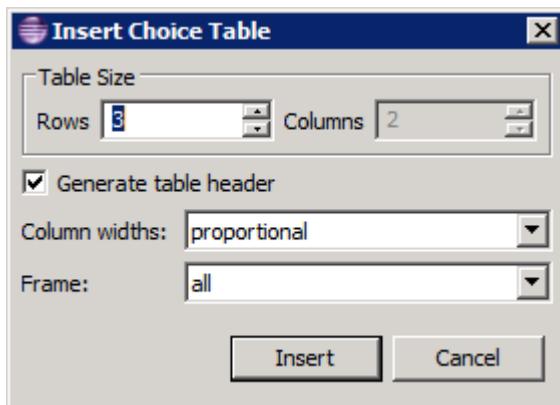


3. In the Model section of the Insert Table wizard, select **CALS**.
4. If you want a table title, select **Title** and type an appropriate title for the table.  
If you do not want a table title, clear the selection.
5. In the Table Size section, select or type the desired number of rows and columns.
6. If you want a table header row, select **Generate table header**.  
If you do not want a table header row, clear the selection.
7. Accept all the remaining default settings, and click **Insert**.

## Creating Choice Tables with the Insert Choice Table Dialog Box

The Insert Table dialog box allows you to quickly create and configure choice tables using a GUI.

**Figure 3: The Insert Choice Table Dialog Box**



### Procedure

1. Insert the cursor in your topic where you would like to add the table.



**NOTE:** You can only add choice tables in `<step>` elements. The **Insert Table** icon is grayed out if your cursor is in an incorrect position in the topic.

2. Click the Insert a table icon.



3. In the Table Size section, select the desired number of rows.



**NOTE:** By default, the number of columns is set to 2. You cannot change this default for choice tables.

4. If you want a table header row, select **Generate table header**.  
If you do not want a table header row, clear the selection.
5. Accept all the remaining default settings, and click **Insert**.

## Resizing Columns in Simple Tables, Choice Tables, or Property Tables

You can use the **@relcolwidth** attribute to resize columns in simple tables, choice tables, or property tables.



**NOTE:** The **@relcolwidth** attribute is not available for complex tables. For information on resizing columns in complex tables, see [Resizing Columns in Complex Tables](#) on page 225.

### Procedure

1. Insert your cursor in the root table element for your table.  
For example:
  - <simpletable>
  - <choicetable>
  - <properties>
2. In the Attributes view, enter a relative value in the **@relcolwidth** attribute to specify the width of a column in relationship to the width of the other columns.



**NOTE:** Each value must be followed with an asterisk (\*).

The values are totaled and made a percent.

For example, the following results in widths of 16.7%, 33.3%, and 50.1%:

```
<simpletable relcolwidth="1* 2* 3*">
```

The following results in widths of 20% and 80%:

```
<simpletable relcolwidth="20* 80*">
```

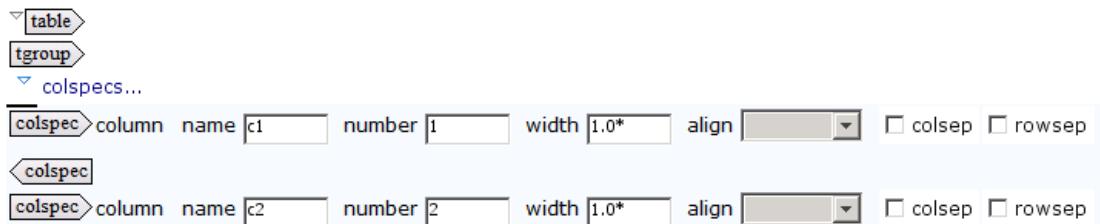
## Resizing Columns in Complex Tables

You can use the **@colwidth** attribute to resize columns in complex tables. The quickest and easiest way to do this is using the GUI in the oXygen XML Author editor.

**NOTE:** The **@colwidth** attribute is only available for complex tables. For information on resizing columns in simple tables, choice tables, or property tables, see [Resizing Columns in Simple Tables, Choice Tables, or Property Tables](#) on page 224.

### Procedure

1. In the oXygen XML Author editor in Author mode, expand **colspeсs** under the **<tablegroup>** element.



2. For each column in the table, type a relative value in the **width** text box to specify the width of a column in relationship to the width of the other columns.

**NOTE:** Each value must be followed with an asterisk (\*).

The values are entered in the **@colwidth** attribute of the **<colspec>** element. The values are totaled and made a percent.

For example, the following results in widths of 16.7%, 33.3%, and 50.1%:

```
<colspec colwidth="1* 2* 3*">
```

The following results in widths of 20% and 80%:

```
<colspec colwidth="20* 80*">
```

## Spanning Columns in a Complex Table

You can join cells of a complex table so that one entry spans across two or more columns of the table.

The easiest way to span columns is by using the options in the **Join or split tables** drop-down list on the table toolbar.



**NOTE:** You cannot span columns or rows in simple tables, choice tables, or property tables.

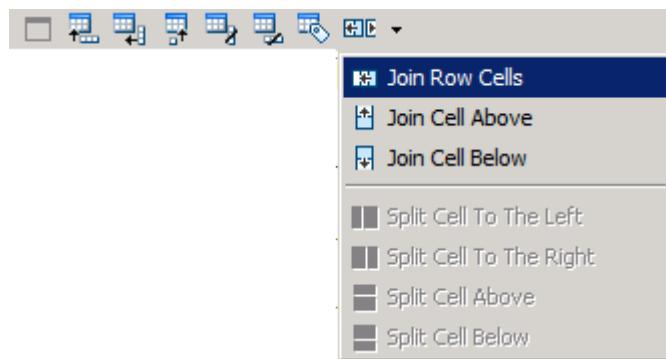
### Procedure

1. In Author mode, select the desired cells.

Type of Table	Method
<b>Two-column table</b>	Insert your cursor in the <entry> element in one of the columns for the row.
<b>Tables with more than two columns</b>	Select the cells for the columns you want to span.

[entry] p Heading 1 p [entry]	[entry] p Heading 2 p [entry]	[entry] p Heading 3 p [entry]
[entry] p This p [entry]	[entry] p That p [entry]	[entry] p The other thing p [entry]
[entry] p More text p [entry]	[entry] p Even more text p [entry]	[entry] p Another entry p [entry]

2. Click the **Join or split cells** icon on the table toolbar, and select **Join Row Cells** from the drop-down menu.



The cells are joined.

If there is content in both cells, the content from both cells is included in the <entry> element of the newly joined cell.

[entry] p Heading 1 p [entry]	[entry] p Heading 2 p [entry]	[entry] p Heading 3 p [entry]
[entry] p This p [entry]	[entry] p That p [entry]	[entry] p The other thing p [entry]
[entry] p More text p [entry]	[entry] p Even more text p [entry] [entry] p Another entry p [entry]	

## Spanning Rows in a Complex Table

You can join cells of a complex table so that one entry spans across two or more rows of the table.

The easiest way to span rows is by using the options in the **Join or split tables** drop-down list on the table toolbar.



**NOTE:** You cannot span columns or rows in simple tables, choice tables, or property tables.

### Procedure

1. In Author mode, insert your cursor in the <entry> element in one of the desired cells.



**NOTE:** If you select more than one cell and try to join them, the results are not always what you expect.

2. Click the **Join or split cells** icon on the table toolbar, and select one of the following options from the drop-down list:

Option	Description
<b>Join Cell Above</b>	Joins the current cell with the cell above it.
<b>Join Cell Below</b>	Joins the current cell with the cell below it.

The cells are joined.

If there is content in both cells, the content from each cell is included in the <entry> element of the newly joined cell.

<entry> p Category <p>	<entry> p Item <p>	<entry> p Description <p>
<entry> p These things: <p>	<entry> p This <p>	<entry> p Description of this. <p>
		<entry> p <p>
	<entry> p That <p>	<entry> p Description of that. <p>
		<entry> p <p>

### Postrequisites

If you want to span more than two rows, you must repeat the procedure for those additional rows.

## Splitting Cells in a Spanned Row or Column

You can easily split cells in a spanned row or column by using the options in the **Join or split tables** drop-down list on the table toolbar

 **NOTE:** You can only split cells that were previously joined to span a row or column. You cannot use this feature to create a split cell in place of a existing single cell.

### Procedure

1. Insert your cursor in the spanned cell.
2. Click the **Join or split cells** icon on the table toolbar, and select one of the following options from the drop-down list:

Option	Description
<b>Split Cell To The Left</b>	Used for spanned columns.  The single, spanned cell is split into two cells. If more than two columns are spanned, you will have a single cell on the left and a spanned cell on the right. The content from the original cell appears in the right column.
<b>Split Cell To The Right</b>	Used for spanned columns.  The single, spanned cell is split into two cells. If more than two columns are spanned, you will have a single cell on the right and a spanned cell on the left. The content from the original cell appears in the left column.
<b>Split Cell Above</b>	Used for spanned rows.  The single, spanned cell is split into two cells. If more than two rows are spanned, you will have a single cell on the top and a spanned cell on the bottom. The content from the original cell appears in the bottom cell.
<b>Split Cell Below</b>	Used for spanned rows.  The single, spanned cell is split into two cells. If more than two rows are spanned, you will have a single cell on the bottom and a spanned cell on the top. The content from the original cell appears in the top cell.

## Postrequisites

If more than two rows or columns are spanned, you must repeat this procedure to split all the cells in the spanned row or column.

## The Table Toolbar Reference

The oXygen XML Author editor provides a table toolbar that allows you to quickly insert new tables or modify existing tables, without having to add each DITA element separately.

**Figure 4: The Table Toolbar**



Icon	Description
	<p>Insert a table.</p> <p>You can insert the following types of tables:</p> <ul style="list-style-type: none"><li>• complex tables</li><li>• simple tables</li><li>• choice tables</li></ul> <p><b>NOTE:</b> This option is only available if your cursor is placed inside a <b>step</b> element.</p>
	For more information, see: <ul style="list-style-type: none"><li>• <a href="#">Creating Complex Tables with the Insert Table Dialog Box</a> on page 222</li><li>• <a href="#">Creating Simple Tables with the Insert Table Dialog Box</a> on page 221</li><li>• <a href="#">Creating Choice Tables with the Insert Choice Table Dialog Box</a> on page 223</li></ul>
	Insert a new table row below the current row.
	Insert a new table column after the current column.
	Insert a table cell (<entry>).
	Delete the current table column.
	Delete the current table row.
	Customize the display properties. <p>The Table Properties dialog box provides separate tabs, allowing you to customize the display properties for the entire table or the selected rows, columns, or cells.</p>

Icon	Description
	<p>Join or split table cells.</p> <p>The drop-down list provides the following options:</p> <ul style="list-style-type: none"><li>• <b>Join Row Cells</b></li><li>• <b>Join Cell Above</b></li><li>• <b>Join Cell Below</b></li><li>• <b>Split Cell To The Left</b></li><li>• <b>Split Cell To The Right</b></li><li>• <b>Split Cell Above</b></li><li>• <b>Split Cell Below</b></li></ul> <p>For more information, see:</p> <ul style="list-style-type: none"><li>• <a href="#">Spanning Columns in a Complex Table</a> on page 225</li><li>• <a href="#">Spanning Rows in a Complex Table</a> on page 227</li><li>• <a href="#">Splitting Cells in a Spanned Row or Column</a> on page 228</li></ul>

- [\*\*About Indexes\*\* 231](#)
- [\*\*Adding an Index\*\* 231](#)
- [\*\*Nesting Index Elements\*\* 232](#)
- [\*\*Additional Index Elements\*\* 233](#)

## About Indexes

Use `<indexlist>` to cause your bookmap to generate an index and `<indexterm>` in the topic prolog to create each entry.

When you generate output, the transform creates an index. The index appears as an appendix in the PDF and on the index tab in the Eclipse Help pane for HTML.

## Adding an Index

Index terms are defined in the `<prolog>` element of each topic file that you want to have referenced.

Use the following steps to add an index to your DITA documents.

## Procedure

1. Create an empty `<indexlist>` element in your bookmap in the **backmatter** section.

```
<backmatter>
    <booklists>
        <indexlist/>
    </booklists>
</backmatter>
```

2. In each topic file where you want index terms to be referenced, define the index terms in the `<prolog>` element with the `<indexterm>` element.

```
<prolog>
    <metadata>
        <keywords>
            <indexterm>costs</indexterm>
        </keywords>
    </metadata>
</prolog>
```

 **NOTE:** The `<author>` element is also inside the `<prolog>` element. For clarity, it is best to place the `<author>` element immediately after the `<prolog>` element, then place the `<metadata>` element after that.

The index is automatically created when you run the transform for the book. A link to the index displays in the list of contents. When a customer clicks a term in the index, they are taken to the top of the linked topic.

The [Wind River Technical Publications House Style Guide](#) contains some helpful indexing guidelines.

 **NOTE:** The guide is best viewed in PDF format, as some of the topics are deeply nested and "hidden" in the HTML version.

## Nesting Index Elements

You can nest `<indexterm>` elements to create secondary and tertiary terms.

## Procedure

To nest index terms, add additional `<indexterm>` elements within the parent `<indexterm>` element.

For example:

```
<indexterm>configuring
    <indexterm>using command line</indexterm>
    <indexterm>using Workbench</indexterm>
</indexterm>
```

Avoid going beyond two levels where possible.

## Additional Index Elements

You can use the `<index-see>`, `<index-see-also>`, and `<index-sort-as>` elements to cross-reference similar terms or use alternative spellings and references without the need for duplication or unnecessarily long indexes.

### `<index-see>`

Use this element to redirect readers to a term that they should consult instead of the one they have looked up.

### `<index-see-also>`

Use this element to redirect readers to a term that they should consult in addition to the term that they have looked up.

### `<index-sort-as>`

Use this element to specify the sort order of an index term in the index list.

For example:

```
<prolog>
  <keywords>
    <indexterm>costs
      <indexterm>estimating</indexterm>
      <indexterm>tracking
        <index-see-also>monitoring</index-see-also>
      </indexterm>
    </indexterm>
    <indexterm>expenses
      <index-see>costs</index-see>
    </indexterm>
    <indexterm>#variable
      <index-sort-as>variable</index-sort-as>
    </indexterm>
  </keywords>
</prolog>
```



# 28

## *Reusing Content*

[Content Reuse](#) 235

[Including a Topic in Multiple Books](#) 236

[Including a Topic Multiple Times in the Same Book](#) 236

[Reusing Elements from Referable-Content Topics](#) 237

[Conditional Content](#) 240

[Keyrefs/Variables](#) 251

### **Content Reuse**

DITA provides multiple ways to create content once and reuse it many times. You must determine which reuse strategy is most appropriate for your content.

A well-written topic provides users with accurate, minimal information that focuses on the user's goal. Once you have well-written DITA content, you can easily reuse it from one product release to the next. DITA content can be reused across different products as well, and can be used to single-source slightly different versions of the same information. For example, the compiler produces similar guides for different target architectures. Rather than maintaining ten different guides, Wind River uses one information set to generate ten different guides.

Content reuse does not happen magically. It takes a certain amount of analysis and effort. Look for reuse opportunities before you create or clone new topics. Before writing a new topic in the repository, search for duplicate or near duplicate content. Near-duplicate content can be cleaned up and refactored to make it reusable. Before cloning topics, consider other avenues of reuse such as storing variable text in keys or conditionalizing the content.

## Including a Topic in Multiple Books

The simplest method of reuse is to include the same content in multiple maps without customizing the content.

**NOTE:** You can reuse the same topic in multiple books, but you cannot include a topic multiple times in the same bookmap. If you need to include a topic several times in the same bookmap, create a referable-content topic and create conrefs to the referable-content topic.

For more information, see [Including a Topic Multiple Times in the Same Book](#) on page 236.

### Procedure

To reuse a topic, simply create a <topicref> to the topic in the appropriate maps.

## Including a Topic Multiple Times in the Same Book

If you need to include a topic several times in the same bookmap, you must create a referable-content topic that contains a topic.

**NOTE:** If you want to include the same topic in multiple books, see [Including a Topic in Multiple Books](#) on page 236.

### Procedure

1. Create a referable-content topic that includes the entire topic you want to reuse.

For example, if you want to include a task topic, your referable-content topic would look similar to the following:

```
<referable-content>
  <title>[RC:task] Assigning a New Host Node</title>
  <rcbody>
    <task-wr>
      <title>Assigning a New Host Node</title>
      <shortdesc>You can use controller-0 to assign a configuration or personality (controller, compute, or storage, for example) to a new node.</shortdesc>
      <prolog>
        <author/>
      </prolog>
      <taskbody>
        <context/>
        <steps>
          <step>
            <cmd>Add the host to the system inventory.</cmd>
            <info>
              <note>
                <p>The host must be added to the system inventory before it is powered on.</p>
              </note>
            </info>
          </step>
          <step>
            <cmd>Power on the host.</cmd>
          </step>
          <step>
            <cmd>Assign a personality.</cmd>
          </step>
          <step>
            <cmd>Obtain the host ID.</cmd>
          </step>
        </steps>
      </taskbody>
    </task-wr>
  </rcbody>
</referable-content>
```

2. Create a separate topic for each place you want the content to appear in your book.  
For example, if you want a task topic to appear in two different chapters of your book, you would create two separate task topics.
3. For each of the topics you created in Step 2, create a conref to the referable-content topic.  
For more information, see [Reusing Elements from Referable-Content Topics](#) on page 237.

## Reusing Elements from Referable-Content Topics

You can pull in content defined in a referable-content topic into another topic by adding a content reference, or *conref*, to the topic. You can reuse individual elements in the referable-content topic or you can reuse the entire topic.



**CAUTION:** Only conref content stored within referable-content topics. If you create conrefs between regular topics, you risk creating complex and unintended file dependencies.

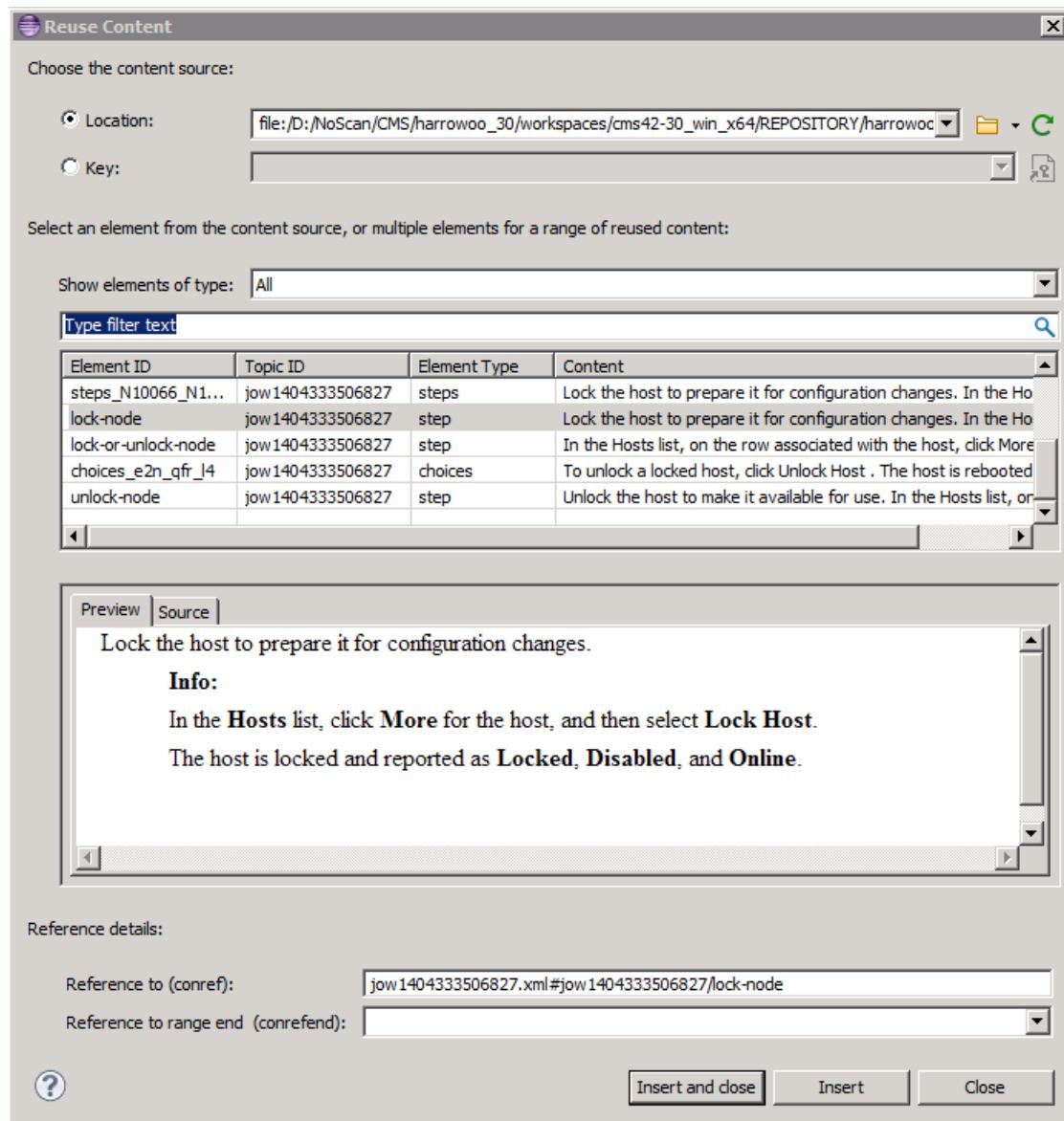
Content references are updated in real time. If you reference content and the content changes, your topic updates automatically.

When you reference content, consider the risk of the content changing. For example, if someone else updates the referable content topic before you generate output, the reference in your topic uses the different, updated content without any notification.

#### **Procedure**

1. Locate your referable-content topic in the **Search Results** view.
2. Open the referable-content topic and note its ID.
3. Open the topic where you want to add the reusable content in Author view and place your cursor where you want to pull in the content.  
Your cursor must be in a valid position for the type of element you want to insert. For example, you cannot add a reusable table element into a short description, because tables are not allowed within short descriptions.
4. Select **DITA > Reuse Content**.

The Reuse Content dialog box appears.



5. Select **Location** and then select the referable-content topic by its ID from the drop-down list.

**NOTE:** It is easier to select the correct topic if you only have the referable-content topic and the topic where you want to add the reusable content in open in Author view.

6. Click on the element content you want to insert in the dialog to preview and confirm that you selected the appropriate content.

7. Insert the content by doing one of the following:

Option	Method
<b>Insert only one element from the reusable-content topic.</b>	Click <b>Insert and close</b> .
<b>Insert multiple elements from the reusable-content topic.</b>	Click <b>Insert</b> and repeat Steps 6 and 7 until you have added all content. Then, click <b>Close</b> .



**NOTE:** The Ixiasoft CCMS allows you to select multiple elements at a time (**Control + Click**). When you do this, you create a conref range. It allows you to add multiple elements at the same time without creating a series of small content references in your topic. However, creating a conref range should be done with caution. If content is added to the referable-content topic between the start of your range and the end of your range, it is automatically added to your topic. There is no indication in the referable-content topic that tells you the content is being used in a range.

8. If a different writer is responsible for the conref source file, inform that writer that you are reusing their content and ask about the volatility of the content during your project's time line. The Ixiasoft CCMS does not notify you when content is updated, therefore communication must be from writer to writer.

## Conditional Content

### Conditional Content Overview

You can use DITA conditions to include or exclude content targeted to a particular audience, operating system platform, or product. Setting conditions allows you to write topics and maps that contain multiple variations of essentially identical content.

When documenting a set of software that contains topics that are nearly identical but have chunks of content specific to one particular deliverable, apply conditions to identify each product variation.

When you generate output, you can then decide what content to include or exclude. This allows you to maintain one set of topics but generate output for different customers or products.

There are many synonyms for conditions including conditional text, conditional processing, filters, and profiling attributes.

**Table 2 Default DITA Conditional Attributes**

Conditional Attributes	Usage
<b>@audience</b>	Identifies the intended audience for the content such as a programmer, administrator, or user.  Can also be used to filter by expertise such as novice, intermediate, or expert.
<b>@platform</b>	Normally used for filtering for operating systems or hardware such as "windows", or "linux".  Do not use the platform attribute for specific Wind River product versions such as "wr_linux_4". Use the <b>@product</b> attribute.
<b>@product</b>	Used for filtering by product, product features, or code names such as "vxworks_7", or "wr_linux".
<b>@otherprops</b>	Used for non-semantic processing such as identifying content that is targeted for PDF output. For example:  <code>otherprops="printonly"</code>

## Guidelines for Using Conditional Content

Using conditional text in your topics and maps requires careful planning and implementation. Incorrect use or overuse can introduce complexity and unintended file dependencies.

### Best Practices for Applying Conditional Text to Topic Content

- When working collaboratively, ensure that everybody is using the same list of conditional attribute values.
- Apply conditions only when they are needed. Common content that applies to multiple product versions does not need to be conditionalized.
- Do not use cross-references to other topics in conditional text. They cause complexity and unintentional file dependencies.
- Single sourcing is designed to help increase writer productivity so that you do not need to maintain multiple sets of topics. If a topic does not contain common content across releases and contains complex nested and compound conditions, you may consider splitting the content. Before you split your content into separate topics, consider the following:
  - You will end up with multiple topics with the same name in the repository.
  - You will need to apply product metadata within each topic to identify the product variant.
  - The Ixiasoft CCMS does not provide an interface for finding topics using product metadata.

- When adding topics to maps and adding cross-references to topics, you must be careful to select the correct topic file because there are multiple topics with the same name.
- To filter out an entire topic, apply the condition on the `<topicref>` element in the map. Applying the condition at the root element of the topic file causes a publishing error.
- Use the following table to determine where to apply conditions.

Description	Strategy
Block-level elements such as paragraphs, lists, tables, sections, or notes.	Set the condition on the current element.
Inline elements such as <code>&lt;uicontrol&gt;</code> , <code>&lt;cmdname&gt;</code> , <code>&lt;filepath&gt;</code> , <code>&lt;nameliteral&gt;</code> , or <code>&lt;xref&gt;</code> .	Do not set the condition on the inline element. Duplicate the block-level element and conditionalize at that level.
An entire sentence within a paragraph, info, table entry, or other block-level element that contains multiple sentences.	Do not conditionalize one sentence within an element. Duplicate the block-level element and conditionalize at that level.
Titles, short descriptions, book titles, and any other element that can only appear once.	Make the title or short description generic so it does not need to be conditionalized. If that strategy fails, your fallback is to apply a phrase tag and conditionalize that tag. This is an allowable exception as there is no other way to filter out conditional content in a title or short description.

### Best Practices for Applying Conditional Text to Maps

- If you are new to conditionalizing maps, schedule a code review and generate output frequently.
- When you filter out an entire topic, apply the condition on the `<topicref>` element in the map. Applying the condition at the root element of the topic file causes a publishing error.
- Eclipse maps contain Eclipse plug-in identifiers that cannot be conditionalized. Every Eclipse information set has its own Eclipse top-level map and Eclipse submap.
- Bookmaps contain metadata such as `<title>`, `<prodname>`, `<mainbooktitle>`, or `<vrm>` elements that cannot be conditionalized. For information about how to use conditional text with these elements, see the **Bookmap** entry in the following table.
- Use the following table to determine whether to conditionalize or duplicate a map.

Map Type	Strategy
Eclipse Help Top Level Map	Create a separate Eclipse map for each Help System plug-in.
Eclipse Help Submap	Create a separate Eclipse submap for each plug-in.

Map Type	Strategy
Bookmap	<p>Apply conditions to the <code>&lt;part&gt;</code>, <code>&lt;chapter&gt;</code>, <code>&lt;mapref&gt;</code>, and <code>&lt;topic*&gt;</code> elements.</p> <p>Bookmaps contain metadata such as <code>&lt;title&gt;</code>, <code>&lt;prodname&gt;</code>, <code>&lt;mainbooktitle&gt;</code>, or <code>&lt;vrm&gt;</code> elements that cannot be conditionalized.</p> <p>There are several options:</p> <ul style="list-style-type: none"> <li>- If you need to create variants of a guide with different book titles, you should duplicate your map. You should apply a condition to a map title only when it cannot be rewritten to be generic.</li> <li>- Create a separate bookmap for each book to ensure that the book-level metadata is correct and pull in the chapters from a common bookmap that contains conditions on the <code>&lt;part&gt;</code>, <code>&lt;chapter&gt;</code>, <code>&lt;mapref&gt;</code>, and <code>&lt;topic*&gt;</code> elements.</li> <li>- Create a single bookmap and use keyrefs that contain conditions to define the book-level metadata.</li> </ul>
Map	<p>Use conditions on the <code>&lt;mapref&gt;</code> and <code>&lt;topic*&gt;</code> elements.</p>

## Conditionalizing Topic Content

You can assign conditional attributes to include or exclude specific elements within topics. Set a condition to a piece of content by applying an attribute on the element.

Applying the conditions on block-level elements ensures that Wind River content can be translated to different languages and prevents the creation of "spaghetti" conditions. You can set conditions on the following elements:

- `<section>`
- `<p>`
- `<table>`
- `<ol>`
- `<ul>`
- `<li>`
- `<step>`
- `<note>`
- `<figure>`
- `<codeblock>`
- `<info>`

- <row>



**NOTE:** To filter out an entire topic, apply the condition on the <topicref> elements in the map. Applying the condition at the root element of the topic file causes a publishing error.

### Procedure

1. Insert your cursor in the desired block-level element and select one of the following attributes from the Attributes view:

Conditional Attributes	Usage
@audience	Identifies the intended audience for the content such as a programmer, administrator, or user.  Can also be used to filter by expertise such as novice, intermediate, or expert.
@platform	Normally used for filtering for operating systems or hardware such as "windows", or "linux".  Do not use the platform attribute for specific Wind River product versions such as "wr_linux". Use the @product attribute.
@product	Used for filtering by product, product features, or code names such as "vxworks_7", or "wr_linux".
@otherprops	Used for non-semantic processing such as identifying content that is targeted for PDF output. For example:

otherprops="printonly"

2. Type the desired value in the Value column or select it from the drop-down list.

### Example: Applying Conditional Text to a Block-Level Element

The following example demonstrates correctly tagged DITA markup for a topic that contains information targeted for two different hardware platforms.

```
<p platform="mips">Characters and shorts are extended to 32-bits and returned in register <nameliteral>$2</nameliteral>.</p>  
<p platform="ppc">Characters and shorts are extended to 32-bits and returned in register<nameliteral>r3</nameliteral>.</p>
```

Notice that the content has been duplicated at the paragraph level. The product conditions are set at the paragraph level, not at the word or phrase level.

### Example: Applying Conditional Text to a Title or Short Description

The following example demonstrates how to apply a phrase element to filter out conditional content in a title. You can use the same method for applying conditional text to a short description.

```
<title>
  <ph product="vxworks">RTP Application Specification for Safety Profile Systems</ph>
  <ph product="helix_drive">RTP Application Specification for Helix Drive Systems</ph>
</title>
```

 **NOTE:** You should apply a condition to a title or short description only when they cannot be rewritten to be generic.

## Conditionalizing Maps

When working with a DITA map, you can apply conditional attributes to elements to filter out entire topics, parts, chapters, or maps from your deliverable.

In a topic submap, you can apply conditional attributes to your `<topicref>` elements to filter out entire topics.

In a bookmap, you can use conditions to filter out `<part>`, `<chapter>`, `<mapref>`, and `<topic*>` elements.

 **NOTE:** To filter out an entire topic, apply the condition on the `<topicref>` elements in the map. Applying the condition at the root element of the topic file causes a publishing error.

Conditionalizing content in maps is identical to conditionalizing content in topics.

 **NOTE:** You should apply a condition to a map title only when it cannot be rewritten to be generic.

## Procedure

1. Insert your cursor in the desired element and select one of the following attributes from the Attributes view:

Conditional Attributes	Usage
<code>@audience</code>	Identifies the intended audience for the content such as a programmer, administrator, or user.  Can also be used to filter by expertise such as novice, intermediate, or expert.

---

Conditional Attributes	Usage
<b>@platform</b>	Normally used for filtering for operating systems or hardware such as "window", or "linux".
	Do not use the platform attribute for specific Wind River product versions such as "wr_linux". Use the <b>@product</b> attribute.
<b>@product</b>	Used for filtering by product, product features, or code names such as "vxworks_7", or "wr_linux".
<b>@otherprops</b>	Used for non-semantic processing such as identifying content that is targeted for PDF output. For example:
	<code>otherprops="printonly"</code>

---

2. Type the desired value in the Value column or select it from the drop-down list.

#### Example: Applying Conditional Text to a Topic Reference

The following example demonstrates how to apply conditional text to a **<topicref>** element in a map so that you can include or exclude it from your deliverable.

```
<topicref platform="linux" href="yet1430934645406.xml"/>
```

## Creating a Ditaval File

A ditaval file contains specific instructions to the build system about which conditions to include or exclude. The Ixiasoft CCMS helps you create a ditaval file based on pre-defined values for the **@audience**, **@platform**, and **@product** attributes.

Ditaval files also allow you to share conditional filters with team members when you are working on a collaborative project.

#### Procedure

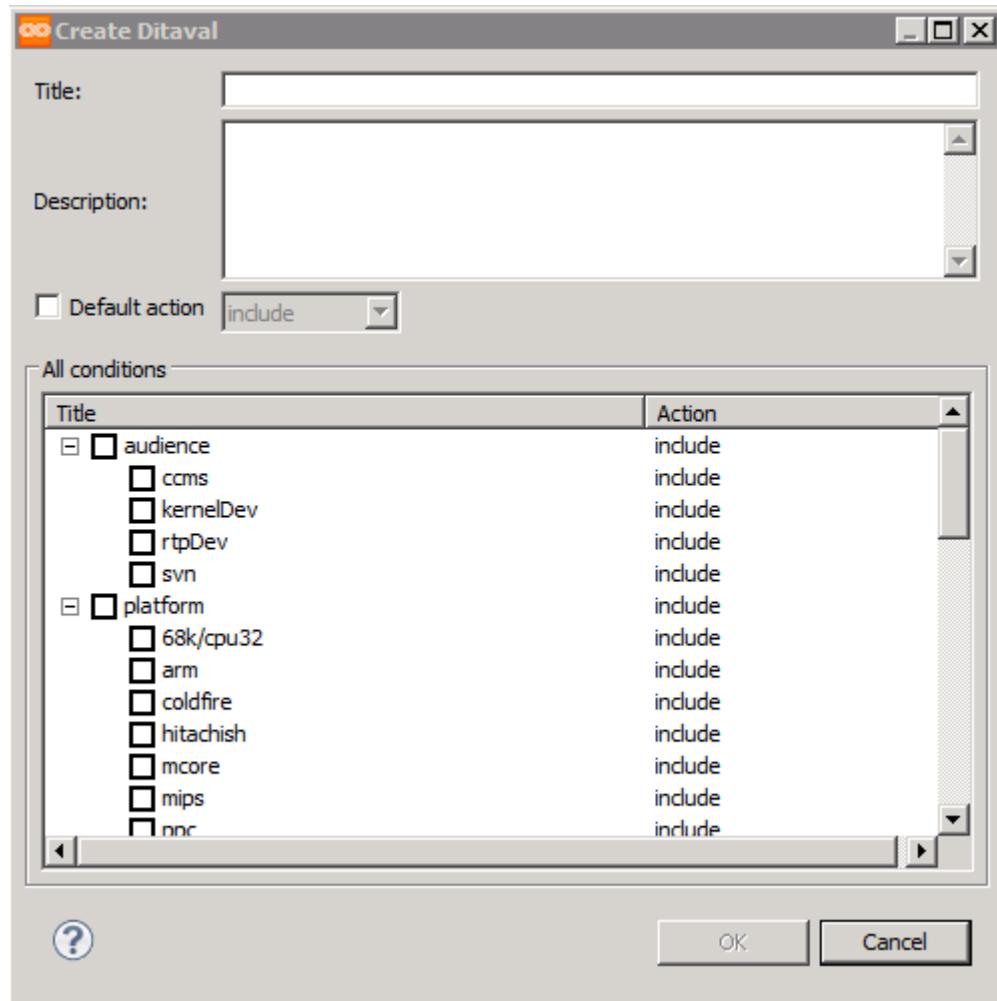
1. Open the Ditaval view.
  - a) Select **Window > Show View > Other**.
  - b) Expand **IXIASOFT CCMS - General**.
  - c) Select **Ditaval** and click **OK**.

2. If a folder for your ditaval file does not already exist in the Ditaval view, create one.



- a) In the Ditavalview, right-click **Ditaval** and select **New Ditaval Folder**.
- b) Type a name for your folder and click **OK**.
3. Right-click the desired folder and select **New Ditaval**.
4. In the Create Ditaval dialog box, type a title for your ditaval file.

The title should include a short product name and a short description; for example: **ARM - Diab**.



5. Type a description of your ditaval file.

The description should include which attribute and values are included; for example, "Includes the platform value 'arm'".

6. Select **Default action** and select **exclude** from the drop-down list.

The default DITA action is to include everything if there are no conditions specified. Selecting **exclude** is the quickest and safest way to ensure only those conditions you specify appear in your output.

7. Select the condition you want to include in the output.



**NOTE:** If the condition you need does not appear in the list, file a Jira ticket for the SUPPTOOLS project.

8. Change the action for your condition by clicking on **exclude** next to the condition and selecting **include** from the drop-down list.



**NOTE:** The **passthrough** and **flag** selections are not currently used by Wind River.

- 
9. Repeat the previous step if more than one condition is required.



**NOTE:** If you have compound or multiple conditions, refer to the *DITA Best Practices* book, which covers the Boolean logic behind compound conditions. There are several copies of this book at the various Wind River office locations.

- 
10. Click **OK**.

Your new ditaval file appears in the Ditaval view.

## Finding the ID of a ditaval File

If your document is produced by using conditional text, the ditaval ID is appended to the bundle name during transformation. If you are creating a link to the conditionalized document on the Product Documentation site, you must include the ditaval ID as part of the bundle name.

If the document you want to link to is not already posted to staging or production, you can still add the link to it by finding the appropriate ditaval file ID in the Ixiasoft CCMS.

### Procedure

1. In the Ditaval view, right-click the desired ditaval file and select **Copy > Copy ID**.
2. Add an underscore to the end of the bundle name in your href, and paste in the ditaval ID.  
For example:

```
href="/bundle/vxworks_7_programmers_guide_SR0610_dcw1556724585797"
```

## Generating Filtered Output

When you generate output in the Ixiasoft CCMS, you specify which conditions to include or exclude from your output by selecting the appropriate ditaval file.

### Procedure

1. Open the desired map in the DITA Map view.
2. Right-click the map and select **Generate Output**.
3. Select the output format.  
For more information, see [Running Generate Output](#) on page 257.
4. Make sure none of the conditions check boxes are selected.

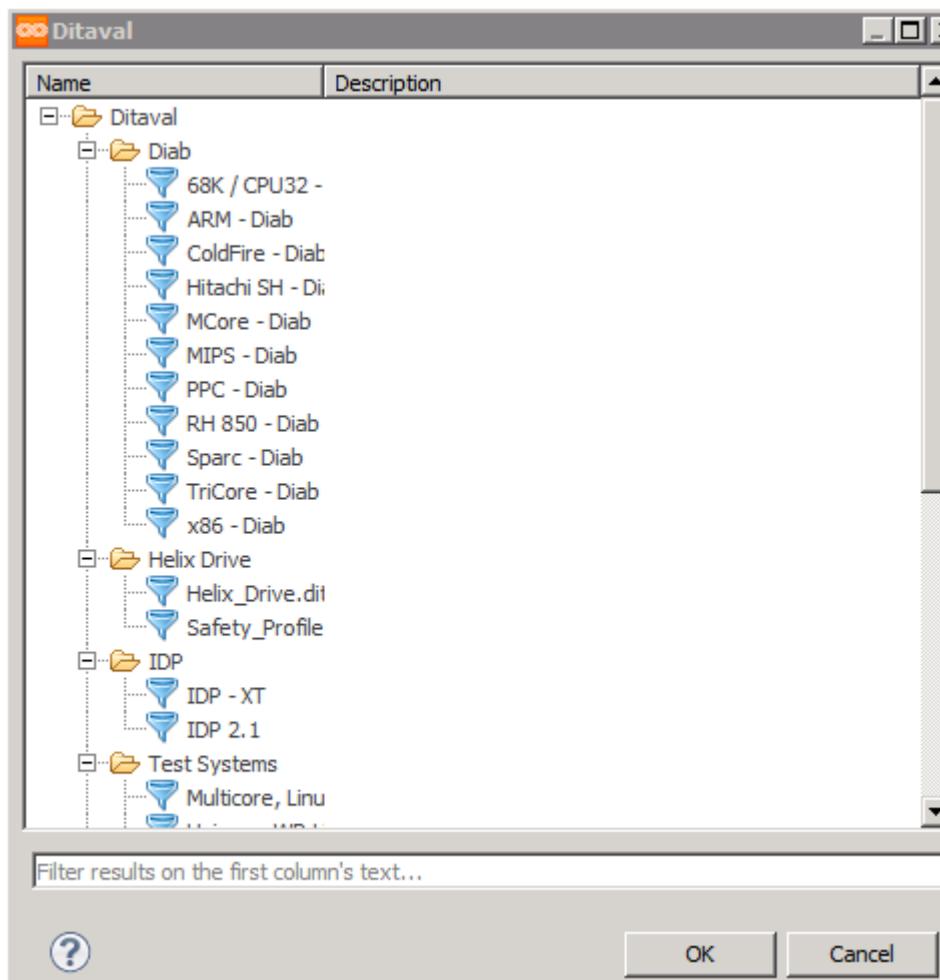


**NOTE:** The method of selecting check boxes for each condition you want to include in the output has been deprecated. Do not use this method.

5. To the right of **DITAVAL file to apply to output**, click the button with the green ellipsis (...).



6. From the Ditaval dialog box, select the desired ditaval file and click **OK**.



Ditaval files are indicated by a blue icon:



7. Click **Create**.

## Conditional Dependencies Can Prevent Publish

At the end of a documentation project, writers must tag their information set by using the Publish feature in the Ixiasoft CCMS. The algorithm that determines whether the information set can be Published does not take into account conditional processing.

Adding `<topicref>`, `<mapref>`, `<xref>`, or `<link>` elements to your information set creates dependencies in your information set. The Publish algorithm assumes that all content that you have referenced in the information set needs to be ready. Even though you may have filtered out content in the PDF and Wind River Help outputs, before you can Publish, all the content must be ready, even content that you are filtering out.

The Publish feature is only available when:

- Every topic in the information set has a status of Authoring:done.
- Every internal cross-reference target topic has been added to a DITA map.
- Every internal cross-reference target topic has a status of Authoring:done.

You are unable to Publish your topic set if your content contains internal cross-references to topics not included in the map or to topics that are not in the Authoring:done state.

### Related Links

[Validating Links](#) on page 270

Before you can publish your document, you must ensure it contains no out-of-scope links. You cannot publish with invalid links.

## Keyrefs/Variables

### Keywords/Variables Overview

You can use keywords, or variables, to reuse strings that may change often.

#### Workflow

The workflow for using keywords in your documentation is as follows:

1. [Create a key map.](#)

The key map stores the key definitions, or the actual strings for your variables.

2. [Add the key map to your bookmaps.](#)

To enable the use of keywords in your content, you must add the key map containing the key definitions to your bookmap as a resource. You must add the key map to each bookmap in your top-level map. If the key map is not included, the keywords do not resolve.

3. [Add the keyword, or variable, to your content.](#)

You must add a `<keyword>` element to your content and define the `@keyref` attribute to point to the defined string in the key map.

## DITA Elements and Attributes

Before adding keywords to your content, be familiar with some of the DITA elements and attributes.

### <keydef> element

You use this element in a key map to define each string you want to reuse.

### @keys attribute

This is essentially the ID for your <keydef> element. The value of this attribute must be globally unique. You use this ID as the value of the @keyref attribute when you add the <keyword> element in your topics.

### <keywords> element

This is the container element you must add to your key map to add a <keyword> element.

### <keyword> element

In a key map, this element contains the actual string that you want to reuse. For example:

```
<keydef keys="linux-transition-guide-title">
    <topicmeta>
        <keywords>
            <keyword>Wind River Linux Transition Guide for the Yocto Project</
keyword>
        </keywords>
    </topicmeta>
</keydef>
```

In a topic, you use an empty <keyword> element (with only a @keyref attribute defined) to indicate where you want the string to be added in your content. For example:

```
<p>For more information, see the <keyword keyref="linux-transition-guide-
title"/>. </p>
```

### @keyref attribute

The value of this attribute is the same as the @keys attribute for the desired <keydef> in the key map.

## Guidelines for Using Keywords/Variables

You can use keywords, or variables, to reuse strings or items that change frequently. Using keywords in your topics and maps requires careful planning and implementation. Incorrect use or overuse can introduce complexity.

Use the following guidelines when adding variables to your content:

- When working collaboratively, ensure that everybody is using the same key definitions.
- Use a separate map to create your key definitions. Do not add key definitions to your existing Eclipse maps, bookmaps, or chapter maps.
- Add your key definition map as a resource to each of your bookmaps.
- Do not use keywords in topic or map titles. The keywords do not resolve when using Search in the Ixiasoft CCMS.
- Use keywords sparingly, and only for those items that may change frequently; for example, a product or component name that changes frequently.

- Do not use keywords just because you do not want to type the string or item over many times in a document. If it takes you longer to create the keyword than it does to type the item, just type the item.
- Do not use keywords inside `<keydef>` elements. For example, if you have a keydef that defines an `@href` attribute for a cross-reference, and you want to use a keyword for the link text, the following markup in your key map will not resolve properly into a link in the output:

```
<keydef href="njh1572366777737.xml" keys="admin_tutorials_xref">
    <topicmeta>
        <linktext><keyword keyref="admin_tutorials_title"/></linktext>
    </topicmeta>
</keydef>
```

For examples of using keywords, see [Keyword Examples](#) on page 255.

## Creating a Key Map

A key map stores the key definitions, or the actual strings for your variables.

### Procedure

1. Create a new chapter map.

Use **Product Doc - Topics Submap** for Information Development or **Training - Lab Submap (Chapter)** for Training.

Give the map a meaningful name, and add "Key Map" as part of your map title; for example, "Linux Global Variables Key Map".

2. Add a `<keydef>` element to your key map.

3. Add a `@keys` attribute to the `<keydef>` element with a unique value.

The value of the `@keys` attribute is the string you will use to add the keyword to your content. It must be globally unique within the Ixiasoft CCMS and should be descriptive of the string defined in your key definition. For example, for a key definition that stores the name of a product guide, you could use something like the following:

```
<keydef keys="linux-transition-guide-title"></keydef>
```

4. Add a `<topicmeta>` element to the `<keydef>` element.

For example:

```
<keydef keys="linux-transition-guide-title">
    <topicmeta></topicmeta>
</keydef>
```

5. Add a `<keywords>` element to the `<topicmeta>` element.

For example:

```
<keydef keys="linux-transition-guide-title">
    <topicmeta>
        <keywords></keywords>
    </topicmeta>
</keydef>
```

6. Add a **<keyword>** element containing the string you want to reuse to the **<keywords>** element.

For example, if you want to reuse the name of a product manual, your key definition would look similar to the following:

```
<keydef keys="linux-transition-guide-title">
    <topicmeta>
        <keywords>
            <keyword>Wind River Linux Transition Guide for the Yocto Project</
        keyword>
        </keywords>
    </topicmeta>
</keydef>
```

7. Repeat steps 2 through 6 for each definition you want to add to the key map.

## Adding a Key Map to Your Bookmap

To enable the use of keywords in your content, you must add the key map containing the key definitions to your bookmap as a resource.

You must add the key map to each bookmap in your top-level map. If the key map is not included, the keywords do not resolve.

### Procedure

1. Add a **<mapref>** element to your bookmap.

The **<mapref>** element must be added in the **<frontmatter>** element before the **<booklists>** element.

For example:

```
<frontmatter>
    <mapref/>
    <booklists>
        <toc/>
    </booklists>
</frontmatter>
```

2. Add an **@href** attribute pointing to the appropriate key map to the **<mapref>** element.

For example:

```
<frontmatter>
    <mapref href="mmo1441999784297.ditamap"/>
    <booklists>
        <toc/>
    </booklists>
</frontmatter>
```

3. Add a **@format** attribute set to **ditamap** to the **<mapref>** element.

For example:

```
<frontmatter>
    <mapref href="mmo1441999784297.ditamap" format="ditamap"/>
    <booklists>
        <toc/>
    </booklists>
</frontmatter>
```

4. Add a @processing-role attribute set to **resource-only** to the <mapref> element.

For example:

```
<frontmatter>
    <mapref href="mmo1441999784297.ditamap" format="ditamap" processing-
role="resource-only"/>
    <booklists>
        <toc/>
    </booklists>
</frontmatter>
```

## Adding a Keyword/Variable to Your Content

To reuse strings in your content, you must add an empty <keyword> element and add the appropriate value to the @keyref attribute.

### Procedure

1. Add a <keyword> element to your content where you want the variable to appear.
2. Add a @keyref attribute to the <keyword> element and set the value to the appropriate key as defined in your key map.



**NOTE:** As long as you have included the appropriate key map in your bookmap as a resource, you can select the @keyref values from a drop-down list in both Text mode and Author mode.

## Keyword Examples

There are many ways you can use keywords in your documentation.

Following are some of the more common reuse scenarios.

### Reusing a Product Name

Suppose you have a product called Wind River Hurricane Cloud. However, the product name is volatile and changes from release to release. You want to use a variable for the product name so that you can change/maintain it in one place and have it automatically populate into many places in your document.

To do this, define the product name as "Wind River Hurricane Cloud" in a <keyword> element in your keymap; for example:

```
<keydef keys="wr-hurricane-prod-name">
    <topicmeta>
        <keywords>
            <keyword>Wind River Hurricane Cloud</keyword>
        </keywords>
    </topicmeta>
</keydef>
```

In your topics, add a **<keyword>** element where you want the product name to appear and set the value of the **@keyref** attribute to the same value you used in the **@keys** attribute in your key map. For example:

```
<p><keyword keyref="wr-hurricane-prod-name"/> provides the following features:</p>
```

The **<keyword>** element is replaced with "Wind River Hurricane Cloud". If the product name changes, you only have to change the keyword text in the key map. It is automatically updated in your content.

### Reusing a Document Title

Suppose you want to reuse a document title. You define it in your key map as follows:

```
<keydef keys="linux-transition-guide-title">
  <topicmeta>
    <keywords>
      <keyword>Wind River Linux Transition Guide for the Yocto Project</keyword>
    </keywords>
  </topicmeta>
</keydef>
```

In your topics, add the **<keyword>** element as follows:

```
<p>For more information, see the <keyword keyref="linux-transition-guide-title"/>.</p>
```

### Reusing an @href Attribute

Suppose you want to link to the top-level of a book. To do so, you must link to the first topic in the book. Since that topic ID may change from release to release, you define the **@href** attribute in your keymap as follows:

```
<keydef href="njh157236677737.xml" keys="admin_tutorials_xref"/>
```

In your topic, add the **<xref>** as follows:

```
<p>For more information, see the <xref keyref="admin_tutorials_xref" scope="peer">Cloud Platform Kubernetes Admin Tutorials</xref>.</p>
```

If you have the book name already defined in your keymap, you could also use a **<keyword>** element inside the **<xref>** element to define the link text; for example:

```
<p>For more information, see the <xref keyref="admin_tutorials_xref" scope="peer"><keyword keyref="admin_tutorials_title"/></xref>.</p>
```

# *Generating Output*

[Running Generate Output](#) 257

[Wind River Supported Output Formats](#) 259

## **Running Generate Output**

Wind River provides transforms for generating PDF and HTML. You can generate PDFs and HTML for review or final output.

### **Prerequisites**

- Ensure all your changes have been released to the Ixiasoft CCMS.

The Ixiasoft CCMS generates output from the topics in the repository; changes you have made in locked topics do not appear.

If you are reviewing topics but intend to continue working on them, you can select **Keep locked** when you release the topics. This updates the document base but leaves the topics locked so you can continue working on them.

- You must create an Eclipse Help map and an Eclipse Help submap if you are creating documentation for Wind River Help.

### **Procedure**

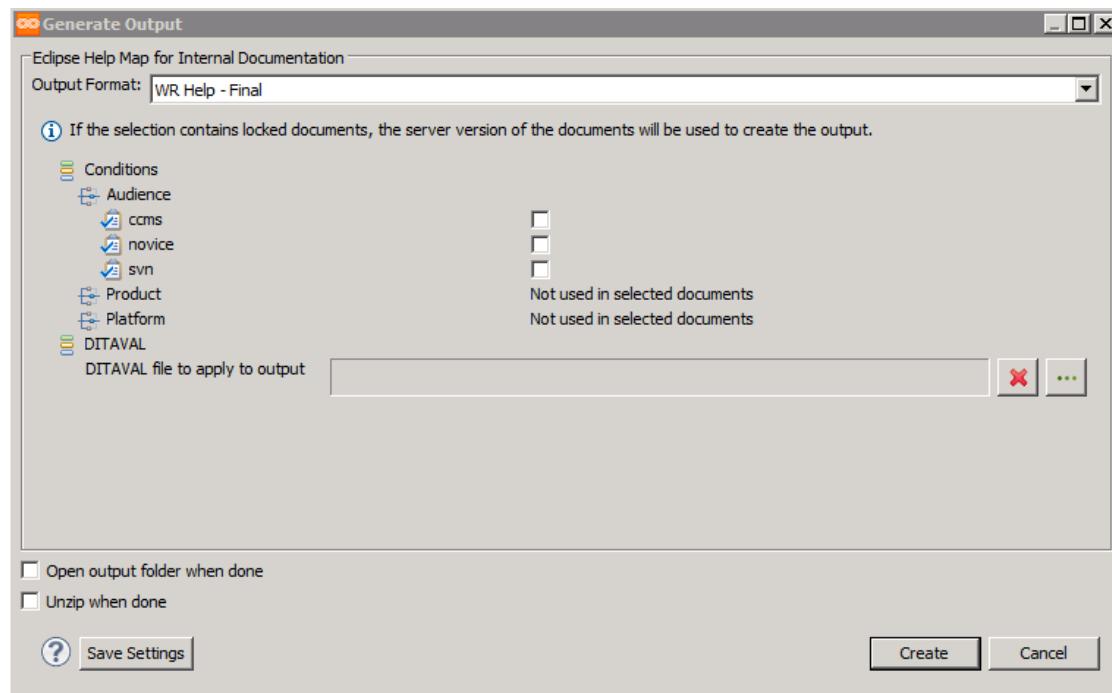
1. Open the map you want to generate output for in the DITA Map view.

For product documentation, you can generate output from an Eclipse Help map or from a bookmap.

For training documentation, you can generate output from a lab bookmap or a lab submap (chapter).

You can also generate output from topics or topic submaps; however, the formatting of the output is not optimal. You should create output from topics or topic submaps for review purposes only.

2. Right-click on the map and select **Generate Output**.



3. Select the kind of output to generate.

For more information, see [Wind River Supported Output Formats](#) on page 259.

4. If your content includes conditional text, select the appropriate ditaval file.

For more information, see [Generating Filtered Output](#) on page 249.



**NOTE:** The method of selecting check boxes for each condition you want to include in the output has been deprecated. Do not use this method.

5. Click **Create**.

When the build finishes, a save dialog appears.

6. Save the output.

The .zip file name reflects the book title and the transform, for example:

**Wind\_River\_DITA\_Internal\_Technical\_Writer\_s\_Guide\_  
WR PDF - Draft.English.zip**

Create a folder in your workspace to save the ZIP file. The folder name should identify the document, date, and possibly the time or sequence, for example:

E:\NoScan\041114\TechWritersGuide  
E:\NoScan\041414\pluginA

E:\NoScan\041414A  
E:\NoScan\041414-10:15am

In the next steps, these folders with their path are represented as *outputDir*.

7. Unzip the file and test the document.

Type of File	File to Open
Single book PDF:	<i>outputDir\book-id.pdf</i> The PDF has a name similar to <b>dia1396906071725_0.0.pdf</b> .
PDF in plugin:	<i>outputDir\plugin-name-release\topics\book-name-release.pdf</i>
HTML in plugin:	<i>outputDir\plugin-name-release\book-name-release\html\index.html</i>
Plugin in Wind River help:	<ol style="list-style-type: none"><li>1. Copy the plugin into a Workbench installation after renaming the existing plugin of the same name.</li><li>2. Start Workbench or the standalone help browser and navigate to your documents.</li></ol>

#### Postrequisites

After you generate output, you should review the *fileID.ot.log* file, whether your build was successful or not.

For more information, see [Reviewing the Error Logs](#) on page 294.

## Wind River Supported Output Formats

Wind River provides transforms for generating PDF and HTML for product and training documentation. Selecting the appropriate output format ensures your documentation is structured and formatted appropriately.

**NOTE:** Some output formats are supported for review purposes only. The formatting of the output is not optimal. Do not use these formats for generating customer-facing documentation.

**NOTE:** Some of the output formats are appended with a version number; for example, PDF - Draft-1.5.4. These are specific versions to be used only for the Linux and Titanium Cloud products. Due to the size and complexity of these products the output formats used for other documentation cause the build to fail.

	Content Type						
Output Format	Eclipse Help Map	Eclipse Help Submap	Bookmap	Topics Submap	Topic	Lab Book	Lab Submap
<b>PDF - Draft</b>	Supported for review purposes only		X	Supported for review purposes only	Supported for review purposes only		
<b>PDF - Draft-1.5.4</b>	Supported for review purposes only		X	Supported for review purposes only	Supported for review purposes only		
<b>Training PDF - Draft</b>						X	X
<b>Training PDF - Final</b>						X	X
<b>WR Help - Draft</b>	X						
<b>WR Help - Draft-1.5.4</b>	X						
<b>WR Help - Final</b>	X						
<b>WR Help - Final-1.5.4</b>	X						
<b>Zoomin-SFTP-production</b>			X				
<b>Zoomin-SFTP-staging</b>			X				

# 30

## *Delivering Documentation*

[Release Management Practices for Product Documentation Delivery \(Placeholder\)](#) 261

[Release Management Practices for Customer Training Documentation Delivery](#) 262

### **Release Management Practices for Product Documentation Delivery (Placeholder)**

#### **What to Deliver (Placeholder)**

Concept definition.

#### **Where to Deliver (Placeholder)**

Concept definition.

#### **Delivering Product Documentation (Placeholder)**

Context for the current task

#### **Procedure**

Task step.

## Release Management Practices for Customer Training Documentation Delivery

### What to Deliver

When you release customer training documents, you must build and deliver the complete set of deliverables for the course.

The complete set includes the following:

- separate lecture and lab books. Each book may exist as a single volume or as multiple volumes (identified on the cover as "Volume 1 of 2", and so forth.).
- individual lecture chapters, as both the original MSPowerPoint source files and as PDF files generated from the MSPowerPoint source
- individual lab chapters, as PDF files generated from the DITA source (or, for older labs, the MSWord source)

Before you start the release process you must build these deliverables with the ats-bb book build tool (based on files described in the **.metadata** file for the course and content in the **Learning Modules** repository on the WRU server, [http://wru.wrs.com/svn/  
Learning\\_Modules](http://wru.wrs.com/svn/Learning_Modules)), and have the course owner verify and approve them for release. For detailed book building instructions, see:

<https://jive.windriver.com/docs/DOC-29130>

In addition to the course books and related chapter files, course owners release a lab software image that provides a standard, preconfigured environment that students use to perform the lab exercises. The course owner is responsible for the lab software image — it is not considered to be part of the work of releasing the course books.

### Where to Deliver

You must release all customer training course documents in the appropriate release repository on the WRU server.

The release repositories are SVN version-controlled repositories located at:

<http://wru.wrs.com/websvn/?sc=0>

### Where to Release Standard Course Documents

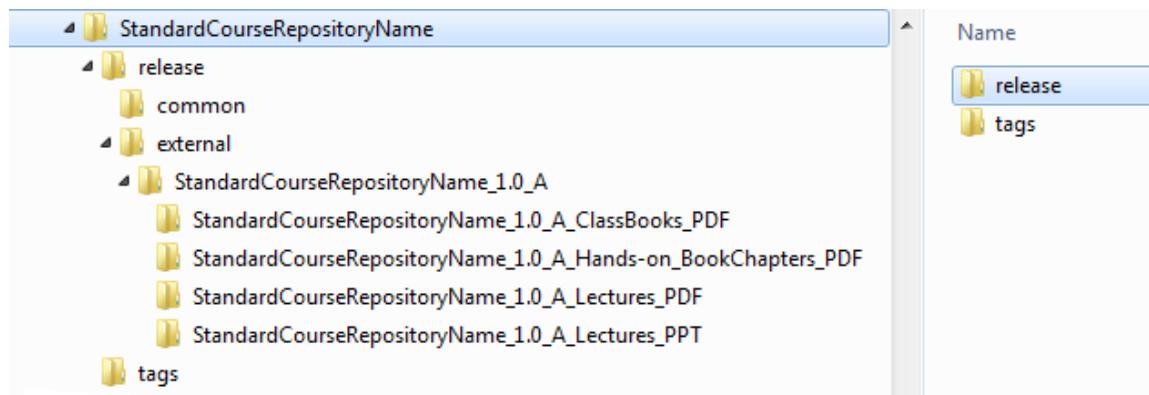
Release documents for standard courses in the appropriate release repository on the WRU SVN server, such as [VxWorksEssentials](#), [LxAppDevelopment](#), and so forth.

The file [Release Repositories for Standard Customer Training Courses](#) lists the standard courses and the correct release repository to use for each course.

Each standard course release has a standard folder structure. You must create the folder structure that will hold the deliverables that constitute the release. For instructions about how to create the folder structure for a standard course release, see:

## [Creating the Folder Structure for a Standard Customer Training Course Release](#)

The following figure shows an example of the standard structure.



The standard folders must contain the following:

- **StandardCourseRepositoryName/release** – Contains the folders for currently active releases (books and chapter files). This may include multiple versions for different product versions on which we offer training courses (for example, Wind River Linux product versions 4.0, 4.1, 5.0.1, 6.0, and 7.0). It must also contain the metadata file that ats-bb uses to build the release documents (for example, **StandardCourseRepositoryName\_Books.metadata**).
- **StandardCourseRepositoryName/release/common** – Must contain the PPT source files that describe the prerequisites and objectives for the course. For example:
  - 00\_StandardCourseRepositoryName\_Prerequisites.ppt**
  - 00\_StandardCourseRepositoryName\_Objectives.ppt**
- **StandardCourseRepositoryName/release/external** – Contains the folders that hold the books and chapter files for the currently active releases. For example:
  - StandardCourseRepositoryName\_1.0\_B**
  - StandardCourseRepositoryName\_2.0\_A**

Note the following:

- The name of the folder for any release must use the standard naming convention shown—the repository name, followed by the release ID (number and letter), separated by underscore characters.
- The release ID is typically the number that represents the Wind River product that the training addresses, plus an alphabetic character to identify the succession of releases of the course for that product version. For example, the VxWorks Essentials courses use the identifiers 6.9 A, 6.9 B, 7.0 A, 7.0 B, 7.0 C, and so forth. Courses that do not relate to a Wind River product typically use an ID such as 1.0 A, 1.0 B, and so forth.
- **StandardCourseRepositoryName/release/external/StandardCourseRepositoryName\_1.0\_A** – Contains the folders that hold the books and chapter files for an individual release (in this example, release 1.0 A). Always name these folders according to the following convention:
  - **StandardCourseRepositoryName\_1.0\_A\_ClassBooks\_PDF** – Holds the course books.
  - **StandardCourseRepositoryName\_1.0\_A\_Hands-on\_BookChapters\_PDF** – Holds the lab book chapter PDF files.
  - **StandardCourseRepositoryName\_1.0\_A\_Lectures\_PDF** – Holds the lecture books chapter PDF files.

- **StandardCourseRepositoryName\_1.0\_A\_Lectures\_PPT** – Holds the lecture books chapter MS PowerPoint source files.
- **StandardCourseRepositoryName/tags** – Contains the archive of obsolete, superseded releases (books and chapter files).

### Where to Release Tailored Course Documents

Release documents for tailored courses in an appropriate location in the [http://wru.wrs.com/svn/Custom\\_Trainings](http://wru.wrs.com/svn/Custom_Trainings) repository.

Put the folder structure for the tailored course release in a subfolder; for example, **Custom\_Trainings/2015-H1** for courses released in the first half of 2015.

For instructions on how to create the folder structure for each tailored course similar to the folder structure for a standard course, see:

[Creating the Folder Structure for a Tailored Course Release](#)

Name the release folder according to the following convention:

*CustomerName TrainingLocation\_StandardCourse1-StandardCourse2*

For example, the folder for a tailored course for General Dynamics (the customer, abbreviated to GD), delivered at their offices in Scottsdale, AZ, with modules selected from the standard courses Wind River Linux 7 Essentials and Wind River Linux User Space Programming would be:

**GD\_ScottsdaleAZ\_Lx7Ess-Appdev**

This naming convention helps Operations identify any specific course easily.

### Delivering Customer Training Documentation

The delivery process for customer training documentation entails creating the proper directories in the repository and building and tagging the documentation. You must also get the appropriate approvals, ensure that the release notes have been updated, and perform repository maintenance.

#### Procedure

1. Create the folder structure for the release in the appropriate release repository.

Go to one of the following links for detailed instructions:

- For standard courses, go to <https://jive.windriver.com/docs/DOC-35162>.
- For tailored courses, go to <https://jive.windriver.com/docs/DOC-29122>.

2. Build the test books and obtain approval.

For detailed instructions, see <https://jive.windriver.com/docs/DOC-29130>. You may have to complete additional fix/rebuild cycles if the course owner finds unexpected content problems.

- a) Build the test books using the content from the **/trunk** branch of each module with the **Draft** option selected.
- b) Implement any fixes required to build good books.
- c) Place the resulting books in the folder for the release.
- d) Obtain approval from the course owner that the test books look good.

3. Tag all modules to `/tags/latest` (or `/tags/latest-xx`, if you are building books that are not for the latest product version) and to a snapshot archive for this release (for example, `AbbrevCourseTitle_1.0_A`).

For detailed instructions, see <https://jive.windriver.com/docs/DOC-15062>.

4. Build release candidate (RC) books and related chapter files using content from the `/tags/latest` branch of each module.
5. Obtain verification and release approval from the course owner.
6. Have the course owner update the course release note, stored in the release repository in the folder `StandardCourseRepositoryName/release/external`.
7. Have the course owner update the PAM.

The PAM is located at:

[http://wru.wrs.com/svn/CommonDocs/TrainingPAM/WR\\_Training\\_PAM.xlsx](http://wru.wrs.com/svn/CommonDocs/TrainingPAM/WR_Training_PAM.xlsx)

8. Move any old releases that the new release supersedes, to archive them in the release repository.

Move the old release from `StandardCourseRepositoryName/release/external` to `StandardCourseRepositoryName/tags`.



---

**NOTE:** Remember that you must do an SVN move, not just a Windows folder move.

---

9. Have the course owner announce general availability (GA) to Operations.

For detailed instructions, see <https://jive.windriver.com/docs/DOC-29139>.



# 31

## *Cloning*

[Cloning Best Practices](#) 267

[Cloning Files](#) 268

### **Cloning Best Practices**

Cloning files can save you time and effort when you want to split a topic into two, copy the structure of an existing file, or make extensive changes that should not affect the original file. However, you should adhere to the Wind River best practices.

When cloning a file, always create a new Modification History entry and indicate what file you copied to create this one. Ensure that the **Author** field shows your name.

Cloning is the opposite of content reuse. Cloning files implies that you may need to maintain multiple sets of topics. Before you clone your content into separate topics, consider the following:

- You will end up with multiple topics with the same name in the repository.
- You will need to apply product metadata within each topic to identify the product variant.
- When adding topics to maps and adding cross-references to topics, be careful to select the correct topic file because there are multiple topics with the same name.
- Cloning a file does not manage dependencies. Therefore, if the original file contains a cross-reference to another topic, and you include the clone in a different bookmap than the original file, you will have out-of-scope links.

## Cloning Files

Create a clone to split a topic into two, copy the structure of an existing file, or make extensive changes that should not affect the original file. The cloning process copies the current file in the Ixiasoft CCMS and assigns it a new topic ID.

### Prerequisites

You can only clone released files. If a file is locked, the **Clone** option is grayed out.

### Procedure

1. Locate the file in the DITA Map view or by searching.
2. Right-click the file name and select **Clone** in the drop-down list.  
The **Clone** option is grayed-out and unavailable if the file is locked.
3. Select the appropriate options in the Clone dialog box.

Option	Description
<b>Replace original component</b>	The cloned map, topic, or image is created and added to the containing map and the original map, topic, or image is removed from the map but is not deleted. All external cross references that point to the original map, topic, or image are kept as is; they are not updated to point to the cloned map, topic, or image. Internal cross references (from point A to point B in the original topic) are updated in the cloned topic.
<b>Append to parent</b>	The cloned map, topic, or image is added to the containing map. When you select this option, the <b>Replace original component</b> option is disabled.
<b>Open in default editor</b>	Lock and open the file for editing.

 **NOTE:** If you do not append the file to the parent, it does not appear in a map. You must locate the file in your Todo List or search for it.

# *Publishing*

[Publishing Overview](#) 269

[Validating Links](#) 270

[Changing Resource Status](#) 272

[Publishing a Map](#) 272

[Regenerating Content from a Published Map](#) 274

## **Publishing Overview**

Publishing is the action of tagging a final, official version of a document in the Ixiasoft CCMS at a given point in time (usually a product release).

The publishing action is done on the top-level map for your document (usually an Eclipse Help map or bookmap for product guides or a bookmap for training). During publishing, you assign a version tag to the map. This tag is automatically applied to all child maps, topics, and resources. The tag allows you to locate the published version of the document so you can either regenerate your published document or make changes to it (through branching).

A copy of the document is created in a separate "Published" area in the Ixiasoft CCMS. Because this is just a copy of the document, and not an editable version, all the content IDs remain the same until you branch from the published copy.

Further work in the mainline trunk does not affect the published copy. Likewise, regenerating your published document or updating it by branching does not affect the work you are doing on the mainline trunk.

### **When to Publish**

Publish your document when it is final:

- the content has gone through final review
- final output has been generated

- the output has been uploaded and checked on the final release location (Product Documentation site, on the media, the course release repository on the WRU SVN server, and so forth)
- all files are unlocked and free of validation errors (see [Validating Links](#) on page 270)
- the map and all its referenced files are in the Authoring:done state (see [Changing Resource Status](#) on page 272)

Do not publish a document before it is final. You can generate draft output at any time, regardless of the current status.

### What to Publish

Always publish the top-level map for your document. Typically, this is the Eclipse Help map for product guides or a bookmap for training.

## Validating Links

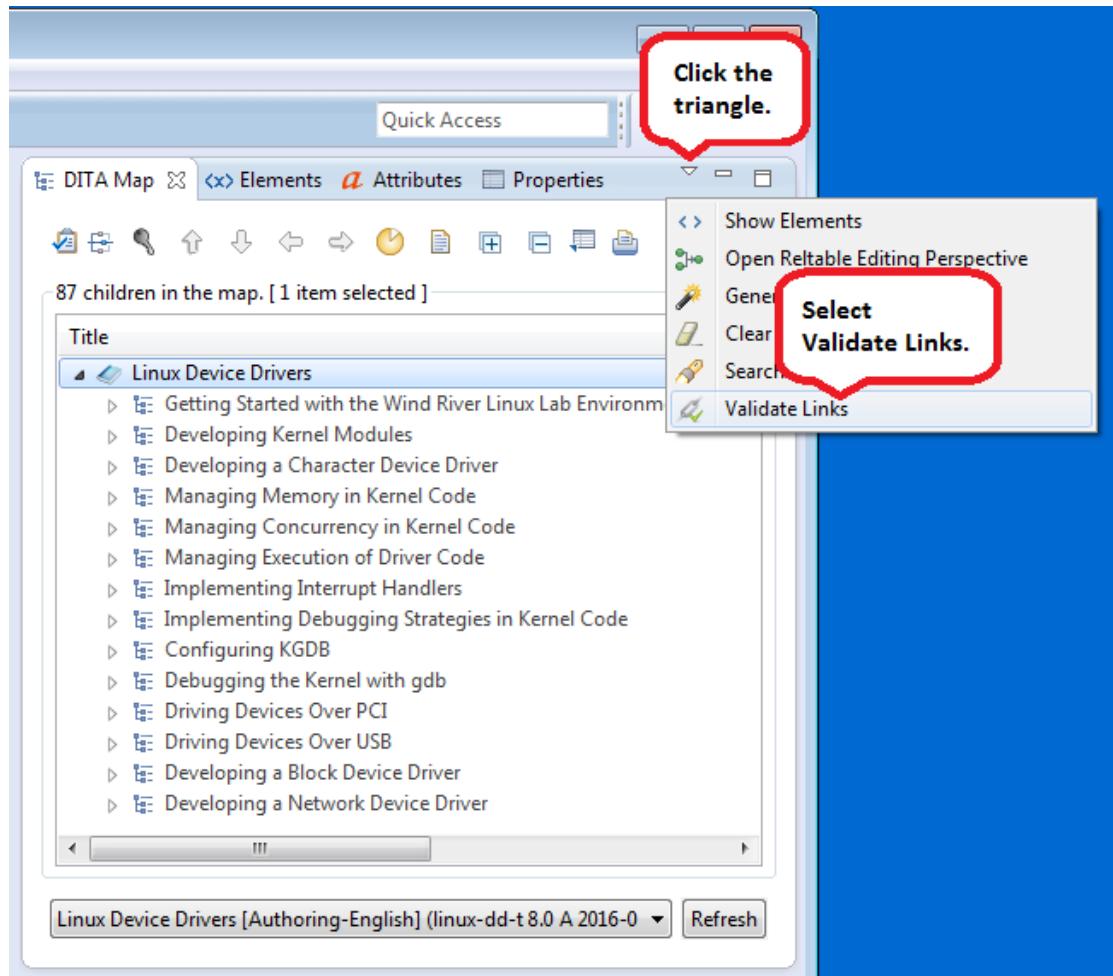
Before you can publish your document, you must ensure it contains no out-of-scope links. You cannot publish with invalid links.

### Procedure

1. Open the map you want to publish in the DITA Maps view.
2. Click on the top-level document you wish to validate.

For product documentation this is either an Eclipse Help map or a bookmap. For training documents this is usually a bookmap.

3. Click on the triangular menu icon at the upper right of the view and select **Validate Links** from the drop-down list.



If the validation succeeds, a confirmation window appears.

If the validation fails, an error window appears.

Detailed Information		
Level	Title	Description
⚠	Building Advantech UTX-3115 Boards [rec1394031140025]	This reference [/content/authoring/rec1394031140025.xml] is out of the map scope.
⚠	Deploying Quark Boards Using a Script [dia1394827580320]	This reference [/content/authoring/dia1394827580320.dita] is out of the map scope.
⚠	Building Platform Projects for Quark Boards [rec1394031139822]	This reference [/content/authoring/rec1394031139822.xml] is out of the map scope.
⚠	Deploying Advantech UTX-3115 Boards Using a Script [rec1394031147583]	This reference [/content/authoring/rec1394031147583.xml] is out of the map scope.
⚠	Performing a Secure Boot on Cross Hill and Clanton Hill Boards [rec1394031147896]	This reference [/content/authoring/rec1394031147896.xml] is out of the map scope.
⚠	Deploying Advantech UTX-3115 Boards Manually [rec1394031147740]	This reference [/content/authoring/rec1394031147740.xml] is out of the map scope.
⚠	Deploying Quark Boards Using a vfat-Formatted USB Drive [dia1394827472515]	This reference [/content/authoring/dia1394827472515.dita] is out of the map scope.
⚠	Performing a Verified Boot [rec1394031148099]	This reference [/content/authoring/rec1394031148099.xml] is out of the map scope.
⚠	McAfee Embedded Control [rec1394031140369]	This reference [/content/authoring/rec1394031140369.xml] is out of the map scope.
⚠	Setting Up a Zigbee Network [rec1394031143720]	This reference [/content/authoring/rec1394031143720.xml] is out of the map scope.
⚠	Performing a Secure Boot on Advantech UTX-3115 Using UEFI [dia1395696238114]	This reference [/content/authoring/dia1395696238114.dita] is out of the map scope.
⚠	Preparing USB Boot Media [rec1394031147286]	This reference [/content/authoring/rec1394031147286.xml] is out of the map scope.

## Postrequisites

If the validation succeeds, you are ready to change resource states.

If the validation fails, you must find and fix the out-of-scope links. For more information, see [Finding Out-of-Scope Links](#) on page 297.

## Changing Resource Status

Before you can publish a map, all maps and topics included in the map must have a status of Authoring:done.

The status appears by default in both the Search Results view and the DITA Map view. You can use either view to change the resource status of your maps and topics. However, you can use the Search view to limit your search to just those items that need to be changed. For more information, see [Using Search to Prepare Your Book for Publishing](#) on page 56.

---

 **NOTE:**

- You cannot move any map to the Authoring:done status until all topics and maps it contains are in the Authoring:done status.
  - You cannot move any topic to the Authoring:done status until all images and any other objects it references are in the Authoring:done status.
- 

### Procedure

1. Move all images to Authoring:done.
  - a) Highlight one or more images, right-click on one of them, and select **Change status**.
  - b) Select **Authoring:done** and click **Change**.
2. Move all topics to Authoring:done.
  - a) Highlight one or more topics, right-click on one of them, and select **Change status**.
  - b) Select **Authoring:done** and click **Change**.
3. Move all maps to Authoring:done.
  - a) Highlight one or more maps, right-click on one of them, and select **Change status**.
  - b) Select **Authoring:done** and click **Change**.

## Publishing a Map

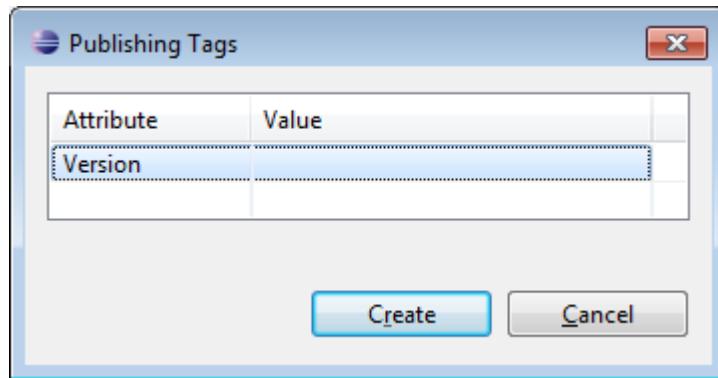
After your document is final and you have generated output for release to the customer, you must publish your document to create a final, official copy in the Ixiasoft CCMS.

When you publish a map, you add a version tag that is applied to all child maps and topics, so you can reproduce the document if necessary in the future.

### Procedure

1. In the DITA Map view, right-click the map you are publishing and select **Publish**.

The Publishing Tags dialog box appears.



2. In the Publishing Tags dialog box, in the **Value** field of the **Version** attribute, type the version tag.

Your version tag must include the following information, in the order listed:

- a product name abbreviation

Use the appropriate standard label from the list maintained on the Jive page <https://jive.windriver.com/docs/DOC-36486>.

- a product release number
- a publication date in the format YYYY-MM-DD
- an optional note wrapped in parenthesis

For example:

idp-xt 3.0 2015-08-06 (GA)

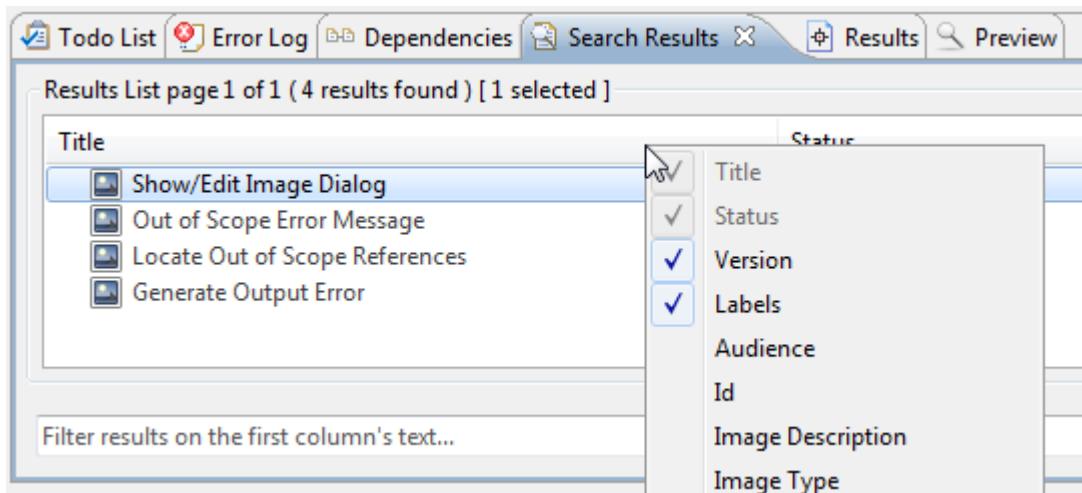
3. Click **Create**.

The Ixiasoft CCMS creates a copy of the book in a separate Published area of the Ixiasoft CCMS.

When you search for a published version of your document, the version tag appears in the **Version** column of the Search Results view.

Title	Status	Id	Type	Version
Changing WiFi Mode from AP to Client Using LuCI	Published:done	dho140779669498	task-wr	IDP-XT 3.0 2015-08-06 (GA)
About Authentication, Authorization, and Auditing Examples	Published:done	jke1437158462192	concept-wr	IDP-XT 3.0 2015-08-06 (GA)
Configuring the BIOS for Intel Bay Trail and Intel Haswell Boards	Published:done	dho1403613732418	task-wr	IDP-XT 3.0 2015-08-06 (GA)
Changing WiFi Mode from Client to AP Using LuCI	Published:done	dho140779797103	task-wr	IDP-XT 3.0 2015-08-06 (GA)
Overview	Published:done	rec1394031132585	concept-wr	IDP-XT 3.0 2015-08-06 (GA)
Determining IDP Version Information	Published:done	dho1406205354886	task-wr	IDP-XT 3.0 2015-08-06 (GA)
About Customizing LuCI	Published:done	rec1394031175488	concept-wr	IDP-XT 3.0 2015-08-06 (GA)

If your Search Results view does not display the **Version** column, right-click in the heading row and select **Version**.



#### Postrequisites

You can now continue working in the mainline trunk for the next revision of the document.

## Regenerating Content from a Published Map

After you have published a document, you may need to regenerate the output to produce an exact copy of the content that has already been released to customers.

Reasons for regenerating your output include:

- The document has been inadvertently deleted from the final release location.
- There has been a change to our transform tools; for example, to handle new branding or new formatting (perhaps to make the Product Documentation site more efficient).

#### Procedure

1. Do a search for the published version of the document.
2. Right-click the top-level map for the document and select **Generate Output**.

# 33

## *Branching*

<a href="#">When to Branch</a>	275
<a href="#">What to Branch</a>	276
<a href="#">Creating a New Branch</a>	278
<a href="#">Branching a Bookmap or Submap</a>	279
<a href="#">Branching a Topic</a>	280
<a href="#">Branching an Image</a>	280
<a href="#">Branching a Conref</a>	281
<a href="#">Branching a Referenced Resource</a>	282

### **When to Branch**

In most cases, you only need to branch when you have already started work on a new version of the document on the mainline trunk and need to go back and make changes or bug fixes to a previously published version of the document.

Branching in the Ixiasoft CCMS differs from branching during code development. In code development, the use case is to branch and create a new version of the code for the next release, rather than developing on the mainline trunk. However, in the Ixiasoft CCMS it is the opposite. In almost all cases, you write your documentation on the mainline trunk and branching is the exception. You only branch when you must make changes or bug fixes to a previously published version of the documentation.

Branching lets you create a new, editable version of a previously published map. You can modify content in the branched map without affecting the original published content or the content on the mainline trunk.

The following table provides several scenarios that may help in your decision about when to branch.

Scenario	Method
<p>You have already started work on a new version of the document on the mainline trunk and need to go back and make changes to a previously published version of the document.</p> <p>For example, you have published and released the <i>IDP Programmer's Guide</i> for the 2.0 release. You have started work on the 2.1 release on the mainline trunk and need to go back and update the 2.0 document for the 2.0.1 release.</p>	<p>You must branch.</p> <p>For example, you would create a branch of the published map for the 2.0 document and make changes as necessary.</p>
<p>You need to update a previously published version of a document and have not made any updates or changes to the content in the mainline trunk since publishing the document.</p> <p>For example, you published the <i>Helix Cockpit Platform Developer's Guide, 1.0</i>, but there are some changes that must be made to some topics and a new topic must be added. You have not started work on the next version of Helix Cockpit.</p>	<p>Do not branch.</p> <p>Change the state of the topics that need to be updated back to Authoring:work, make your changes, and then publish the document again.</p>
<p>An old product has been discontinued and replaced by a new product with similar content. You want to use the topics from the old product as a starting point for the new product documentation.</p>	<p>Do not branch.</p> <p>Make sure the last version of documentation for the old product is published. This allows you to return to old product documentation to make changes in the future, if needed.</p> <p>Start work on the new product documentation using the maps and topics in the mainline trunk as needed. The mainline topics can be safely repurposed because no new development is expected on this product and the last final version is preserved through publication.</p>

## What to Branch

When you branch correctly, the Ixiasoft CCMS handles dependencies. If branching is done at the wrong level, you will have out-of-scope validation errors.

You must create a new branch using a published version of the top-level map for your document.

 **NOTE:** You can only branch from published maps.

For most product documentation the top-level map is the Eclipse Help map (**Product Doc - Eclipse Help Map.ditamap**).

If your document does not use an Eclipse Help map (for example, training documentation), you must branch at the bookmap level (**Training - Lab Book.ditamap**, **Product - Bookmap.ditamap**, or **Product - Bookmap for Books with Parts.ditamap**).

When you branch, the branched copy of the map is created in the Authoring cycle and given the status of **Authoring:work**. Initially, the new map references the same published content as the original map. The submaps, topics, and other content in the map remain in the Published cycle.

If you want to modify the content of any of the items (topics, submaps, bookmaps, and so forth) within the branched map, you must branch the individual item as well. The system creates a new copy of the item and automatically takes care of external references. If any topics or maps containing a reference to the branched document are still in the Published cycle, then they are automatically branched as well.



**CAUTION:** When you branch bookmaps, submaps, or individual topics, you must branch them from within your branched top-level map.

If you attempt to branch them from the published versions, rather than from within the top-level map, in essence you create a new, separate branch. Your maps or topics will be divorced from the top-level map, and your changes will not appear when you build from the top-level map.

Scenario	Method
You need to modify a topic in a previously published version of a document.	Create a new branch from the top-level map for the document, branch the topic from within the top-level map in the DITA Map view, and modify the topic.
You need to add a topic to a chapter in a previously published version of a document.	Create a new branch from the top-level map for the document, branch the chapter submap from within the top-level map in the DITA Map view, and add the new topic to the chapter submap.
You need to add a chapter to a previously published version of a document.	Create a new branch from the top-level map for the document, branch the bookmap from within the top-level map in the DITA Map view, and add the new chapter submap to the bookmap.
You need to add a new book to a previously published Eclipse Help map.	Create a new branch from the Eclipse Help map for the document, branch the Eclipse Help Submap from within the Eclipse Help map in the DITA Map view, and add the new book reference to the Eclipse Help Submap.

## Creating a New Branch

You must create a new branch if you have already started work on a new version of the product on the mainline trunk and need to go back and make changes to a previously published version of the document.

### Procedure

1. Do a search for the top-level map for a previously published version of the document.



**CAUTION:** You must create a new branch from the top-level map only. Do not create a new branch from an individual topic or a submap included in a top-level map.

For most product documentation the top-level map is the Eclipse Help map (**Product Doc - Eclipse Help Map.ditamap**).

If your document does not use an Eclipse Help map (for example, training documentation), you must branch at the bookmap level (**Training - Lab Book.ditamap**, **Product - Bookmap.ditamap**, or **Product - Bookmap for Books with Parts.ditamap**).

2. Right-click the map and select **Branch > Create New Branch**.



**NOTE:** You can only branch from published maps. If you try to branch from a map on the mainline trunk, the **Create New Branch** option is grayed out.

The Branch Map dialog box appears. The **Tag** field automatically populates with the version number for the published map. For example:

idp-xt 2.0.1 2015-01-06

3. Accept the default for the version tag and click **OK**.

A new editable version of the map is created in the Authoring:work state. The version tag appears in brackets before the title of your map. This tag provides a record of the published version of the document used to create the branch. You can use the version tag to search the repository for all documents belonging to a specific branch.

### Postrequisites

If you want to modify the content of any of the items (topics, submaps, bookmaps, and so forth) within the branched map, you must branch each item as well.

## Branching a Bookmap or Submap

You must branch a bookmap or submap if you want to add new topics or maps to a branched document.

For example, you have branched the Eclipse Help map for the *IDP Programmer's Guide* for the 2.0.1 release, and you need to add a new topic to a chapter. You must branch the submap for the chapter (**Product Doc - Topics Submap.ditamap**) before you can add the topic.

 **NOTE:** When you branch a bookmap or submap from within an already branched top-level map, you are not creating a new branch, you are just changing the state of the bookmap or submap to allow it to be edited on the previously created branch.

If you just need to edit or modify an existing topic, you can branch at the topic level. For more information, see [Branching a Topic](#) on page 280.

### Prerequisites

You must branch only from an already branched top-level map. For more information, see [Creating a New Branch](#) on page 278.

 **CAUTION:** Do not create a new branch from a bookmap or a submap included in a top-level map.

If you attempt to branch them from the published versions, rather than from within the top-level map, in essence you create a new, separate branch. Your maps or topics will be divorced from the top-level map, and your changes will not appear when you build from the top-level map.

### Procedure

1. Do a search for the branched top-level map for the document.
2. Open the map in the DITA Map view.
3. In the DITA Map view, right-click the appropriate map and select **Branch > Branch Map**.

The system creates a new editable version of the map in the Authoring:work state. The Ixiasoft CCMS applies the same version tag to the map as the branched top-level map.

## Branching a Topic

If you want to modify the content of any of the topics within a branched map, you must first branch the topics.

### Prerequisites

- You must branch only from an already branched top-level map. For more information, see [Creating a New Branch](#) on page 278.



### CAUTION:

Do not create a new branch from a topic included in a top-level map.

If you attempt to branch from the published version of a topic, rather than from within the top-level map, in essence you create a new, separate branch. Your topics will be divorced from the top-level map, and your changes will not appear when you build from the top-level map.

- All referring documents must be in a state that allows them to be branched.

For example, if your topic references another topic or is referenced by another topic, the top-level map, any parent maps for the affected topics, and the topics themselves must not be locked. If they are locked, the Ixiasoft CCMS is unable to resolve the references and branch the maps and topics, as necessary.

### Procedure

1. Do a search for the branched top-level map for the document.
2. Open the map in the DITA Map view.
3. In the DITA Map view, right-click the topic you want to branch and select **Branch > Branch Topic**.

The system creates a new copy of the topic and automatically takes care of external references. If any topics or maps containing a reference to the branched document are still in the Published cycle, then they are automatically branched as well.

The Ixiasoft CCMS applies the same version tag to the topic as the branched top-level map.

## Branching an Image

After you have branched a topic, you can then branch any images it contains.

### Prerequisites

- You must branch from an already branched topic. For more information, see [Branching a Topic](#) on page 280.

- The top-level map that contains the branched topic must be opened in the DITA Map view.
- The image file must not be locked.
- All topics or maps referring to the image must be in a state that allows them to be branched.

#### Procedure

1. In the DITA Map view, right-click your topic and select **Branch > Branch Referenced Images**.
2. If the topic contains more than one image, select the desired image from the Branch Images dialog box and click **OK**.  
If branching the image requires that other topics must also be branched, the Branch Documents dialog box appears. It displays a list of topics that are affected by the change and asks if you want to continue.  
The system creates a new copy of the image. The Ixiasoft CCMS applies the same version tag to the image as the branched top-level map.
3. Perform a search for your branched image and update as necessary.

For information on how to edit images, see [Editing Images](#) on page 184.

## Branching a Conref

After you have branched a topic, you can then branch the conrefs contained in the topic.

#### Prerequisites

- You must branch from an already branched topic that contains the conref. For more information, see [Branching a Topic](#) on page 280.



**CAUTION:** Do not create a new branch directly from a referable-content topic.

If you attempt to branch from the published version of the referable-content topic, rather than from within the branched topic inside the top-level map, in essence you create a new, separate branch. Your topics will be divorced from the top-level map, and your changes will not appear when you build from the top-level map.

- 
- The top-level map that contains the branched topic must be opened in the DITA Map view.
  - All topics containing a conref to the referable-content topic must be in a state that allows them to be branched.

#### Procedure

1. In the DITA Map view, right-click your topic and select **Branch > Branch Conrefs**.  
The system creates a new copy of all the referable-content topics referenced by the branched topic. The Ixiasoft CCMS applies the same version tag to the referable-content topics as the branched top-level map.
2. Perform a search for your branched referable-content topics and update as necessary.

## Branching a Referenced Resource

After you have branched a topic, you can then branch any resources it references.

### Prerequisites

- You must branch from an already branched topic that contains the reference to the resource file. For more information, see [Branching a Topic](#) on page 280.
- The top-level map that contains the branched topic must be opened in the DITA Map view.
- All topics containing a reference to the resource file must be in a state that allows them to be branched.

### Procedure

1. In the DITA Map view, right-click your topic and select **Branch > Branch Referenced Resources**.

The system creates a new copy of all the resource files referenced by the branched topic. The Ixiasoft CCMS applies the same version tag to the resource files as the branched top-level map.

2. Perform a search for your branched resource file and update as necessary.

# *Snapshots*

<a href="#">Snapshot Overview</a>	283
<a href="#">Creating a Snapshot</a>	284
<a href="#">Searching for a Snapshot</a>	285

## **Snapshot Overview**

A snapshot captures the status of a map at a particular point in time.

A snapshot is an **.xml** file that points to the exact versions of all the items (maps, topics, images, and so forth) in the map at the time the snapshot is created. The **.xml** file contains the following information for each of the items in the map:

- status of the item
- link to the item in the Ixiasoft CCMS
- navtitle
- revision number
- object type

You can reproduce the map exactly as it was when the snapshot was created. You can also generate output, publish, and branch from a snapshot.



**NOTE:** The Show Preview feature does not currently work for snapshots. It is a known problem, and should be fixed with the next upgrade to the Ixiasoft CCMS.

### **Benefits of Snapshots**

Snapshots have the following benefits over the publish procedure:

- You do not have to have all the items in your map in the Authoring:done state to create a snapshot. You can create a snapshot of a map in any state, as long as it is not locked.



---

**NOTE:** The map must not be locked, but items within the map can be locked; however, the most recent changes may not be captured. The snapshot only includes versions of items that have been released in the Ixiasoft CCMS.

- Unlike when you publish a map, copies of the map and all the items contained in the map are not made. This saves space on the server.

### Drawbacks of Snapshots

If a topic (or other item) included in your map is locked, the most recent changes may not be captured. The snapshot only includes versions of items that have been released in the Ixiasoft CCMS.

### Best Practices

- Make sure all the items in the map you are creating a snapshot of are not locked. This ensures that all changes to the items are captured.
- Create a snapshot when you send your document for review. This creates a historical record of the map. You can recreate it at any time, or you can revert to the snapshot version if necessary.
- Create a snapshot before you post your document to the Product Documentation site (or other location). However, the snapshot does not replace the publish procedure.
- Follow the same guidelines for creating the version tag as you do for publishing.

Your version tag must include the following information, in the order listed:

- a product name abbreviation
- Use the appropriate standard label from the list maintained on the Jive page <https://jive.windriver.com/docs/DOC-36486>.
- a product release number
  - the date the snapshot is created in the format YYYY-MM-DD
  - an optional note wrapped in parenthesis

For example:

idp-xt 3.0 2015-08-06 (review)

## Creating a Snapshot

When you create a snapshot, the Ixiasoft CCMS creates an **.xml** file that points to the exact versions of all the items (maps, topics, images, and so forth) in the map at the time the snapshot is created.

### Prerequisites

Your map must not be locked.

#### Procedure

1. Use the **Search** view to locate the map.
2. Right-click the map in the **Search Results** view (or in the DITA Map view if the map is open) and select **Create snapshot**.
3. In the Create Snapshot dialog box, leave the title of the map as the default.
4. Type the version tag in the **Version** text box.

Your version tag must include the following information, in the order listed:

- a product name abbreviation  
Use the appropriate standard label from the list maintained on the Jive page <https://jive.windriver.com/docs/DOC-36486>.
- a product release number
- the date the snapshot is created in the format YYYY-MM-DD
- an optional note wrapped in parenthesis

For example:

idp-xt 3.0 2015-08-06 (review)

5. Click **OK**.

## Searching for a Snapshot

You must configure the settings in the Search view to search for snapshots.

#### Procedure

1. In the Cycles section of the Search view, select **Authoring**.
2. In the Document Types section, select **snapshot** from the **Others** drop-down list.
3. Clear the selection for all other document types.
4. If you know the title of your snapshot, type the full title or just the first few words of the title in the **Search for** text box.
5. Click **Search**.

#### Postrequisites

You can generate output, publish, or branch from your snapshot as necessary.



**NOTE:** To publish or branch, you must change the status of your snapshot from Authoring:work to Authoring:done.



## *The WIP Site*

[About the WIP Site](#) 287

[Posting a Local Build to WIP](#) 289

[Customizing the TOC by Editing the index.php File](#) 290

[Adding Custom Content Using a custom.html.txt File](#) 291

### **About the WIP Site**

The Information Development work in progress site (WIP site) is a location on a shared server where you can post document drafts for review. For some projects, GA versions of documentation are also available.

The WIP site is located at:

`\delphi\delphi2\httpd\htdocs\engineering\engops\techpubs\docDrafts`

All members of Information Development should have permission to edit these folders. If you do not have access, contact your manager.

### **Directory Structure**

The typical directory structure for the WIP site is as follows:

```
docDrafts
  projDir
    prodRelease
      docDir
      patches
```

`projDir`

This is the top-level folder for your project. It should be named after the product. For example:

**Helix Device Cloud**

**Linux**

This folder contains folders for one or more product releases and the PHP TOC generator file, **index.php**.

You can also [add custom content](#) that is displayed at the top of your project page by including a **custom.html.txt** file.

#### *prodRelease*

Typically, product release folders are named for the particular releases of a product. For example:

**Linux 9**

**Pulsar Linux 8**

However, as an alternative, you may wish to use the standard Eclipse structure. For example:

**com.windriver.ide.doc.vxworks\_mils\_3.0**

**com.windriver.ide.doc.wr\_net\_stack\_6.6**

If you use this approach, you can copy your DITA output folder directly to the project directory, rather than creating the product release folder manually.

The product release folders contain folders for one or more documents. It may also contain a **patches** folder.

For the product release folders to show up in the TOC, they must contain an **index.html** file or an **index.php** file.

#### *docDir*

The documentation directory is used to store the HTML and PDF files for an individual document. The folder name should be descriptive of the document. For example:

**wr\_linux\_9\_kernel\_configuration\_and\_build**

**vxworks\_7\_architecture\_supplement**

When you generate output in DITA, these folders are created in your output directory. You can copy the entire document directory from your output directory to the product release directory, rather than creating the documentation directory manually.

#### **patches**

Used to store downloadable documentation patches and other **.zip** files.

### **The PHP TOC Generator**

The **index.php** file automatically generates nested TOCs for folders that contain either an **index.html** file or an **index.php** file.

The **.zip** files in the **patches** folder are also linked in the HTML output.

Links back up to the parent directory are created for child TOCs.

### **Search**

The WIP search facility uses the Google appliance inside the Wind River firewall. It allows you to use a search form that searches only the **docDrafts** folder and the folders it contains.

## Posting a Local Build to WIP

For reviewers to see the output you have generated, you must post it on a public server, usually the Work in Progress site.

### Procedure

1. Map the location of the **docDrafts** folder to your local machine.

**\\\ala-nas2\delphi2\httpd\htdocs\engineering\engops\techpubs\docDrafts**



**NOTE:** If you wish to use FTP to transfer files to the WIP site, connect to:

- Server: **delphi.wrs.com**
- Password: Use your UNIX password.
- Path: **/delphi2/httpd/htdocs/engineering/engops/techpubs/docDrafts**

2. Open the folder in Windows Explorer.

3. Create a new project folder if one does not already exist.

This folder name appears on the TOC for the Work in Progress page. Use the following guidelines when creating the folder:

- For product documentation, the folder name should be the product name.
- The folder name should not contain a version number.
- The folder name may contain spaces.
- The folder name must describe the contents.
- Long names are a better choice than cryptic abbreviations.

4. Create a product release folder.

Typically, product release folders are named for the particular releases of a product. For example:

**Linux 9**  
**Pulsar Linux 8**

However, as an alternative, you may wish to use the standard Eclipse structure. For example:

**com.windriver.ide.doc.vxworks\_mils\_3.0**  
**com.windriver.ide.doc.wr\_net\_stack\_6.6**

5. Optional: Create a folder named **patches** in the product release folder.

This is where you will place any **.zip** files that you want to make available for download.

6. Create a documentation folder.

The folder name should be descriptive of the document. For example:

## **wr\_linux\_9\_kernel\_configuration\_and\_build** **vxworks\_7\_architecture\_supplement**

7. If your folders do not contain either an **index.html** or **index.php** file, copy the **index.php** file from **docDrafts** to your folder.

You must have an **index.php** file or **index.html** file at every level. Typically, the book folders already contain an **index.html**, so a copy of **index.php** is not needed.

For content that has an **index.html** file plus subfolders, do not add an **index.php** file to the subfolders.

8. Copy your content to desired documentation folder.

9. Test your WIP posting in your browser at [the WIP site](#).

## **Customizing the TOC by Editing the index.php File**

The **index.php** file automatically generates a title and a subtitle for your WIP pages. You can customize these titles by editing the HTML in the **index.php** file.

By default, the title and subtitle are created as follows:

- The title is created by appending **- Work in Progress** to the project directory name.
- The subtitle is created from text contained in an **<h3>** element. The default is **Projects**.

### **Procedure**

1. Open the **index.php** file in your favorite editor.

2. Modify the title by editing the following line:

```
<title><?php echo $mydir;?> - Work in Progress</title>
```

The PHP tag shown below displays the name of the folder:

```
<?php echo $mydir;?>
```

Do not modify the HTML tags, **<title>** or **</title>**.

3. Modify the subtitle by editing the following line:

```
<h3>Projects</h3>
```

Do not modify the HTML tags, **<h3>** or **</h3>**.

4. Save the file.

## Adding Custom Content Using a `custom.html.txt` File

By default, the text displayed at the top of a WIP page (in addition to the automatically generated titles and list of documents and patches) comes from the `index.php` file. You can replace this text with custom content by creating a `custom.html.txt` file.

### Procedure

1. Create a file named `custom.html.txt` using your favorite editor.
2. Using HTML markup, enter any content you want to appear at the top of your WIP page.

For example:

```
<p>Note: This site contains the work in progress for any new VxWorks 7  
documentation. This site is built regularly from the documentation source.  
All documentation included on this page is in Draft form and is for internal  
use only.</p>
```

This content will replace the text contained in the `index.php` file.

3. Save the file in your project folder.



# 36

## *Troubleshooting*

[Isolating the Reason for Build Failure](#) 293

[Reviewing the Error Logs](#) 294

[Common Error Messages Reference](#) 295

[Finding Out-of-Scope Links](#) 297

### **Isolating the Reason for Build Failure**

If your build fails, you must isolate the reason for failure before you can fix the problem. The failure can be due to a problem with the output generator, usage of an improper output format, or an issue with the source content.

If you build your documents regularly, isolating the reason for the build failure can be relatively easy. You can assume the problem is with one of the source files you recently updated. However, if you have not built your document for quite some time, isolating the problem can be more difficult.

#### **Procedure**

1. Ensure that you are building the proper files and using the proper output format.

For example, some common mistakes are:

- building the Eclipse submap instead of the Eclipse Help map
- using the **WR Help - Final** or **WR Help - Draft** output format when attempting to build a bookmap.

For more information, see [Wind River Supported Output Formats](#) on page 259.

2. Try to build a different map using the same output format.

For example, try to build the last published version of the map you are trying to build.

If the build succeeds, then the issue is with your current content.

If the build fails, it may be due to a problem with the output generator. Contact an Infrastructure team member.

3. If you have determined the problem is with your content, you can try narrowing down the maps/topics by commenting them out.

For example, if you are building an Eclipse Help map that contains multiple books, comment out half the books and rebuild. If the build is successful, you know the problem is with one of the books you commented out.

Once you have isolated the book that is failing, use the same method to narrow it down to the chapter map, and finally the topic.

4. [Review the error logs.](#)

## Reviewing the Error Logs

After you generate output, review the *fileID.ot.log* file, whether your build was successful or not.

Some errors do not block the build. However, they should be resolved as a part of regular file maintenance.

### Procedure

1. Open the *fileID.ot.log* file located in the *.zip* file for your build.
2. If your build failed, go to the end of the file and look at the messages above "Build Failed".  
If there are Java stack trace lines above "Build Failed", look for useful information in the lines above those stack trace lines.



**NOTE:** Although there are sometimes only 10 or 20 stack trace lines, there can be 100 or more.

Stack trace lines look similar to this:

```
at
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43
)
```

3. Search the file for DITA Open Toolkit errors.

These errors contain a message ID, severity information, and message text. They all start with "DOT"; for example, DOTA001F.

A list of DITA Open Toolkit errors is located at:

<http://www.dita-ot.org/1.8/readme/DITA-messages.html>

4. Search for the term *error*.



**NOTE:** When searching make sure you are not doing a case-sensitive search. Sometimes the term error is lowercase and sometimes it is capitalized or in all capital letters.

For a list of some of the more common errors, see the [Common Error Messages Reference](#) on page 295.

If your error is not listed in the reference, or you need assistance, contact an Infrastructure team member.

## Common Error Messages Reference

Understanding what some of the more common error messages mean will help you to find and fix the errors in your source files.

### Unable to Parse Invalid href Attribute Value

Error message:

```
[filter] D:\out-gen\CMS42\temp\WR Help - Final.harrowoo.139214.1507569556445\content
\authoring\har1427988616728.xml:27:114:
[DOTJ054E] [ERROR] Unable to parse invalid href attribute value "http://
www.windriver.com/licensing/documents",
using 'http://www.windriver.com/licensing/documents%20'.
```

Cause:

There is an extra space in the URL.

Fix:

Remove the extra space.

### File Not Found in Our Books

Error message:

```
[xslt] [ERROR] file [dho1416412915507.xml] not found in our books (#1)
```

Cause:

A topic links to another topic that is not included in the top-level map.



**NOTE:** Most of the time, you can also see the out-of-scope links by using the Validate Links tool. However, if your content includes conditional text, and you must build your top-level map several times to get the final output (for example, if you are using one bookmap to produce four different, conditionalized documents), these error messages will appear, even if no out-of-scope links are found using the Validate Links tool. If this is the case, you can safely ignore the error messages.

Fix:

Find the out-of-scope links using either of the following methods:

- Use the topic ID from the error message to search for the topic, and then use the Dependencies view to find which topics in your book reference the topic.
- Use the Validate Links tool. For more information, see [Finding Out-of-Scope Links](#) on page 297.

### Error Code: 10768 (2A10)

Error message:

```
[exec] AHFCmd :WARNING: Error Level : 2
[exec] AHFCmd :WARNING: Error Code : 10768 (2A10)
[exec] AHFCmd :WARNING: Duplicate id value:
id="unique_34_Connect_42_steps_fhb_vp4_lbb".
[exec] AHFCmd :WARNING: Line 525, Col 1458, file:///D:/out-gen/CMS42/temp/WR%20Help%20-%20Final.harrowoo.139214.1507569556445/temp/topic.fo
```

Cause:

Duplicate ID values. Most likely, this error is caused by copying and pasting elements from one topic to another.

Fix:

Use the Search and Replace tool for the top-level map to find the error by using the last part of the ID listed in the error message as the search term and clicking **Search**. For example, the ID from the error message above is:

```
id="unique_34_Connect_42_steps_fhb_vp4_lbb"
```

You would use **steps\_fhb\_vp4\_lbb** as the search string.

If the element with the duplicate ID is an element you need to link to, delete the ID and then generate a new ID attribute. For more information, see [Generating ID Attributes](#) on page 202.

If the element is not an element you would link to (for example, the **<steps>** element from the above error), you can just delete the ID; you do not need to generate a new one.

### Error Code: 23427 (5B83)

Error message:

```
[exec] AHFCmd :INFO: Error Level : 1
[exec] AHFCmd :INFO: Error Code : 23427 (5B83)
[exec] AHFCmd :INFO: Fallbacked glyph U+2610 (->) to 'MS PGothic'.
[exec] AHFCmd :INFO: Line 1652, Col 615, file:///D:/out-gen/CMS42/temp/WR%20Help%20-%20Final.harrowoo.146555.1507584512999/temp/topic.fo
```

Cause:

The most common cause is a symbol that has been copied and pasted from another source; for example, a paragraph symbol from a Word document.

Fix:

This error is much harder to find. If you have recently built your document without the error, you can narrow it down to recently updated topics. Otherwise, you must use the process of elimination to find the topic. Because there is no way to tell what the symbol is, you must carefully review the topic to find and remove the symbol.

## Build Failed Due to Input Error

Error Message:

This is a fatal error. The build fails with an error message similar to the following:

```
-----
-----
-----
BUILD FAILED DUE TO INPUT ERROR
-----
-----
"Wind River DITA User's Guide" contains an error which must be fixed by modifying
your DITA maps.

FATAL ERROR: Referencing the same topic(s) more than once in the same bookmap is
unsupported.

Refer to the DITA User's Guide or seek assistance from the Infrastructure team to
learn how to correctly reuse the same topic in a book.

Correct the following duplicate topic references:

"Creating Links Between Topics" dia1400196785130.dita      (@class=- map/topicref )
Total time: 8 minutes 50 seconds
```

Cause:

The same topic is included in more than one place in your book.

Fix:

If you need to include a topic several times in the same bookmap, you must create a referable-content topic that contains the reused topic. For more information, see [Including a Topic Multiple Times in the Same Book](#) on page 236.

## Finding Out-of-Scope Links

If you find that you have out-of-scope links after running the Validate Links tool, you must find and resolve those links before you can publish your document.

### Procedure

1. Run the Validate Links tool.

For more information, see [Running the Validate Links Tool](#) on page 34.

If the validation fails, an error window appears.

Detailed Information		
Level	Title	Description
Warning	Building Advantech UTX-3115 Boards [rec1394031140025]	This reference [/content/authoring/rec1394031140025.xml] is out of the map scope.
Warning	Deploying Quark Boards Using a Script [dia1394827580320]	This reference [/content/authoring/dia1394827580320.dita] is out of the map scope.
Warning	Building Platform Projects for Quark Boards [rec1394031139822]	This reference [/content/authoring/rec1394031139822.xml] is out of the map scope.
Warning	Deploying Advantech UTX-3115 Boards Using a Script [rec1394031147583]	This reference [/content/authoring/rec1394031147583.xml] is out of the map scope.
Warning	Performing a Secure Boot on Cross Hill and Clanton Hill Boards [rec1394031147896]	This reference [/content/authoring/rec1394031147896.xml] is out of the map scope.
Warning	Deploying Advantech UTX-3115 Boards Manually [rec1394031147740]	This reference [/content/authoring/rec1394031147740.xml] is out of the map scope.
Warning	Deploying Quark Boards Using a vfat-Formatted USB Drive [dia1394827472515]	This reference [/content/authoring/dia1394827472515.dita] is out of the map scope.
Warning	Performing a Verified Boot [rec1394031148099]	This reference [/content/authoring/rec1394031148099.xml] is out of the map scope.
Warning	McAfee Embedded Control [rec1394031140369]	This reference [/content/authoring/rec1394031140369.xml] is out of the map scope.
Warning	Setting Up a Zigbee Network [rec1394031143720]	This reference [/content/authoring/rec1394031143720.xml] is out of the map scope.
Warning	Performing a Secure Boot on Advantech UTX-3115 Using UEFI [dia1395696238114]	This reference [/content/authoring/dia1395696238114.dita] is out of the map scope.
Warning	Preparing USB Boot Media [rec1394031147286]	This reference [/content/authoring/rec1394031147286.xml] is out of the map scope.

**Save**   **Locate**   **Copy**

2. Click **Locate** in the error window to locate the files that are not in the map you are testing.
3. In the Search Results view, expand the **Locator** line to see the list of topics.

Combined Results List (14 results found)				
Title	Status	Version	Labels	
Locator [12 / 12]				
Deploying Advantech UTX-3115 Boards Manually	Authoring:work	IDP XT 2.0.1-EAR; IDP XT 2.0	IDP	
Performing a Secure Boot on Cross Hill and Clanton Hill Boards	Authoring:done	IDP XT 2.0.1-EAR; IDP XT 2.0	IDP	
Preparing USB Boot Media	Authoring:done	IDP XT 2.0.1-EAR; IDP XT 2.0	IDP	
Deploying Quark Boards Using a vfat-Formatted USB Drive	Authoring:done	IDP XT 2.0.1-EAR; IDP XT 2.0		
Building Platform Projects for Quark Boards	Authoring:done	IDP XT 2.0.1-EAR; IDP XT 2.0	IDP	
Deploying Quark Boards Using a Script	Authoring:done	IDP XT 2.0.1-EAR; IDP XT 2.0		
Performing a Secure Boot on Advantech UTX-3115 Using UEFI	Authoring:done	IDP XT 2.0.1 04/01/14; IDP X		
Deploying Advantech UTX-3115 Boards Using a Script	Authoring:work	IDP XT 2.0.1-EAR; IDP XT 2.0	IDP	
McAfee Embedded Control	Authoring:done	IDP XT 2.0.1-EAR; IDP XT 2.0	IDP	
Performing a Verified Boot	Authoring:done	IDP XT 2.0.1-EAR; IDP XT 2.0	IDP	
Building Advantech UTX-3115 Boards	Authoring:work	IDP XT 2.0.1-EAR; IDP XT 2.0	IDP	
Setting Up a Zigbee Network	Authoring:done	IDP XT 2.0.1-EAR; IDP XT 2.0	IDP	
Search for [] -- (2014-04-11 12:42:34) [2 / 2]				

4. Right-click an item in the **Locator** results list and select **View Dependencies**.
5. Expand the **Cross-referenced by** section and identify the topic in your book that references the out-of-scope topic.

**NOTE:** You may need to go down several levels to find the topic included in your book.

6. Open the topic, locate the faulty link, and fix it.

If the link points to something outside the document or plug-in, remove it.

If the link is conditionalized for a particular product, audience, or other condition that is not included in your topic set but is included in a different topic set, you cannot remove it. You must comment out the paragraph in order to publish. If this is the case, you may want to revisit the way you are using conditionalization.

7. Run the validation again until it succeeds.



**NOTE:** Sometimes fixing one faulty link fixes many (or all) out-of-scope validation errors - validate the links after each fix.

---

